Kollmorgen Automation Suite

Integrated Development Environment (KAS-IDE) User Manual



Document Edition: JJ, July 2026
Valid for KAS Software Revision: 5.00
Valid for AKD Firmware Version: 1-25-00-000
Valid for AKD2G Firmware Version: 02-16-00-003
Valid for Kollmorgen Essentials Drive Firmware Version: 02-16-00-003
Part Number: 959715



For safe and proper use, follow these instructions.
Keep for future use.

KOLLMORGEN
A REGAL REXNORD BRAND

Table of Contents

1 Preface ... 43
1.1 System Requirements ... 43
1.1.1 Digital Signatures ... 44
2 PCMM2G Quick Start ... 45
3 Migrate from an AKD PDMM / PCMM (PxMM) to a PCMM2G Controller ... 46
3.1 Migration Overview ... 46
3.2 Prepare for Migration ... 46
3.2.1 Download the Required Software ... 46
3.2.2 Backup the Project ... 46
3.3 Optional: Prepare the Existing .KAS Project for Retained Variable Transfer ... 46
3.4 PCMM2G Hardware Preparation ... 48
3.5 PCMM2G Configuration ... 49
3.6 Migrate the .KAS Project ... 50
3.7 Migration Tips ... 55
3.8 Migration Validation Checklist ... 55
4 Kollmorgen Automation Suite Introduction ... 57
4.1 Learning Kollmorgen Automation Suite ... 58
4.1.1 Where to Start ... 58
4.1.1.1 Beginner ... 58
4.1.1.2 Advanced User ... 58
4.1.1.3 Online ... 58
4.1.2 Read the KAS Manuals ... 59
4.2 Key Features ... 61
4.2.1 Kollmorgen Automation Suite-Integrated Development Environment (KAS-IDE) ... 62
4.2.2 KAS Runtime ... 63
4.2.3 KAS Simulator ... 63
4.2.4 KAS Architecture ... 63
4.2.4.1 Real-Time ... 64
4.2.4.2 Non-Real-Time ... 64
4.2.5 Logical View ... 65
4.2.6 Physical View ... 67
4.3 KAS Breakdown ... 68
4.3.1 Human-Machine Interface ... 73
4.3.2 Controllers ... 73
4.3.2.1 PCMM2G Hardware ... 74
4.3.2.2 Controller Rotary Switch ... 74
4.3.2.3 Controller Web Server ... 75
4.3.2.4 Controller Performance ... 75
4.3.2.4.1 Performance Graphs ... 76
4.3.2.4.2.1 PCMM2G Controllers ... 76
4.3.2.4.3.2 PxMM Controllers ... 76
4.3.2.4.4 Application Complexities ... 77
4.3.2.4.5.1 High Complexity ... 77
4.3.2.4.6.2 Middle Complexity ... 77
4.3.2.4.7.3 Low Complexity ... 78
4.3.2.4.8 Pipe Network and PLCopen Applications ... 78

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4.3.3 Communication and Fieldbus ... 79
4.3.3.1 Fieldbus ... 79
4.3.3.2 Motion Bus ... 79
4.3.3.3 Ethernet ... 79
4.3.3.3.1 EtherCAT ... 79
4.3.4 Input/Output Bus ... 79
4.3.5 Drives ... 80
4.3.5.1 Related Documents ... 80
4.3.6 Kollmorgen Motors ... 80
4.3.6.1 Servomotors ... 80
4.3.6.2 Cartridge Motors ... 81
4.3.6.3 Direct Drives ... 81
4.4 Implement Multi-Axis with PCMM2G ... 81
5 KAS Graphical User Interface Description ... 83
5.1 Windows and Panels Overview ... 84
5.1.1 Main Window ... 84
5.1.2 Toolboxes ... 85
5.1.3 Project Explorer ... 85
5.1.3.1 System ... 86
5.1.3.2 Controller ... 86
5.1.3.3 Controller Node - Context Menu ... 87
5.1.3.3.1 PLC ... 88
5.1.3.3.2 Programs ... 89
5.1.3.3.3 Program Item - Contextual Menu ... 89
5.1.3.3.4 Subprograms ... 90
5.1.3.3.5 Subprogram Item - Context Menu ... 90
5.1.3.3.6 Defines ... 91
5.1.3.3.7 Internal Defines ... 91
5.1.3.3.8 Motion ... 91
5.1.3.3.9 Profiles ... 91
5.1.3.3.10 Pipe Network ... 92
5.1.3.4 PLCopen ... 92
5.1.3.4.1 Axes ... 92
5.1.3.4.2 Axes Groups ... 92
5.1.3.5 Control Panel ... 93
5.1.3.5.1 Controller Onboard I/O ... 93
5.1.3.5.2 EtherCAT ... 93
5.1.3.5.3 AKD2G Drive ... 94
5.1.3.5.4 AKD Drive ... 94
5.1.3.5.5 AKD-C Central Power Supply ... 94
5.1.3.5.6 AKD-N Drive ... 94
5.1.3.5.7 AKT (K-Bus) I/O Coupler ... 95
5.1.3.5.8 AKT2G (EtherCAT) I/O Coupler ... 95
5.1.3.5.9 I/O Slice ... 95
5.1.3.5.10.1 Device ... 95
5.1.3.5.11 Module ... 96
5.1.3.6 References ... 96
5.1.3.6.1 Fieldbus ... 96

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

5.1.3.7 HMI Device ... 96
5.1.3.7.1 KVB Project ... 96
5.1.3.8 SafePLC2 Project ... 97
5.1.3.8.1 Access the Web server From the KAS-IDE ... 97
5.1.3.8.2.1 Export a Control Panel ... 99
5.1.3.8.3.2 Import a Control Panel ... 99
5.1.3.8.4.3 Fieldbus Editor ... 100
5.1.4 Libraries ... 100
5.1.4.1 Function Blocks tab ... 101
5.1.4.2 Controls tab ... 101
5.1.4.3 Properties tab ... 101
5.1.5 Dictionary ... 101
5.1.5.1 Variables tab ... 102
5.1.5.1.1 Variables ... 106
5.1.5.1.2 Structures ... 107
5.1.5.1.3 Variable Editor ... 108
5.1.5.1.4.1 Create a New Variable ... 109
5.1.5.1.5 Variable Table List ... 109
5.1.5.1.6 Sort Variables ... 110
5.1.5.1.7 Define Structures ... 110
5.1.5.1.8 Name a Variable ... 110
5.1.5.1.9.1 Name Physical I/Os ... 111
5.1.5.1.10.2 Change the Name of an I/O Variable ... 111
5.1.5.1.11 Rename Variables ... 111
5.1.5.1.12 Variable Initial Value ... 115
5.1.5.1.13 Variable Tag and Description ... 115
5.1.5.1.14.1 Change the Tag or Description of a Variable ... 115
5.1.5.1.15 Edit Variables as Text using IEC 61131-3 Syntax ... 116
5.1.5.1.16.1 Basic Syntax for Declaring a Variable ... 116
5.1.6 Data Type: STRING ... 116
5.1.7 Read-only Attribute ... 116
5.1.8 Declare an Array ... 117
5.1.9 Single Variable Input Value ... 117
5.1.9.0.1.1 Additional Information and Description Texts ... 117
5.1.9.1 Enums tab ... 117
5.1.9.1.1 Adding Enums ... 118
5.1.9.1.2 Create Enums ... 118
5.1.9.1.3 Declaring Enums ... 118
5.1.9.2 Bit Fields tab ... 119
5.1.9.2.1 Adding Bit Fields ... 119
5.1.9.2.2 Using Bit Fields ... 120
5.1.9.2.3 Modbus Support ... 122
5.1.10 Information and Logs window ... 123
5.1.10.1 Log Messages tabs ... 123
5.1.10.1.1 Log Messages - Local Log ... 123
5.1.10.1.2 Log Messages - List of Fields ... 123
5.1.10.1.3.1 Log Messages - List of Buttons ... 124
5.1.10.2 Log Messages Settings ... 124

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

5.1.10.2.1 Configuration Settings 124
5.1.10.2.2 Message Sources 125
5.1.10.2.3 Log Message Levels 125
5.1.10.2.4.1 Choose the Appropriate Level for a Log 126
5.1.10.2.5 Local Log File Settings 126
5.1.10.2.6 Log File Storage 127
5.1.10.2.7 Filtering 128
5.1.10.2.8.1 Filtering Rules 128
5.1.10.2.9.2 Log Scrolling 128
5.1.10.3 Find and Replace tab 129
5.1.10.3.1 Location Details 131
5.1.10.4 Find and Replace Operations 132
5.1.10.4.1 Information and Logs 132
5.1.10.4.2 Dictionary 132
5.1.10.4.3 Editor 132
5.1.10.5 Browse Variable tab 133
5.1.10.5.1 Using the Browse Variable tab 133
5.1.10.5.2 Using the Dictionary's Browse Variable menu item 134
5.1.10.6 Breakpoints tab 135
5.1.10.7 Compiler Output tab 136
5.1.11 Watch Window 138
5.1.11.1 Multiple Watch Windows 139
5.1.11.1.1 Watch Window toolbar 139
5.1.11.1.2 Variable Information 139
5.1.11.2 Access Structure and Arrays 140
5.1.11.3 Add a Variable 140
5.1.11.4 Add an Expression 140
5.1.11.5 Force a Variable 142
5.1.11.5.1 Force 142
5.1.11.5.2 Lock 142
5.1.11.5.3 Unlock 143
5.1.12 Kollmorgen AKD Series Servo Drive 143
5.1.12.1 WorkBench Toolbar 143
5.1.12.2 WorkBench Status Bar 144
5.1.13 Status Bar 144
5.1.13.1 Local Version 144
5.1.13.2 Controller Version 145
5.1.13.3 Drives State 145
5.1.13.4 Controller State 145
5.1.13.5 Connection State 145
5.1.13.6 Color Codes 146
6 Getting Started 148
6.1 KAS-IDE Runtime Compatibility 148
6.1.1 Digital Signatures 149
6.2 KAS PLC Code Runtime Compatibility 149
6.3 KAS-IDE Project File Compatibility 150
6.3.1 Backward Compatibility 150
6.3.1.1 Update Options 150

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.3.2 Forward Compatibility ... 151
6.4 Start the KAS-IDE ... 151
6.4.1 Find the KAS-IDE Version ... 151
6.4.2 Access the KAS-IDE Help System ... 152
6.4.3 KAS GUI ... 152
6.4.4 KAS-IDE Local Log tab ... 152
6.4.5 Create a Project ... 153
6.4.5.1 Add the Controller ... 153
6.4.5.1.1 Available Application Templates ... 154
6.4.5.1.2 Coordinated Motion ... 154
6.4.5.1.3 KAS Runtime ... 155
6.4.5.1.4 KDN Download ... 155
6.4.5.1.5 Pipe Network ... 155
6.4.5.1.6 PLCopen ... 155
6.4.5.2 Configure the Controller ... 156
6.4.5.3 Configure Project Settings ... 157
6.4.5.3.1 Configure Custom Project Settings ... 159
6.4.5.4 Single-Core Compatibility Mode ... 161
6.4.5.4.1 Limitations and Guidelines ... 161
6.4.5.4.2.1 Operational Behavior ... 161
6.4.5.4.3.2 Impact on Migrated Applications ... 162
6.4.5.4.4.3 Application Guidelines ... 162
6.4.5.4.5.4 Recommendations ... 162
6.4.6 Configure EtherCAT Motion Bus ... 164
6.4.6.1 Slave Devices ... 164
6.4.6.2 EtherCATScan and Association ... 164
6.4.6.3 Scan and Associate Network Devices ... 167
6.4.6.3.1 Unknown – Missing ESI File ... 167
6.4.6.3.2 Missing ESI File - MDP Device ... 169
6.4.6.4 Re-Scan and Change Device Associations ... 171
6.4.6.5 Re-Scan and Change Coupler/Slice Associations ... 173
6.4.6.5.1 EtherCAT Scan & Association Example 1 - Associating Couplers and I/O Slices ... 173
6.4.6.5.2 EtherCAT Scan & Association Example 2 - Adding Physical I/O Slices ... 175
6.4.6.6 Re-Scan and Change Device/Module Associations ... 177
6.4.6.6.1 EtherCAT Scan & Association Example 3 - Add a Module to a Device ... 177
6.4.6.6.2 EtherCAT Scan & Association Example 4 - Remove a Module From a Device ... 179
6.4.7 Create a Program ... 182
6.4.7.1 Create Variables ... 183
6.4.7.2 Create and/or Call Functions or Function Blocks ... 185
6.4.7.2.1 Call a Function or Function Block ... 185
6.4.7.3 Using Constants ... 186
6.4.7.4 Map Input and Output to Variables ... 186
6.4.8 Add Motion ... 187
6.4.8.1 Design Motion with Pipe Network ... 187
6.4.8.1.1 Create the Pipe Network ... 187
6.4.8.1.2.1 Delete a Pipe Block ... 188
6.4.8.1.3.2 Change a Pipe Network Link ... 188
6.4.8.1.4 Edit Properties of Pipe Blocks ... 189

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.8.1.5 Map the Axis to the Drive ... 189
6.4.8.1.6 Add Comments to the Pipe Network ... 189
6.4.8.1.7 Set the Axis Block Position Units ... 190
6.4.8.1.8 Show Pipe Network and Profiles-generated Code ... 191
6.4.8.1.9 Pipe Network Functions for the PLC ... 191
6.4.8.1.10 Initialize and Start a Pipe Network ... 192
6.4.8.1.11 Pipe Network Interaction with PLC Programs ... 193
6.4.8.2 Design Motion with PLCopen Axis ... 195
6.4.8.2.1 Create a PLCopen Axis ... 195
6.4.8.2.2 Change a PLCopen Axis ... 196
6.4.8.2.3.1 Axis Name and Number ... 197
6.4.8.2.4.2 Common Axis Parameters ... 199
6.4.8.2.5.3 Digitizing Axis Parameters ... 200
6.4.8.2.6.4 Axis Data Parameters ... 202
6.4.8.2.7.5 Axis Limits Parameters ... 203
6.4.8.2.8 Create PLCopen Axes Groups ... 205
6.4.8.2.9 Initialize and Start PLCopen ... 206
6.4.8.3 Design Coordinated Motion ... 206
6.4.8.3.1 Create a Linear or Circular Coordinated Motion Application ... 206
6.4.9 Groups ... 207
6.4.9.0.1.1 Function Block Calls ... 207
6.4.9.0.2.2 What are Axes Groups? ... 210
6.4.9.0.3.3 Perform a Circular Move ... 210
6.5 Absolute Circular Move ... 210
6.6 Relative Circular Move ... 211
6.6.0.0.1.1 Perform a Linear Move ... 212
6.7 Absolute Linear Move ... 212
6.8 Relative Linear Move ... 213
6.9 Linear Move With More Than Two Axes ... 213
6.9.0.0.1.1 Circular Moves Diagrams ... 214
6.10 CircMode = BORDER ... 214
6.11 CircMode = CENTER ... 214
6.11.1 Add an HMI Device ... 215
6.11.1.1 Create a KVB Project ... 215
6.11.1.2 Map Variables to HMI ... 216
6.11.1.2.1.1 Variables to Export ... 217
6.11.1.2.2.2 Structures to Export ... 217
6.11.1.2.3.3 Structure that Cannot be Exported ... 217
6.11.1.3 Design KVB Panel with Kollmorgen Visualization Builder ... 218
6.11.2 SafePLC2 Projects ... 219
6.11.3 Overview: Compile, Connect, Download, and Run ... 220
6.11.4 Create Project - Advanced ... 220
6.11.5 Add and Configure a Servo Drive ... 220
6.11.5.1 Configure the Kollmorgen Servo Drive ... 221
6.11.5.2 Save Parameters to Non-Volatile Memory ... 223
6.11.5.3 AKD Setup Wizard ... 224
6.11.5.4 Configure Onboard I/O ... 224
6.11.5.5 Digital Input Mode ... 224

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.6 Add and Configure a Stepper Drive ... 225
6.11.6.1 AKT2G-IO-SM-Lxx Stepper Drive I/O Tab ... 225
6.11.6.2 AKT2G-IO-SM-Lxx Stepper Drive Configuration tab ... 226
6.11.6.2.1 Properties tab ... 226
6.11.6.2.2 Mode Section ... 227
6.11.6.2.3 Drive Section ... 228
6.11.6.2.4 Motor Section ... 228
6.11.6.2.5 Encoder Section ... 229
6.11.6.2.6 Diagnostics tab ... 230
6.11.7 Add and Configure I/O Slices ... 231
6.11.7.1 Add an AKT (K-Bus) I/O Coupler ... 231
6.11.7.2 Add an AKT (K-Bus) I/O Slice ... 232
6.11.7.3 Add an AKT2G-IO Stepper Drive ... 232
6.11.7.4 Configure the I/O Slice ... 233
6.11.8 Add and Configure Third-party EtherCAT Devices ... 234
6.11.8.1 Add Third-party EtherCAT Modules ... 236
6.11.8.1.1 Limitations ... 236
6.11.8.2 Configure a Third-party Drive ... 236
6.11.8.3 Delete EtherCAT Devices or Modules ... 239
6.11.8.3.1 Limitations ... 240
6.11.8.4 Modules - General Properties tab ... 242
6.11.9 Use Pre-defined Libraries ... 243
6.11.9.1 Drag-and-Drop into the Editors ... 243
6.11.9.2 Drag-and-Drop into the Dictionary ... 243
6.11.10 Create and Use Custom Libraries ... 244
6.11.10.1 Create a Custom Library ... 244
6.11.10.2 Use the Custom Library ... 244
6.11.10.2.1 What happens when you remove a library from your project? ... 246
6.11.10.2.2 What happens when a library no longer exists? ... 246
6.11.10.2.3 Broken Link Displayed in Italics ... 246
6.11.11 Add CAM Profiles ... 247
6.11.11.1 Create a CAM Profile ... 247
6.11.11.1.1 CAM Profile Parameters ... 247
6.11.11.1.2.1 Four Parameters Transforming the CAM Profile ... 248
6.11.11.2 Use Cam Profiles ... 250
6.11.11.3 CAM Profile Switching ... 251
6.11.12 Define Scheduling ... 252
6.11.12.1 Periodicity ... 252
6.11.12.2 Order of Program Execution ... 252
6.11.12.3 Define the PLC Cycle ... 253
6.11.12.3.1 Specify the Duration of a Cycle ... 254
6.11.12.3.2 Ensure Variables are Exported ... 254
6.11.12.4 Parent-Child Relationships and Execution Order ... 256
6.11.12.4.1 Heirarchy ... 256
6.12 Run the Project ... 257
6.12.1 Set the Compilation Options ... 257
6.12.1.1 Code Generation ... 258
6.12.1.1.1 Normal ... 258

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.12.1.1.2 Optimized ... 259
6.12.1.2 Conditional Compiling ... 259
6.12.1.2.1 Define Conditional Compiling Directives ... 260
6.12.2 Compile the Project ... 261
6.12.3 Start the KAS Simulator ... 262
6.12.4 Connect to the Controller ... 262
6.12.4.1 Actions to Prevent Compatibility Issues ... 264
6.12.4.2 Application Status Bar ... 264
6.12.4.3 Message Window ... 265
6.12.5 Download the Project ... 265
6.12.5.1 Project Name - Same Name ... 265
6.12.5.1.1 Actions ... 265
6.12.5.2 Project Name - Different Name ... 266
6.12.5.2.1 Actions ... 267
6.12.5.3 Application Status Bar ... 267
6.12.6 Device Control ... 267
6.12.6.1 Start / Stop the Device ... 267
6.12.6.2 Log Window ... 267
6.13 Test and Debug the Project ... 268
6.13.1 Step-By-Step Debugging ... 268
6.13.2 Breakpoints ... 270
6.13.2.1 Active or Inactive Breakpoints ... 270
6.13.2.2 Breakpoint Usage ... 271
6.13.2.3 Online Change ... 271
6.13.3 Set, Remove, Enable, and Disable Breakpoints ... 271
6.13.3.1 Set Breakpoints ... 271
6.13.3.1.1 About SFC ... 272
6.13.3.2 Enable/Disable a Breakpoint ... 272
6.13.4 Printf Function ... 272
6.13.4.1 Customize Output in the Log window ... 273
6.13.5 Soft Oscilloscope Debugging ... 273
6.13.5.1 Plug Motion Variables ... 274
6.13.5.1.1 Usage Example with the Pipe Network ... 275
6.13.5.2 Plug PLC Variables ... 275
6.13.6 Compare PLC Programs ... 277
6.13.6.1 Limitations of FFLD Diagrams ... 277
6.13.7 Variable Animation ... 278
6.13.7.1 Online Change ... 278
6.13.7.1.1 Limitations ... 278
6.13.7.2 Variable Monitoring ... 278
6.13.7.2.1 Monitoring UDFBs ... 279
6.13.7.2.2 Force a Variable ... 279
6.13.7.2.3.1 Force ... 280
6.13.7.2.4.2 Lock ... 280
6.13.7.2.5.3 Unlock ... 281
6.13.7.3 IEC 61131-3 Editor Debugging ... 281
6.13.7.4 Pipe Network Editor Debugging ... 282
6.14 Manage a Project ... 282

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.14.1 Use a Version Control System ... 282
6.14.2 Print ... 282
6.14.2.1 Printable Elements ... 283
6.14.2.2 Page Setup ... 283
6.14.2.2.1 Margin and Header/Footer tab ... 283
6.14.2.3 Print ... 283
6.14.2.4 Print Preview ... 284
6.14.2.5 Print the Project ... 284
6.14.3 Use the Reference Folder ... 284
7 Understanding KAS ... 286
7.1 IEC 61131-3 ... 286
7.1.1 Data Types ... 286
7.1.1.1 Different Data Types ... 286
7.1.1.2 List of Data Types ... 286
7.1.1.3 Arrays ... 287
7.1.1.4 Bitfields ... 288
7.1.1.5 Enumerations ... 288
7.1.1.6 Structure ... 288
7.1.1.6.1 Limitation ... 289
7.1.2 Variables ... 289
7.1.2.1 Retain Variables ... 289
7.1.2.2 Retained Variable Initialization Behavior ... 289
7.1.2.3 Warning - Dependency on Project Compatibility ... 289
7.1.2.4 Non-volatile Storage Considerations ... 289
7.1.2.5 Actions that Reset Retained Values to Init Values ... 290
7.1.2.6 Variables ... 290
7.1.2.6.1 Groups ... 291
7.1.2.6.2 Data Type and Dimension ... 291
7.1.2.6.3 Name a Variable ... 291
7.1.2.6.4 Variable Attributes ... 292
7.1.2.7 Retain Variables ... 292
7.1.2.7.1 Retained Variable Initialization Behavior ... 292
7.1.2.7.2 Warning - Dependency on Project Compatibility ... 292
7.1.2.7.3 Non-volatile Storage Considerations ... 293
7.1.2.7.4 Actions that Reset Retained Values to Init Values ... 293
7.1.2.7.5 Best Practices Retained Variables ... 293
7.1.2.7.6 Copy Retained Variable Values to Init Value ... 294
7.1.3 Preconditions ... 294
7.1.3.0.1 Copy Retained Variables Values to Init Value from another Controller ... 298
7.1.3.0.2.1 Transfer Retain Variables from Source Controller to the KAS Project ... 299
7.1.4 Validation ... 300
7.1.5 Transfer Failures ... 300
7.1.5.0.1.1 Transfer Retained Variables values to the Target Controller ... 301
7.1.6 Constant Expressions ... 301
7.1.6.1 Examples ... 305
7.1.6.1.1 Valid Constant Expressions ... 305
7.1.6.1.2 Invalid Constant Expressions ... 306
7.1.7 Program Organization Units ... 306

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.1.7.1 Differences between Functions and Function Blocks ... 306
7.1.7.1.1 Description of FB Operation ... 306
7.1.7.1.2 Examples of Operations Overrunning the Cycle Duration ... 306
7.1.7.1.3 Operation Sequence ... 306
7.1.7.2 Functions (POU) ... 307
7.1.7.3 Function Blocks (POU) ... 307
7.1.7.4 Programs (POU) ... 308
7.1.7.4.1 Sub-programs ... 308
7.1.7.4.2 Program Guidelines ... 309
7.1.7.4.3.1 Program Name ... 309
7.1.7.4.4.2 Child SFC Programs ... 309
7.1.7.4.5 Program Limitations ... 310
7.1.7.4.6.1 FFLD Programs ... 310
7.1.7.4.7.2 Other Programs ... 310
7.1.7.4.8.3 SFC Programs ... 310
7.1.7.4.9 Action Blocks (POU) ... 310
7.1.7.4.10.1 Concept and Purpose ... 312
7.1.7.4.11.2 Supported Environments ... 312
7.1.7.4.12.3 Action Block Ownership and Scope ... 312
7.1.7.4.13.4 Creating an Action Block ... 312
7.1.7.4.14.5 Implementing Logic Inside an Action Block ... 313
7.1.7.4.15.6 Using Action Blocks ... 313
7.1.8 Unqualified Invocation ... 313
7.1.9 Qualified Invocation (Fully Qualified Name) ... 314
7.1.10 Name Disambiguation Example ... 314
7.1.10.0.1.1 Debugging and Maintenance ... 315
7.1.11 Online Monitoring ... 315
7.1.12 Maintenance Benefits ... 315
7.1.12.0.1.1 Action Blocks have an Unexpected Variable Selector Prompt in FFLD Editors ... 315
7.1.12.1 User-Defined Function Blocks (POU) ... 315
7.1.13 Programming Languages ... 316
7.1.13.1 Free Form Ladder Diagram (FFLD) ... 316
7.1.13.2 Function Block Diagram (FBD) ... 316
7.1.13.3 Instruction List (IL Language) ... 316
7.1.13.4 Sequential Function Chart (SFC) ... 316
7.1.13.4.1 Alternative and Parallel Sequences ... 317
7.1.13.5 Structured Text (ST Language) ... 317
7.1.14 Alias Definitions ... 318
7.1.14.1 Example ... 319
7.1.15 Exception Handling ... 319
7.1.15.1 Division By Zero Exceptions ... 319
7.1.15.2 Shutdown Exceptions ... 320
7.1.15.3 Startup Exceptions ... 320
7.2 Motion Concepts ... 320
7.2.1 Introducing Motion ... 320
7.2.1.1 Motion Control Main Functions ... 320
7.2.1.1.1 Setpoint Generation ... 320
7.2.1.1.2 Regulation ... 320

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.1.2 Single and Multi-Axis Motion ... 321
7.2.1.3 Hardware Organization of Motion Functions ... 321
7.2.1.4 Motion Profile ... 322
7.2.1.5 Executing Motion ... 323
7.2.1.6 Motion State Machine ... 323
7.2.1.7 KAS Touch Probe and Fast Input - Overview ... 323
7.2.1.7.1 Configuration ... 323
7.2.1.7.2.1 Drive Configuration ... 324
7.2.1.7.3.2 Pipe Network Configuration ... 324
7.2.1.7.4.3 PLCopen Configuration ... 324
7.2.1.7.5 List of Function Blocks and Pipe Blocks ... 324
7.2.1.7.6.1 PLCopen Function Blocks ... 324
7.2.1.7.7.2 Pipe Network Blocks ... 324
7.2.2 Pipe Network or PLCopen ... 324
7.2.2.1 Pipe Network ... 325
7.2.2.2 PLCopen ... 325
7.2.2.3 Pipe Network and PLCopen Motion Engine Differences ... 325
7.2.3 Pipe Network Concept ... 326
7.2.3.1 Pipe Network ... 327
7.2.3.2 Pipe Structure ... 328
7.2.3.3 Pipe Block ... 328
7.2.3.3.1 Pipe Network - List of Pipe Blocks ... 329
7.2.3.3.2 Change the Information Flow from Position to Velocity ... 332
7.2.3.4 Axis Pipe Block ... 332
7.2.3.4.1 Associated Data on Positions ... 333
7.2.3.4.2 Axis Block Initialization ... 334
7.2.3.4.3 Axis Connection to a Pipe ... 334
7.2.3.4.4 Change Axis Position Functions ... 334
7.2.3.4.5 Homing ... 335
7.2.3.4.6 Monitor an Axis ... 335
7.2.3.4.7 Multi-Axis Operation ... 335
7.2.3.4.8 Realign Positions ... 336
7.2.3.4.9 Set Zero Axis ... 336
7.2.3.4.10 Single-Axis Operation ... 336
7.2.3.4.11 Units ... 336
7.2.3.4.12 View Axis Block Position Functions ... 336
7.2.3.5 Pipe Block Lifetime ... 337
7.2.3.5.1 Activation ... 337
7.2.3.5.2 Usage ... 337
7.2.3.5.3 Deactivation ... 337
7.2.3.6 Pipe Network Phase Execution ... 337
7.2.3.6.1 Absolute Phase of Execution ... 337
7.2.3.6.2 Relative Phase of Execution ... 337
7.2.3.7 Use Motion Function Block for Pipe Network ... 337
7.2.3.7.1 Buffer Mode ... 338
7.2.3.7.2 Motion Init ... 338
7.2.3.7.3 Motion Start ... 338
7.2.3.7.4 Using the Q Output of ML Function Blocks for the Pipe Network ... 338

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.3.8 Pipe Network - General Rules ... 338
7.2.3.8.1 Languages ... 338
7.2.3.8.2 BlockID Inputs ... 338
7.2.3.8.3 Output Status ... 339
7.2.3.8.4 What is the Difference between Q and OK? ... 339
7.2.3.8.5 When Q is Set to TRUE ... 339
7.2.3.8.6 Input Parameters ... 340
7.2.3.8.7 Missing Input Parameters ... 340
7.2.3.8.8 Position versus Distance ... 340
7.2.3.8.9 Default Block Parameters ... 340
7.2.3.9 Use the Generated Pipe Network Function ... 340
7.2.4 PLCopen ... 341
7.2.4.1 PLCopen Function Blocks ... 342
7.2.4.1.1 PLCopen Function Block ErrorIDs ... 343
7.2.4.1.2 Axes Group Parameters ... 347
7.2.4.1.3.1 Boolean Parameters ... 347
7.2.4.1.4.2 Non-Boolean Parameters ... 347
7.2.4.1.5 Axis Parameters ... 347
7.2.4.1.6.1 Update Rate Types ... 348
7.2.5 Controller ... 348
7.2.6 EtherCAT Cyclic ... 348
7.2.7 EtherCAT Non-Cyclic ... 348
7.2.8 Static ... 348
7.2.8.0.1.1 User Units to Feedback Units Ratio ... 348
7.2.9 Example 1 ... 348
7.2.10 Example 2 ... 348
7.2.11 Example 3 - Reading the Current to Units of Amps ... 349
7.2.11.0.1.1 Boolean Parameters ... 349
7.2.11.0.2.2 Non-Boolean Parameters ... 355
7.2.11.0.3 Axis Positions Data ... 369
7.2.11.0.4.1 Actual Position ... 369
7.2.11.0.5.2 CommandPosition ... 369
7.2.12 CoordinatedMotionCmdPos ... 369
7.2.13 Group Actual Position ... 369
7.2.14 NormalCmdPos ... 370
7.2.15 PhaseCmd Pos ... 370
7.2.16 SuperimposedCmdPos ... 370
7.2.16.0.1 AXIS_REF Structure ... 371
7.2.16.0.2 Axis Velocity Data ... 371
7.2.16.0.3.1 Actual Velocity ... 371
7.2.16.0.4.2 Command Velocity ... 371
7.2.16.0.5 Buffer Modes ... 371
7.2.16.0.6 Function Block Acceleration and Jerk Parameters ... 373
7.2.16.0.7.1 Rules ... 373
7.2.17 Generalizations ... 374
7.2.17.0.1.1 Methods ... 374
7.2.18 Example 1 ... 374
7.2.19 Example 2 ... 374

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.20 Example 3 ... 374
7.2.20.0.1.1 Limitations on Acceleration and Jerk ... 375
7.2.20.0.2 Move Types ... 375
7.2.20.0.3 Profile Generator ... 375
7.2.20.0.4 Queuing ... 375
7.2.20.0.5 S-curve and Trapezoidal Acceleration / Deceleration ... 376
7.2.20.0.6.1 S-curve ... 376
7.2.20.0.7.2 Small Jerk Acceleration ... 376
7.2.20.0.8.3 Large Jerk Acceleration ... 376
7.2.20.0.9.4 Trapezoidal ... 377
7.2.20.0.10 Velocity Compensation ... 378
7.2.20.0.11.1 Velocity Compensation Factor ... 379
7.2.20.0.12.2 Velocity Compensation Filter ... 379
7.2.20.1 Function Blocks - General Rules ... 379
7.2.20.1.1 Active Output ... 379
7.2.20.1.2.1 Coordinated Motion ... 379
7.2.20.1.3 Behavior of Busy Output ... 380
7.2.20.1.4 Behavior of CommandAborted Output ... 380
7.2.20.1.5 Behavior of Done Output ... 380
7.2.20.1.6 Error Handling Behavior ... 380
7.2.20.1.7 Input Parameters ... 381
7.2.20.1.8.1 PLCopen Function Blocks with Execute ... 381
7.2.20.1.9.2 PLCopen Function Blocks with Enable ... 382
7.2.20.1.10.3 PLCopen Functions with Input Parameter En ... 382
7.2.20.1.11 Inputs Exceed Application Limits ... 383
7.2.20.1.12 Missing Input Parameters ... 383
7.2.20.1.13 Output Exclusivity ... 383
7.2.20.1.14 Output Status ... 383
7.2.20.1.15 Sign Rules ... 383
7.2.20.2 PLCopen State Machine ... 383
7.2.20.2.1 Defined States ... 383
7.2.20.2.2.1 State Machine for Multi-axes Motion Control ... 385
7.2.20.2.3 Disabled State ... 385
7.2.20.2.4 ErrorStop State ... 385
7.2.20.2.5 From StandStill to Stopping ... 386
7.2.20.3 Program a Multi-Core Controller ... 386
7.2.20.3.1 Cached Data ... 386
7.2.20.3.2.1 Function Blocks - Return Cached Data ... 386
7.2.20.3.3 Calling Function Blocks Multiple Times Same Cycle ... 387
7.2.20.3.4.1 Function Blocks - Start Motion-related Actions ... 388
7.2.20.3.5 Calling Conflicting Pipe Network Functions in the Same Cycle ... 389
7.2.20.3.6.1 Pipe Network Functions - Motion-related Actions and Parameters ... 389
7.2.21 Recipes ... 390
7.2.21.1 What is a Recipe? ... 391
7.2.21.2 What Problems Do Recipes Solve? ... 392
7.2.21.2.1 Example Use Cases ... 392
7.2.21.3 Core Concepts ... 392
7.2.21.3.1 Recipe Definition ... 392

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.3.2 Recipe Files ... 393
7.2.21.3.3 Recipe Instance ... 393
7.2.21.3.4 Recipe Variable ... 394
7.2.21.3.5.1 Data Types - Supported ... 394
7.2.21.4 Quick Start: Basic Recipe Workflow ... 395
7.2.21.5 Recipe File Formats and Examples ... 396
7.2.21.5.1 Reserved Recipe Sections ... 397
7.2.21.5.2 Choosing XML or TOML ... 397
7.2.21.5.3 Example: Recipe Definition File ... 397
7.2.21.5.4 Example: Recipe Definition and Multiple Instance File ... 399
7.2.21.5.5 Example: Recipe Definition and Single Instance File ... 401
7.2.21.5.6 Example: Recipe Definition and User Audit Entries ... 402
7.2.21.5.7 Example: Recipe Instance File ... 404
7.2.21.6 Recipe Management ... 405
7.2.21.6.1.1 Recipe Workflow ... 405
7.2.21.6.2 Create and Edit Recipe Files ... 406
7.2.21.6.3 Download Recipes to the Controller ... 407
7.2.21.6.4.1 Transfer via Controller Web Server ... 408
7.3 View File Contents from the User Data File Explorer ... 409
7.4 Upload Folders and Files to the Controller ... 409
7.5 File Management Actions ... 410
7.5.1 Create Folder ... 410
7.5.2 Select All Files ... 410
7.5.3 Deselect All Files ... 410
7.5.4 Download Files ... 410
7.5.5 Delete ... 410
7.5.6 Clear User Data ... 410
7.5.7 Refresh ... 410
7.5.7.0.1.1 Transfer via HTTP API ... 411
7.5.7.0.2 Runtime-based Recipe Management through Function Blocks ... 413
7.5.7.1 Recipe Design Example ... 414
7.5.7.1.1 Variable Configurations ... 414
7.5.7.1.2 Example - Processing Parameters (Thermal and Mixing) ... 414
7.5.7.1.3 Example - Carbonation Parameters (Pressure Controls) ... 414
7.5.7.1.4 Example - Motion and Bottling Parameters (Cam Profiles and Packaging) ... 414
7.5.7.1.5 Architectural Approaches and Tradeoffs ... 415
7.5.7.1.6.1 Approach A: Multiple Modular Recipe Files ... 415
7.5.7.1.7.2 Approach B: Single Master Recipe ... 415
7.5.7.2 Recipe Troubleshooting ... 415
7.6 EtherCAT Motion Bus Concepts ... 416
7.6.1 EtherCATProcess Image ... 416
7.6.1.1 References ... 416
7.6.2 EtherCAT Functional Principle ... 416
7.6.3 EtherCAT Features ... 417
7.6.3.1 EtherCAT is Versatile ... 417
7.6.3.2 Safety over EtherCAT is Available ... 417
7.6.3.3 EtherCAT Device Profiles ... 418
7.6.3.3.1 CANopen over EtherCAT (CoE) ... 418

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.6.3.4 EtherCAT Distributed Clock (Synchronization) 418
7.6.3.5 File Access over EtherCAT (FoE) 419
7.6.3.6 EtherCAT Gateways 419
7.6.3.7 EtherCAT Performance 420
7.6.3.8 EtherCAT Protocol 421
7.6.3.9 Safety over EtherCAT 421
7.6.3.10 EtherCAT Topology 422
7.6.4 EtherCAT Implementation 423
7.6.4.1 EtherCAT Master Configuration 423
7.6.4.1.1 ENI and ESI Files 424
7.6.4.2 EtherCAT Slave Configuration 425
7.6.4.2.1 EtherCAT Slave Controller 425
7.6.4.3 EtherCAT State Machine 426
7.6.4.3.1 Different States 427
7.6.4.4 PDOs for AKD, AKD2G, AKD-N, Kollmorgen Essentials Drive, S300, or S700 427
7.6.4.4.1 PDO Types 427
7.6.4.4.2 Examples 427
7.6.4.4.3.1 Example 1 427
7.6.4.4.4.2 Example 2 428
7.6.4.4.5.3 Example 3 428
7.6.4.4.6 From Controller to Drive (RxPDO) 428
7.6.4.4.7.1 RxPDO - AKD2G Drive CANopen Objects 428
7.6.4.4.8.2 RxPDO - AKD Drive CANopen Objects 429
7.6.4.4.9.3 RxPDO - AKD-N Drive CANopen Objects 430
7.6.4.4.10.4 RxPDO - KED Drive CANopen Objects 431
7.6.4.4.11.5 RxPDO - S300 and S700 Drive CANopen Objects 432
7.6.4.4.12 From Drive to Controller (TxPDO) 433
7.6.4.4.13.1 TxPDO - AKD2G Drive CANopen Objects 433
7.6.4.4.14.2 TxPDO - AKD Drive CANopen Objects 435
7.6.4.4.15.3 TxPDO - AKD-N Drive CANopen Objects 437
7.6.4.4.16.4 TxPDO - KED Drive CANopen Objects 438
7.6.4.4.17.5 TxPDO - S300 and S700 Drive CANopen Objects 439
7.6.5 CANopen 440
7.6.5.1 CANopenStatus Machine 440
7.6.5.1.1 Status Description 441
7.6.5.1.2 Transitions of the Status Machine 442
7.6.5.1.3.1 Transition Events and Actions 442
7.6.5.2 CANopen Control Word 443
7.6.5.2.1 Bit Assignment in Control Word 443
7.6.5.2.2 Coded Commands 444
7.6.5.3 CANopenStatus Word 444
7.6.5.3.1 Bit Assignment in Status Word 444
7.6.5.3.2 State Coding 445
7.7 AKD Drive 445
7.7.1 AKD Drive Configuration 446
7.7.1.1 Connection Modes 446
7.7.1.1.1 Disconnected 446
7.7.1.1.2 Online Configuration Mode 446

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.7.1.1.3 Project Running on the Controller ... 446
7.7.1.2 AKD Configuration According to EtherCAT State ... 447
7.7.2 AKD Drive Configuration ... 448
7.7.2.1 Connection Modes ... 448
7.7.2.1.1 Disconnected ... 448
7.7.2.1.2 Online Configuration Mode ... 448
7.7.2.1.3 Project Running on the Controller ... 448
7.7.2.2 AKD Configuration According to EtherCAT State ... 449
7.8 Tasking Model / Scheduling ... 450
7.8.1 Priority between Motion and PLC ... 450
7.8.1.1 EtherCAT Processing Time ... 451
7.8.1.1.1 Variation during the EtherCAT Processing ... 451
7.8.1.2 NVRAM Processing Time ... 451
7.8.1.3 Results when a PLC Program is Overrunning the Cycle Duration ... 452
7.8.2 Change the Priority between PLC Programs ... 452
8 KAS-IDE Reference ... 453
8.1 Choose a Workspace Layout ... 453
8.1.1 Move Child Windows ... 453
8.1.2 Move Toolbox ... 453
8.1.3 Dock Window ... 453
8.2 Menus Overview ... 455
8.2.1 File menu ... 455
8.2.2 Edit menu ... 460
8.2.3 Project menu ... 461
8.2.3.1 Project Settings dialog box ... 461
8.2.4 Help menu ... 463
8.2.5 Tools menu ... 464
8.2.5.1 User Options dialog box ... 464
8.2.6 View menu ... 467
8.2.7 Window menu ... 468
8.3 Toolbar Overview ... 469
8.3.1 Debug toolbar ... 470
8.3.2 Device toolbar ... 470
8.3.3 EtherCAT Toolbar ... 471
8.3.4 Main (Tools) toolbar ... 471
8.3.5 Online Change toolbar ... 472
8.4 Windows Standard Conventions ... 472
8.4.1 Windows Manipulation ... 472
8.4.2 Mouse Manipulation ... 473
8.4.3 Table Manipulation ... 473
8.4.3.1 Sorting Items ... 473
8.4.3.2 Selecting a Cell ... 473
8.4.3.3 Selecting a Row ... 473
8.4.3.4 Resize a Column ... 473
8.5 Keyboard Shortcuts ... 473
8.5.1 Common Keyboard Shortcuts ... 474
8.5.2 FBD Editor Keyboard Shortcuts ... 475
8.5.2.1 FBD Editor (common) ... 475

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.5.2.2 FBD Editor (when editing) 475
8.5.2.3 FBD Editor (during debug) 475
8.5.3 FFLD Editor Keyboard Shortcuts 476
8.5.3.1 FFLD Editor (when editing) 476
8.5.3.1.1 Bookmark 476
8.5.3.1.2 Delete 476
8.5.3.1.3 Edit 476
8.5.3.1.4 Find 477
8.5.3.1.5 Insert 477
8.5.3.1.6 Move 477
8.5.3.1.7 Select 478
8.5.3.1.8 Trace 478
8.5.3.2 FFLD Editor (during debug) 478
8.5.4 Graphic Editor Keyboard Shortcuts 479
8.5.5 SFC Editor Keyboard Shortcuts 479
8.5.6 ST Editor Keyboard Shortcuts 480
8.5.6.1 ST Editor (common) 480
8.5.6.2 ST Editor (when editing) 480
8.5.6.3 ST Editor (during debug) 480
8.5.7 Table Keyboard Shortcuts 480
8.6 Bookmarks 481
8.7 Programs: Structure and Syntax 481
8.7.1 Project Structure 481
8.7.2 IEC 61131-3 Editors 481
8.7.3 Tips 482
8.7.3.1 Drag-and-Drop 482
8.7.3.2 Auto-completion 482
8.7.3.3 Tooltip on Variable 482
8.7.3.4 Bookmarks 483
8.7.4 Select Function Blocks 483
8.7.5 Select Variables and Instances 483
8.7.6 Autocompletion when Programming 483
8.7.7 Drag and Drop Programming 484
8.7.8 Select Function Blocks 484
8.7.9 Select Variables and Instances 484
8.7.10 Variables Tooltips 485
8.7.11 Change Operators and Functions in FBD or FFLD Editors 485
8.7.12 Change Function Blocks in FBD or FFLD Editors 486
8.7.13 Function Block Diagram (FBD) Editor 487
8.7.13.1 FFLD Components 488
8.7.13.2 Toggle the Connection to Make it Negative 488
8.7.13.3 Display the Execution Order 488
8.7.13.4 FBD Toolbar 489
8.7.13.4.1 FBD Comments 490
8.7.13.4.2 FBD Corners 491
8.7.13.4.3 FBD Network Breaks 491
8.7.13.4.4 FBD OR Vertical Rail 491
8.7.13.4.5 FBD Variables 491

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.7.13.5 Draw FBD Connection Lines ... 492
8.7.13.6 Insert FBD Objects on a Line ... 493
8.7.13.7 Move or Copy FBD Objects ... 493
8.7.13.7.1 Move Objects ... 494
8.7.13.7.2 Extend Object Selection ... 494
8.7.13.7.3.1 Insert or Delete Space in the Diagram ... 494
8.7.13.7.4 Auto-alignment ... 494
8.7.13.8 Resize FBD Objects ... 494
8.7.13.9 Select FBD Variables and Instances ... 495
8.7.13.10 View FBD Diagrams ... 495
8.7.14 Free Form Ladder Diagram (FFLD) Editor ... 496
8.7.14.1 Advanced Graphic Features ... 496
8.7.14.1.1 Change Ladder Diagram Size ... 496
8.7.14.1.2 Color-coded Functions ... 496
8.7.14.1.3 Coordinate-based Positions ... 497
8.7.14.1.4 Select the Entire Ladder Diagram ... 497
8.7.14.1.5 Split Window Capability ... 498
8.7.14.2 FFLD Toolbar and Right-click Menu ... 498
8.7.14.2.1 Toolbar ... 498
8.7.14.2.2 Right-click Menu ... 500
8.7.14.3 FFLD Networks ... 500
8.7.14.3.1 Header Row ... 500
8.7.14.3.2 New Networks ... 501
8.7.14.4 FFLD Runtime ... 501
8.7.14.5 FFLD Power Rail and Lines ... 501
8.7.14.6 FFLD Contacts and Coils ... 502
8.7.14.7 FFLD Function Blocks ... 502
8.7.14.8 FFLD Data In / Data Out ... 502
8.7.14.9 FFLD Jumps and RETURN ... 503
8.7.14.10 FFLD Selection Grid ... 503
8.7.14.11 FFLD Move and Copy Items ... 504
8.7.14.12 View FFLD Diagrams ... 505
8.7.15 Sequential Function Chart (SFC) Editor ... 505
8.7.15.1 SFC Toolbar ... 506
8.7.15.1.1 Edition window ... 507
8.7.15.2 Draw SFC Divergences ... 507
8.7.15.2.1 Simple or Double Divergence Lines ... 508
8.7.15.3 Enter Actions of a Step ... 508
8.7.15.4 Enter the Condition of a Transition ... 509
8.7.15.5 Enter Notes for Steps and Transitions ... 510
8.7.15.6 Move or Copy SFC Charts ... 510
8.7.15.7 Re-number Steps and Transitions ... 510
8.7.15.8 View SFC Charts ... 510
8.7.16 Structured Text (ST) / Instruction List (IL) Editor ... 511
8.7.16.1 ST / IL Language Selection ... 511
8.7.16.2 ST / IL Syntax Coloring ... 512
8.7.16.3 Intellisense ... 512
8.7.16.3.1 Conditional Compiling Coloring ... 512

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.7.16.3.2 Auto-indentation ... 513
8.7.16.4 Auto-completion of Words ... 513
8.7.16.4.1 Other Syntax-related Commands ... 514
8.7.16.5 ST / IL Drag-and-Drop ... 514
8.7.16.6 Read the Output of an MC Function Block in ST ... 514
8.7.16.6.1.1 Example 1 ... 515
8.7.16.6.2.2 Example 2 ... 515
8.8 Create Variables ... 515
8.8.1 Use the Dictionary ... 515
8.8.2 Create Variables from the Editors ... 515
8.8.3 Data Types ... 517
8.8.4 Complex Structures ... 517
8.8.4.1 Declare the Structure ... 517
8.8.4.2 Create an Instance of the Structure ... 518
8.8.5 Variable Editor ... 519
8.8.6 Edit Variables as Text ... 519
8.9 Create Functions and Function Blocks ... 520
8.9.1 Declare Functions or Function Blocks ... 521
8.9.2 Define Parameters and Private Variables ... 522
8.9.3 Finalize Functions or Function Blocks ... 523
8.9.4 Call Functions or Function Blocks ... 523
8.10 Use the Defines List ... 523
8.10.1 Internal Defines ... 524
8.10.1.1 Example: Pre-defined Constants ... 524
8.10.2 Global Defines ... 525
8.10.2.0.1 Definition Inside Another Definition ... 526
8.10.2.1 Local Definitions ... 526
8.11 Map Input and Output to Variables ... 526
8.11.1 Map I/O from the Project Explorer ... 526
8.11.1.1 Access to Device Properties ... 529
8.11.2 PLC Variable Creation Wizard ... 530
8.11.3 PLC Variable Selector dialog box ... 531
8.11.3.1 Create PLC Variable Button ... 531
8.11.3.2 Advanced Button ... 532
8.11.4 Analog I/O Parameters ... 533
8.11.4.1 AKT2G Input Terminals ... 533
8.11.4.2 AKT2G Output Terminals ... 533
8.11.4.3 AKT Input Terminals ... 534
8.11.4.4 AKT Output Terminals ... 534
8.12 Configure an EtherCAT ... 535
8.12.1 EtherCAT Devices tab ... 536
8.12.1.1 EtherCAT Devices ... 536
8.12.1.2 Scan Devices ... 538
8.12.1.3 EtherCAT Network - Logical View ... 538
8.12.1.3.1 EtherCAT Device Icons ... 539
8.12.1.3.2 Scan Limitations ... 539
8.12.2 EtherCAT Master Settings tab ... 539
8.12.2.1 Bandwidth Calculation Algorithm ... 540

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.12.3 ENI File tab ... 541
8.12.3.1 Import or Export an EtherCAT ENI File ... 541
8.12.3.2 Use an Imported ENI File ... 542
8.12.4 ESI File Management ... 542
8.12.4.1 Opening a Project ... 542
8.12.4.2 Adding or Deleting ESI files ... 542
8.12.4.3 Saving a Project ... 543
8.12.5 ESI Files tab ... 544
8.12.6 ESI Files with References ... 545
8.12.6.1 EtherCAT Device Configuration - MDP Devices ... 546
8.12.7 FSoE / PDO Connections tab ... 546
8.13 EtherCAT Devices ... 548
8.13.1 General Properties tab - EtherCAT Device ... 548
8.13.1.1 Information area ... 549
8.13.1.2 Topology area ... 549
8.13.2 Map I/O from the Project Explorer ... 550
8.13.2.1 Access to Device Properties ... 552
8.13.3 Configuration tab ... 554
8.13.4 PDO Selection/Mapping tab ... 554
8.13.4.1 Map PLC Variable to PDO Object ... 555
8.13.4.2 Select Input and Output PDOs ... 556
8.13.4.3 View the Contents of a PDO ... 557
8.13.4.4 Set Up a Variable to Monitor Safety Data ... 558
8.13.5 PDO Editor tab ... 560
8.13.5.1 Editable PDOs ... 561
8.13.5.1.1 Add a Flexible PDO ... 562
8.13.5.1.2 Edit a Flexible PDO ... 562
8.13.5.1.3 Delete a Flexible PDO ... 563
8.13.5.2 PDO Content ... 563
8.13.5.2.1 Add a CoE Object ... 564
8.13.5.2.2 Rename a CoE Object ... 565
8.13.5.2.3 Delete a CoE Object ... 565
8.13.5.3 PDO Restrictions and Compiler Errors ... 565
8.13.5.3.1 Restrictions for AKD Devices ... 565
8.13.5.3.2 Restrictions for AKD2G Devices ... 566
8.13.5.3.3 Restrictions for Kollmorgen Essentials Drive Devices ... 566
8.13.5.3.4 Restrictions for All Other Devices ... 567
8.13.5.4 Redundant PDO Entries ... 567
8.13.5.4.1 Redundant Rx PDO ... 567
8.13.5.4.2 Redundant Tx PDO ... 568
8.13.5.5 Safety, PDOs, and Variables ... 568
8.13.6 Distributed Clock tab ... 568
8.13.6.1 Oversampling Devices ... 570
8.13.7 CoE Init Commands tab ... 570
8.13.7.0.1 Functions ... 571
8.13.7.1 Add CoE Commands ... 573
8.13.8 CoE Object Dictionary tab ... 575
8.13.8.0.1 CoE Table Descriptions ... 576

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

9 Controller Error and Alarm Descriptions 577
9.1 Errors 577
9.2 Alarms 583
9.3 Actual Position Interpolation 584
9.4 CPU Overload (E23) 585
10 Using the KAS Simulator 586
10.1 KAS Simulator Graphical User Interface 586
10.1.1 Main window 586
10.1.2 Log window 586
10.1.3 KAS Simulator - Menus 587
10.1.3.1 File Menu 587
10.1.3.2 Help Menu 587
10.1.4 KAS Simulator window 587
10.2 Start the KAS Simulator 589
10.2.1 KAS Runtime Log window 590
10.3 Stop the KAS Simulator 591
11 Using the Controllers 593
11.1 Booting the AKD PDMM or PCMM Controller 593
11.2 Booting the PCMM2G Controller 593
11.2.1 Active Partition 594
11.2.2 Boot Partition 594
11.2.3 Data Partition 594
11.2.4 Inactive Partition 594
11.3 Boot from Inactive Partition Image - PCMM2G 595
11.3.0.1 Operational Image 595
11.3.0.2 Fallback Image 595
11.3.1 Boot Failure 595
11.3.2 Fallback Mode 595
11.3.2.1 Boot into Fallback Mode 595
11.3.3 Limitations 596
11.3.4 Recovery Mode 596
11.4 Boot Sequence 596
11.4.1 PCMM2G Boot Sequence 597
11.4.1.1 Internal Flash 597
11.4.1.2 Recovery USB Drive 597
11.5 Boot from Recovery Image (B2 Button) 598
11.6 Controller Boot Startup Script 598
11.6.1 PCMM2G Flow 598
11.7 Use the Controller Buttons 599
11.7.1 Button Functionality 600
11.7.1.1 Power-up 600
11.7.1.2 Running 601
11.7.2 B3 Menu Button 601
11.7.2.0.1 Access Functions Menu 601
11.7.2.1 Application is Running 601
11.7.2.2 Application is Not Running 601
11.7.3 View the Controller IP Address 602
11.7.4 Boot from Recovery Image (B2 Button) 602

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

11.7.5 Reset the Controller to Factory Settings ... 603
11.7.5.1 Factory Reset Changes ... 603
11.7.5.2 Reset Controller While the Drive is Running ... 604
11.7.6 Controller Backup and Restore ... 604
11.7.6.1 Backup ... 604
11.7.6.2 Restore ... 604
11.7.6.3 Data Transfer ... 605
11.7.6.4 B3 Button Backup ... 605
11.7.6.5 B3 Button Restore ... 606
11.7.7 Backup and Restore tab ... 606
11.7.7.1 EtherCAT Devices Backup ... 607
11.7.7.1.1 Required Items ... 607
11.7.7.1.2 Device Connection and Configuration ... 608
11.7.7.1.3 Modular EtherCAT ... 608
11.7.7.1.4 Backup Procedure ... 608
11.7.7.2 EtherCAT Devices Restore ... 609
11.7.7.3 AKD / AKD2G / Kollmorgen Essentials Drive Backup and Restore Compatibility ... 610
11.7.7.4 Controller Import / Export Backup from Web server ... 612
11.7.7.4.1 Controller Backup Import / Export ... 613
11.7.7.4.2 EtherCAT Devices Backup Import / Export ... 613
11.7.7.4.3 Export Procedure ... 613
11.7.7.4.4 Import Procedure ... 614
11.7.7.5 EtherCAT Devices Backup / Restore Limitations ... 614
11.7.7.5.1 Backup Conditions ... 614
11.7.7.6 EtherCAT Devices Backup / Restore Troubleshooting ... 614
11.8 Controller Onboard I/O ... 616
11.9 Using SSH ... 617
11.9.1 Login ... 617
11.9.2 SSH Login to a Controller ... 617
11.9.2.1 Login to PCMM2G ... 617
11.9.2.1.1 List of Common SSH Clients ... 618
11.9.3 Change Controller's Password using SSH ... 618
11.9.3.1 PCMM2G: Change Password ... 618
11.9.4 Restore Controller Default Password ... 618
11.9.4.1 Restore to Factory Settings ... 618
11.9.4.2 Restore PCMM2G ... 618
11.10 Using the KAS Web server ... 619
11.10.1 Access the Web server ... 619
11.10.1.1 Web Browser Access ... 619
11.10.2 KAS-IDE Access ... 619
11.10.3 Find the Controller IP Address ... 620
11.10.3.1 Links ... 621
11.10.3.2 Tabs ... 621
11.10.4 Web server Home page ... 621
11.10.4.1 Security ... 622
11.10.4.1.1 Backup and Restore ... 623
11.10.4.1.2 Diagnostics tab ... 623
11.10.4.1.3 KAS Application tab ... 623

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

11.10.4.1.4 Settings tab 623
11.10.4.1.5.1 PCMM2G Only 623
11.10.4.1.6 Timeout after Inactivity 623
11.10.4.2 User Authentication 623
11.10.4.2.1 Login In 623
11.10.4.2.2 Log Out 625
11.10.4.2.3 Changing the Password 625
11.10.5 KAS Application tab 625
11.10.5.1 Axis sub-tab 626
11.10.5.2 Log Configuration tab 627
11.10.5.2.1 Choose the Appropriate Level 628
11.10.5.2.2 Source 628
11.10.5.2.3 Level 628
11.10.5.3 Log Data tab 628
11.10.5.3.1.1 PCMM2G Only 630
11.10.5.3.2 Log Message Content 630
11.10.5.3.3 Controller Log Files 630
11.10.5.3.4.1 Log File Naming Convention 630
11.10.5.3.5.2 Error and Alarm History File (PCMM2G only) 631
11.10.5.4 User Data tab 631
11.10.5.4.1 View File Contents from the User Data File Explorer 633
11.10.5.4.2 Upload Folders and Files to the Controller 633
11.10.5.4.3 File Management Actions 634
11.10.5.4.4.1 Create Folder 634
11.10.5.4.5.2 Select All Files 634
11.10.5.4.6.3 Deselect All Files 634
11.10.5.4.7.4 Download Files 634
11.10.5.4.8.5 Delete 634
11.10.5.4.9.6 Clear User Data 634
11.10.5.4.10.7 Refresh 634
11.10.5.5 Shared Directory tab 634
11.10.5.6 Diagnostic Panels sub-tab 637
11.10.5.6.1 Diagnostic Panels - Home page 637
11.10.6 Settings tab 640
11.10.6.1 Certificates tab 641
11.10.6.1.1 SSL Server Certificate Details 641
11.10.6.1.2 Create SSL Server Certificate (Self-Signed) 642
11.10.6.1.3 Upload SSL Client Certificate 642
11.10.6.1.4.1 OPC UA Server Integration 642
11.10.6.1.5 SSL Client Certificates 642
11.10.6.1.6 Install the SSL Server Certificate (Self-Signed) 642
11.10.6.1.7.1 Certificate Expiration Period 643
11.10.6.1.8.2 Installation Procedure 643
11.10.7 Troubleshooting 662
11.10.7.1 Date/Time tab 662
11.10.7.1.1 Date/Time and NTP Status 664
11.10.7.1.2 Update Date/Time 664
11.10.7.1.3 Network Time Protocol (NTP) 665

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

11.10.7.1.4.1 Correct an A05 Alarm ... 665
11.10.7.1.5 Set Time Zone ... 665
11.10.7.2 File System tab ... 666
11.10.7.2.1 Reset the Controller to Factory Settings ... 666
11.10.7.2.2.1 Factory Reset Changes ... 666
11.10.7.3 Firmware tab ... 667
11.10.7.3.1 Upgrade the Controller's Firmware ... 668
11.10.7.3.2 View Firmware Information ... 671
11.10.7.3.3.1 File Types ... 672
11.10.7.3.4 Recover Using the Web server ... 672
11.10.7.3.5.1 PCMM2G Web server Recovery Mode ... 672
11.10.8 Recovery USB Flash Drive ... 672
11.10.9 Production Image Recovery ... 672
11.10.10 Step 1: Prepare the PCMM2G Recovery Image ... 673
11.10.11 Step 2: Flash the Image to USB Flash Drive ... 673
11.10.12 Step 3: Boot and Recover the PCMM2G ... 673
11.10.13 Customer Image Recovery ... 674
11.10.13.1 Network tab ... 675
11.10.13.1.1 Rotary Switch ... 675
11.10.13.1.2 Manually Change the IP Address ... 675
11.10.13.2 Security tab ... 676
11.10.13.2.1 Change the Password ... 677
11.10.13.2.2 SSH Connection ... 677
11.10.13.2.3 HTTP Configuration ... 678
11.10.13.2.4 KAS-IDE Configuration ... 678
11.10.13.3 USB Flash Drive tab ... 679
11.10.13.4 Change the User's Web server Account Password ... 679
11.10.13.4.1 I Forgot My Password ... 681
11.10.14 Backup and Restore from the Web server ... 682
11.10.14.1 All Controllers ... 682
11.10.14.2 PCMM2G Backup / Restore Only ... 682
11.10.14.2.1 Backup ... 682
11.10.14.2.2 Restore ... 682
11.10.14.3 Backup tab ... 683
11.10.14.3.1 PCMM2G Controller and EtherCAT Devices Backup ... 683
11.10.14.3.2.1 PCMM2G Backup / Restore Only ... 683
11.10.15 Backup ... 683
11.10.16 Restore ... 683
11.10.16.0.1 EtherCAT Devices Backup ... 685
11.10.16.0.2.1 Scenario 1 - Archived Firmware Copy = Firmware used on Drives ... 685
11.10.16.0.3.2 Scenario 2 - Archived Firmware Copy Not = Network Configuration ... 686
11.10.16.0.4.3 Scenario 3 - Only Unique ID Backed Up ... 686
11.10.16.1 Restore tab ... 686
11.10.16.1.1 Restore Controller ... 686
11.10.16.1.2 PCMM2G Backup / Restore Only ... 687
11.10.16.1.3.1 Backup ... 687
11.10.16.1.4.2 Restore ... 687
11.10.16.1.5 Restore EtherCAT Devices ... 687

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

11.10.16.1.6 Advanced ... 687
11.10.16.2 Import / Export tab ... 687
11.10.16.2.1 Controller ... 688
11.10.16.2.2 EtherCAT Devices ... 688
11.10.17 Diagnostics tab ... 688
11.10.17.1 Errors and Alarms sub-tab ... 688
11.10.17.2 Hardware Status sub-tab ... 689
11.10.17.3 Crash Reports sub-tab ... 691
11.10.17.4 EtherCAT Network Topology sub-tab ... 691
11.10.17.5 EtherCAT Diagnostics tab ... 693
12 Tools ... 695
12.1 Cam Profile Editor ... 695
12.1.1 Windows Overview ... 695
12.1.1.1 Splitters ... 696
12.1.2 CAM Profile Editor's CAM Table ... 696
12.1.2.1 Cam Editor - Table Parameters ... 697
12.1.2.2 Cyclic Cam Element ... 697
12.1.2.3 Use the Cam Table to Change an Element ... 698
12.1.2.3.1 Interpolation ... 698
12.1.2.4 Change the Type of a Cam Element ... 698
12.1.2.5 Cam Table Contextual Menu ... 698
12.1.2.6 Add a Point to the Cam Table ... 699
12.1.2.7 Remove a Point from the Cam Table ... 700
12.1.3 Cam Profile Graph ... 700
12.1.3.1 Modifying an Element ... 701
12.1.3.2 Change the Velocity of a Point ... 701
12.1.3.3 Cam Profile Graph Contextual Menu ... 701
12.1.3.4 Zoom In and Out ... 702
12.1.3.5 Panning ... 702
12.1.3.6 Restoring Zoom and Pan Settings ... 702
12.1.4 CAM Profile Segments ... 702
12.1.4.1 Line Segment Type ... 702
12.1.4.2 Parabolic Segment Type ... 703
12.1.4.3 Point Segment Type ... 703
12.1.4.4 Spline Segment Type ... 704
12.1.5 Curve Selection and Color Table ... 704
12.1.5.1 Change the Color ... 705
12.1.5.2 Curves Graph ... 705
12.1.6 Reload, Save, Auto Fit, and Properties Buttons ... 706
12.1.7 Import a CAM Profile ... 706
12.1.7.1 The Import ... 707
12.1.7.2 Displaying the Imported CAMProfile ... 708
12.1.7.3 Invalid Data ... 708
12.2 Custom Input/Output Editor ... 708
12.2.1 Add an Input/Output ... 708
12.2.2 Delete Input/Output ... 709
12.2.3 Modify an Input/Output ... 709
12.3 Human-Machine Interface Editor ... 709

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

12.3.1 Use the Kollmorgen Visualization Builder (KVB) ... 709
12.3.1.1 Create a New KVB Controller ... 710
12.3.1.2 Import Variables into the Project ... 712
12.3.1.3 Design the HMI Panel with KVB ... 714
12.3.1.3.1 Add Object ... 714
12.3.1.3.2 Customize Object ... 714
12.3.1.3.3 Map Variable to the Object ... 714
12.3.1.4 Download the KVB Panel ... 714
12.3.2 Design the HMI with the Internal Editor ... 715
12.3.2.1 Create a Control Panel ... 715
12.3.2.2 Use the Control Panel Control Library ... 717
12.3.2.3 Edit the Control Panel ... 718
12.3.2.3.1 HMI Grid Settings ... 718
12.3.2.4 Map Variables to the Control Panel ... 719
12.3.2.5 Graphic Objects ... 720
12.3.2.5.1 Basic Shapes ... 720
12.3.2.5.2 Bitmaps ... 721
12.3.2.5.3 Scales ... 721
12.3.2.5.4 Text boxes ... 722
12.3.2.5.5 Switches and 2-state displays ... 722
12.3.2.5.6 Analog buttons ... 723
12.3.2.5.7 Bar Graphs ... 723
12.3.2.5.8 Charts ... 724
12.3.2.5.9 Analog meters ... 724
12.3.2.5.10 Digital meters ... 725
12.3.2.5.11 Links ... 726
12.3.2.5.12 Connection status ... 726
12.3.2.5.13 Gauges ... 726
12.3.2.6 Graphic Objects Properties ... 727
12.3.2.7 Operate the Control Panel ... 732
12.3.2.7.1 KAS Simulator Display ... 733
12.3.2.7.2 Exiting Simulation Mode ... 733
12.4 Pipe Network Editor ... 733
12.4.1 Insert Pipe Blocks or Comments ... 734
12.4.2 Insert Pipe Network Connections ... 734
12.4.2.1 Relation Type for Output-Input is 1-n ... 734
12.4.3 Connect a Comment to a Pipe Block ... 735
12.4.4 Pipe Network Editor Controls ... 735
12.4.4.1 Basic Functions ... 735
12.4.4.2 Multiple Selections ... 736
12.4.5 Plug/Unplug Channels ... 737
12.5 Soft Oscilloscope ... 737
12.5.0.1 1. Control Panel ... 738
12.5.0.2 2. Graphical Area ... 738
12.5.1 Access the Soft Oscilloscope ... 738
12.5.2 About OpenGL ... 738
12.5.2.1 Verify the Graphic Card Driver is Up-to-date ... 738
12.5.2.2 Change the Settings of the Graphic Card ... 738

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

12.5.2.3 Display a Given Amount of Samples per the Refresh Rate ... 739
12.5.3 Oscilloscope Control Panel ... 739
12.5.3.1 1. Channel Group Selector ... 739
12.5.3.2 2. Import / Export Scope Channel Configuration buttons ... 741
12.5.3.3 3. Channels List area ... 741
12.5.3.3.1 Trace Times Channel Group ... 743
12.5.3.3.2 Simulator ... 743
12.5.3.4 4. Current Channel area ... 744
12.5.3.5 5. Time Base area ... 746
12.5.3.6 6. Time Position area ... 746
12.5.3.6.1 Set the Time-frame ... 748
12.5.3.7 7. Control Panel buttons ... 748
12.5.4 Oscilloscope Graphical Area ... 748
12.5.5 Traces ... 749
12.5.6 Plugging Probes, Mapping Variables ... 749
12.5.6.1 Map a Variable to the Oscilloscope ... 750
12.5.6.1.1 Un-map Variables from the Scope ... 751
12.5.6.2 Plug a Probe from the Dictionary ... 751
12.5.6.3 Plug a Probe from the Pipe Network ... 752
12.5.7 Setting Scale ... 753
12.5.7.0.1 Unit per Division ... 754
12.5.7.1 Time Scale ... 754
12.5.7.2 Variable Scale ... 754
12.5.8 Trace Zoom Feature ... 754
12.5.9 Import the Oscilloscope Data ... 755
12.5.10 Export the Oscilloscope Data ... 756
12.6 Practical Application: Using Trace Time To Measure CPU Load ... 756
12.6.0.1 PCMM2G ... 756
12.6.1 Oscilloscope Trace Times ... 757
12.6.1.1 Press the Start Button to Collect Some Data ... 758
12.6.2 Check the Peak Times ... 759
12.6.2.1 RealTimeMargin or PLCMargin Time ... 760
12.6.2.2 CycleJitter Trace ... 760
12.6.3 Example: PCMM2G Heavily Loaded CPU ... 760
12.6.3.1 1. MotionExec and PLCProgExec Times ... 761
12.6.3.2 2. PLCProgExec Peaks ... 761
12.6.3.3 3. PLCMarginTime ... 762
12.6.3.4 4. Controller Log ... 763
12.6.4 Example: PCMM2G Over Loaded CPU ... 763
12.6.4.1 1. MotionExec and PLCProgExec Times ... 764
12.6.4.2 2. RealTimeMargin ... 764
12.6.4.3 3. Controller Log ... 765
13 KAS Component Manuals ... 767
13.1 BBH FSoE Master ... 767
13.2 Controllers ... 767
13.2.1 Controller Differences ... 768
13.2.1.1 Hardware Differences ... 768
13.2.2 PCMM2G Hardware Specifications ... 769

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

13.2.3 Part Number Scheme ... 770
13.2.4 NVRAM ... 771
13.2.4.1 NVRAM Life Expectancy ... 771
13.2.4.1.1 Verify the NVRAM Space ... 771
13.3 Drives ... 772
13.3.1 Related Documents ... 772
13.3.2 AKD-M Kx PDMM Controller / Drive Variant ... 772
13.3.2.1 FAQs for the Kx Controller / Drive ... 773
13.3.3 AKD-N "ES" Drive Variant ... 773
13.3.3.1 FAQs for the "ES" Drive ... 773
13.3.4 AKD-P "NC" Drive Variant ... 774
13.3.4.1 FAQs for the "NC" Drive ... 774
13.3.5 AKD-P "ND" Drive Variant ... 775
13.3.5.1 FAQs for the "ND" Drive ... 775
13.3.6 MKD-C "ND" Central Power Supply Variant ... 775
13.3.6.1 FAQs for the "ND" Central Power Supply ... 775
13.4 HMI ... 776
13.4.1 HMI Accessories ... 776
13.5 Remote Input/Output Terminals ... 777
13.6 Safety Solutions ... 779
13.6.1 KSM-compact Safety Module ... 779
13.6.2 KSM-modular Safety PLC ... 780
13.6.3 Safety Functions (PLe & SIL3) ... 780
13.6.4 Connectivity ... 780
14 Troubleshooting ... 781
14.1 FAQs ... 781
14.1.1 Basic ... 781
14.1.2 Computer / Controller ... 786
14.1.3 EtherCAT ... 788
14.1.4 KAS-IDE Crashes ... 790
14.1.5 Performance ... 790
14.1.5.1 Resolution ... 792
14.1.6 Programming ... 792
14.1.6.1 Primary Feedback ... 793
14.1.7 Setup ... 794
14.1.7.1 Description ... 794
14.1.7.2 Firewall ... 795
14.1.7.3 Port Numbers ... 795
14.2 Compiler Errors ... 795
14.2.1 Compile Optimized Code FAILED ... 796
14.2.2 Compiling Optimized Code Fails due to User Permissions ... 796
14.2.3 EtherCAT: ERROR: Failed to retrieve Vendor... ... 797
14.2.3.1 EtherCAT: ERROR: Failed to retrieve EtherCAT device... ... 797
14.2.4 EtherCAT: ERROR: In AKD_1, PDO object index 0x6040, subIndex 0 is redundant. ... 798
14.2.4.1 Failed to export ENI file, check the logs for more details. ... 798
14.2.4.2 EtherCAT: Warning : In AKD_1, PDO object index 0x3470, subIndex 4 is redundant. ... 798
14.2.5 Unexpected complex data in expression. ... 798
14.2.5.1 Boolean expression expected after IF. ... 798

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

14.3 Compiler Warnings ... 800
14.3.1 10001 - Macro Redefinition ... 801
14.3.2 10002 - Not IEC Compliant ... 801
14.3.3 10003 - Same Name to a Global Variable and a Local Variable ... 801
14.3.4 10014 - Empty Line Ignored in Signal Resource (No Time Defined) ... 801
14.3.5 10015 - On Line Change may not be Possible: Too Many POUs or Variables ... 801
14.3.6 10019 - Definition of the Block was Changed ... 801
14.3.7 10020 - Backward Jump may Block the Target Cycle ... 801
14.3.8 10021 - FB Instance Missing (Unsafe) ... 801
14.3.9 10022 - Output Used Before it is Set. Execution may be Unsafe ... 802
14.3.10 10023 - Loop Instructions may Block the Target Cycle ... 802
14.3.11 10024 - Implicit Loops may be Unsafe ... 802
14.3.12 10025 - Input Parameters are not Considered as IN/OUT ... 802
14.3.13 10026 - Backward Jumps may Block the Cycle ... 802
14.3.14 10030 - Invalid PRINTF Format ... 802
14.3.15 10033 - LREAL# Prefix is Required to Avoid Loss of Accuracy ... 803
14.3.16 10034 - Loop Instructions may Block the Target Cycle ... 803
14.3.17 10037 - The Same Modbus Address is Used for Several Variables ... 803
14.4 Connect Remotely ... 803
14.5 CPU Load Reduction Techniques ... 804
14.6 EtherCAT Coupler Error Handling and Diagnosis ... 804
14.6.1 EtherCAT Diagnostic LEDs ... 804
14.6.2 EtherCAT LEDs - Fieldbus Connection Diagnosis ... 805
14.6.3 EtherCAT LED - Off Power Supply Diagnosis ... 805
14.6.4 EtherCAT LED - Power Supply Diagnosis ... 805
14.6.5 EtherCAT LEDs - Standard Data Diagnosis ... 805
14.6.5.1 I/O Errors ... 806
14.6.6 EtherCAT LEDs - State Machine / PLC Diagnosis ... 807
14.7 EtherCAT Diagnostics and Errors ... 807
14.7.1 EtherCAT Diagnostics ... 807
14.7.2 EtherCAT Error Messages ... 808
14.7.2.1 Error Message Causes ... 808
14.7.2.1.1 Wrong / Missing Device ... 808
14.7.2.1.2 Link Loss / Device Fault ... 809
14.7.2.1.3 Frame Loss ... 809
14.7.2.1.4 Working Counter ... 809
14.7.2.2 Other Messages Linked to EtherCAT ... 810
14.7.3 Example: EtherCAT Communication Diagnosis Steps ... 810
14.7.3.1 Diagnosing EtherCAT Communication Errors ... 810
14.7.3.1.1 Code Examples for Diagnosing EtherCAT Communication Errors ... 811
14.8 Re-import Kollmorgen UDFBs ... 812
14.9 Troubleshoot FSoE Safety Parameters ... 813
14.10 Troubleshoot KAS-IDE and SafePLC2 ... 816
15 Advanced Topics ... 820
15.1 Coordinated Motion ... 820
15.1.1 Group State Diagrams ... 820
15.1.2 Coordinate Systems ... 820
15.1.3 Design Coordinated Motion ... 821

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.1.3.1 Blending Between Moves 822
15.1.3.2 Blending with Transitions 823
15.1.3.3 Create a Linear or Circular Coordinated Motion Application 824
15.1.3.3.1.1 Groups 824
15.1.3.3.2 Function Block Calls 824
15.1.3.3.3 What are Axes Groups? 827
15.1.3.3.4 Perform a Circular Move 827
15.1.3.3.5.1 Absolute Circular Move 827
15.1.3.3.6.2 Relative Circular Move 828
15.1.3.3.7.3 Circular Moves Diagrams 829
15.2 CircMode = BORDER 829
15.3 CircMode = CENTER 829
15.3.0.0.1.1 Precision Requirements for Circular Move Input Parameters 830
15.4 Parameter Validation 830
15.5 Default Validation Method 830
15.6 Validation with Custom Tolerance 831
15.6.0.0.1.1 Perform a Complete Circular Move 831
15.6.0.0.2 Perform a Linear Move 831
15.6.0.0.3.1 Absolute Linear Move 832
15.6.0.0.4.2 Relative Linear Move 832
15.6.0.0.5.3 Linear Move With More Than Two Axes 833
15.6.0.1 Differences between MC_GrpHalt and MC_GrpStop 833
15.6.0.1.1 What MC_GrpHalt Does 834
15.6.1 Exceptions 835
15.6.1.0.1 What MC_GrpStop Does 836
15.6.2 Exceptions 838
15.6.2.1 Handling Axis Errors 838
15.6.2.1.1 Optional Behavior 839
15.6.2.1.2 Default and Optional Error Handling Behavior 839
15.6.2.1.3.1 Default Error Handling Behavior 839
15.6.2.1.4.2 Optional Error Handling Behavior 839
15.6.2.1.5 Recovery of the System State After an Axis Error 840
15.6.2.2 Transition Between Moves 840
15.6.2.2.1.1 Transition Mode Parameters 841
15.6.2.2.2 No Transition (TM_NONE) 841
15.6.2.2.3 Corner Distance (TM_CORNER_DISTANCE) 841
15.6.2.2.4.1 Transition Parameter Index 842
15.6.2.2.5 S-Curve Corner (TM_SCURVE_CORNER) 842
15.6.2.2.6.1 Transition Parameter Index 842
15.6.2.2.7 Arc-to-Arc Transitions 843
15.6.2.2.8.1 Intersection Transition 843
15.6.2.2.9.2 Line Segment Transition 844
15.6.2.2.10.3 Same Circle: Opposite Direction Transition 844
15.6.3 n-Degree: Arcs in Opposite Direction 844
15.6.3.0.1.1 Same Circle: Same Direction Transition 845
15.6.4 n-Degree: Arcs in Same Direction 845
15.6.4.0.1.1 Tangent Transition: Opposite Direction 845
15.6.4.0.2.2 Zero-distance Transition 846

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.6.4.0.3 Line-to-Arc and Arc-to-Line Transitions 846
15.6.5 n-Degree Transition 846
15.6.5.0.1.1 Special Case Transitions 846
15.6.6 Intersection Transition 847
15.6.7 Tangent Transition 847
15.6.8 Zero-distance Transition 847
15.6.8.0.1 Line-to-Line Transitions 848
15.6.8.0.2.1 n-Degree: Transition 848
15.6.8.0.3.2 Special Case Transitions 848
15.6.9 180-degree Transition 848
15.6.10 Zero-degree Transition 849
15.6.11 Zero-distance Transition 849
15.6.11.0.1 S-Curve Corner Line-to-Line Transitions 849
15.6.11.0.2.1 n-Degree: Transition 849
15.6.11.0.3.2 Special Case Transitions 850
15.6.12 180-degree Transition 850
15.6.13 Zero-degree Transition 850
15.6.14 Zero-distance Transition 851
15.7 Motion Techniques 851
15.7.1 Create a Digitizing Axis 851
15.7.1.1 Create a Digitizing Axis From Code 852
15.7.1.2 Create a Digitizing Axis with an Imported ENI File 852
15.7.1.3 Create a Digitizing Axis When EtherCAT Devices Are Defined 853
15.7.1.4 Create a Digitizing Axis When EtherCAT Devices Are Not Connected 854
15.7.1.5 Digitizing Axis - PLCopen Setup 855
15.7.1.5.1 First Generation AKD Drives 856
15.7.1.6 Digitizing Axis - Encoders without a Z-Channel 857
15.7.2 Error Management 857
15.7.2.1 Axis Procedures 857
15.7.2.2 Machine Procedures 858
15.7.3 Fast Inputs 858
15.7.3.1 Fast Inputs with Pipe Network Motion 859
15.7.3.2 Fast Inputs with PLCopen Motion 859
15.7.4 Interpret a Timestamp 859
15.7.5 Overtravel Conditions 860
15.7.5.0.1 Configuring Generation 1 AKD Drives 860
15.7.5.1 Coordinated Motion Behavior 861
15.7.5.2 Pipe Network Behavior 861
15.7.5.3 PLCopen Behavior 861
15.7.6 PC Online Change 861
15.7.6.1 What is Online Change 862
15.7.6.2 Activate an Online Change 864
15.7.6.2.1 Set the Parameters 864
15.7.6.2.2 Switch to Edit Mode 865
15.7.6.2.3 Dictionary Behavior 865
15.7.6.3 The Revert Button 866
15.7.6.4 Difference between Local and Controller Versions 866
15.7.6.5 Pulse Limitations with Online Change 866

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.7.6.6 Use the PLC Online Change ... 867
15.7.6.6.1 Set an Application ... 867
15.7.6.6.2 Enable Online Change ... 867
15.7.6.6.3 Use the Online Change ... 868
15.7.6.6.4 Revert the Online Change ... 869
15.7.7 PLCopen Homing ... 869
15.7.7.1 PLCopen Homing Description ... 869
15.7.7.1.1 Utilizing MC_Reference ... 870
15.7.7.1.2 Writing your Own Homing Cycles ... 870
15.7.7.1.3 Utilizing Custom Libraries ... 870
15.7.7.2 PLCopen Homing Methods ... 870
15.7.7.2.1 Home using the Current Position ... 870
15.7.7.2.2 Find Input ... 870
15.7.7.2.3 Find Input then Find Zero Angle ... 870
15.7.7.2.4 Find Input then Find Index ... 871
15.7.7.2.5 Find Index ... 871
15.7.7.3 AKD Capture Engine Configuration ... 871
15.7.7.3.1 AKD Configuration Parameters ... 872
15.7.7.4 Example: Fast Homing with the PLCopen Motion Engine ... 873
15.7.8 Pipe Network Homing ... 874
15.7.8.1 Example: Fast Homing with the Pipe Network Motion Engine Axis Pipe Block ... 874
15.7.9 PLCopen Registration ... 875
15.7.9.1 Decide which Function Blocks to Use for Registration ... 875
15.7.9.2 Master/Slave Registration ... 877
15.7.9.2.1 Master Registration ... 877
15.7.9.2.2 Slave Registration ... 877
15.7.9.3 Single-Axis Registration ... 878
15.7.9.4 Registration Application Guide ... 878
15.7.9.4.1 Mark to Mark Registration ... 879
15.7.9.4.2 Mark to Machine Registration ... 879
15.7.10 Pipe Network Registration and Fast Homing ... 880
15.7.10.1 Example: Fast Homing with the Pipe Network Motion Engine Axis Pipe Block ... 880
15.7.10.2 Example: Registration Position Capture with Pipe Network Trigger Block ... 881
15.7.10.2.1 Configuration of the Trigger Block ... 882
15.7.11 Practical Applications - Camming ... 882
15.7.11.1 Cam on the Fly ... 883
15.7.11.2 Position an Axis Before Starting Camming ... 883
15.7.11.3 Resume Camming After an E-Stop ... 884
15.7.12 Restarting Motion with Pipe Network ... 884
15.7.13 Superimposed Axes with PLCopen ... 885
15.7.14 Torque Feed-forward ... 885
15.7.15 Tuning Controller Performance for EtherCAT Communication Latency ... 885
15.8 Motion Bus and Fieldbuses ... 886
15.8.1 EtherCAT ... 887
15.8.2 EtherNet/IP ... 887
15.8.3 Modbus and TCP/IP ... 887
15.8.4 PROFINET ... 888
15.8.5 OPC UA ... 888

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.8.6 I/O Mapping (for PROFIBUS Fieldbus) ... 888
15.8.7 Add an Unsupported EtherCAT Device ... 889
15.8.7.1 Modify the EtherCAT Image in Cyclic Mode ... 890
15.8.7.2 Configure the EtherCAT Device ... 890
15.8.7.3 Map PLC Variables ... 890
15.8.7.3.1 Examples ... 890
15.8.8 Set Up FSoE Master and an AKD2G with SafeMotion Monitor ... 891
15.8.8.1 1. Create Safety Application ... 891
15.8.8.2 2. KAS-IDE Project Set Up ... 891
15.8.9 AKD2G Safety Parametrization Using FSoE with SCU-1-EC and PxMM / PCMM2G ... 893
15.8.10 Fieldbus Editor ... 893
15.8.10.1 EtherNet/IP Controller Data Communications ... 893
15.8.10.2 EtherNet/IP Controller EDS File ... 894
15.8.10.2.1 Using EDS Files ... 894
15.8.10.3 EtherNet/IP Adapter (Server) ... 895
15.8.10.3.1 Data Exchange - Configuration ... 895
15.8.10.3.2 Configuration ... 896
15.8.10.4 EtherNet/IP FlexIO / PointIO ... 897
15.8.10.5 EtherNet/IP IO Scanner (Client) ... 898
15.8.10.5.1 Data Exchange Configuration ... 898
15.8.10.5.2 Server Configuration ... 898
15.8.10.5.3 Edit the Configuration Data Block ... 899
15.8.10.5.4 Assembly Configuration ... 899
15.8.10.5.5 Configuring Variables ... 900
15.8.10.6 EtherNet/IP Tag Client ... 901
15.8.10.6.1 Data exchange - Configuration ... 901
15.8.10.6.2 Configuration ... 901
15.8.10.7 Link with Embedded HTTP Server ... 903
15.8.10.7.1 Link with embedded HTTP Server dialog box ... 904
15.8.10.8 Modbus Slave ... 907
15.8.10.8.1 Protocol Specification ... 908
15.8.10.8.2 Data Exchange Configuration ... 908
15.8.10.8.3 Modbus Slave Configuration ... 909
15.8.10.8.4 Data Types ... 910
15.8.10.9 PROFINET Configuration ... 910
15.8.10.9.1 Configure a PROFINET IO Controller ... 910
15.8.10.9.2 Configure a PROFINET IO Device ... 915
15.8.10.9.3 PROFINET - Additional Features ... 920
15.8.10.9.4.1 All PROFINET Equipment ... 920
15.8.11 Create IOxS for Slave Modules ... 920
15.8.11.1 Input Modules or Modules without IO Data ... 921
15.8.11.2 Output Modules ... 921
15.8.12 Create Variables ... 921
15.8.13 Device Diagnosis ... 921
15.8.13.0.1.1 PROFINET Controllers Only ... 921
15.8.14 Browse Network for Slaves ... 921
15.8.15 Device Configuration ... 922
15.8.16 Read Module Configuration ... 922

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.8.17 Set Slave Station Name ... 922
15.8.17.0.1 PROFINET - Data Types ... 922
15.8.17.0.2 PROFINET - Error and Status Codes ... 922
15.8.17.0.3.1 Error Codes and Meanings ... 923
15.8.17.0.4.2 PNIO Status Codes - Negative Responses ... 923
15.8.17.0.5.3 Error Code - Negative Responses ... 923
15.8.17.0.6.4 Error Decode - Negative Responses ... 923
15.8.17.0.7.5 Error Code 1 for Error Decode = 80 ... 923
15.8.17.0.8.6 Error Code 1 for Error Decode = 81 ... 924
15.8.17.0.9.7 Error Code 2 for Error Code 1 = 40 ... 925
15.8.17.0.10.8 Error Code 2 for Error Code 1 <> 40 ... 925
15.8.17.0.11.9 Error Code 2 for Error Code 1 = 01 ... 926
15.8.17.0.12.10 Error Code 2 for Error Code 1 = 02 ... 926
15.8.17.0.13.11 Error Code 2 for Error Code 1 = 03 ... 927
15.8.17.0.14.12 Error Code 2 for Error Code 1 = 04 ... 927
15.8.17.0.15.13 Error Code 2 for Error Code 1 = 8 ... 927
15.8.17.0.16.14 Error Code 2 for Error Code 1 = 16 ... 928
15.8.17.0.17 PROFINET - Resolve Errors ... 928
15.8.17.0.18.1 Additional Analyses - Controllers Only ... 928
15.8.17.0.19.2 CL-RPC Lookup - Devices Only ... 928
15.8.18 Timeout Error ... 928
15.8.18.0.1.1 Connect Response Error ... 929
15.8.19 Timeout Error ... 929
15.8.19.0.1.1 Device is not Found ... 929
15.8.19.0.2.2 Error Setting the IP Configuration ... 929
15.8.19.0.3.3 Module Configuration is Different ... 929
15.8.19.0.4.4 Other Errors ... 929
15.8.19.0.5.5 Writing Parameterization Error (with status 0xDF80*) ... 930
15.8.20 Use Wireshark ... 930
15.8.20.1 OPC UA Server ... 930
15.8.20.1.1 OPC UA Server Overview ... 930
15.8.20.1.2.1 Key Features ... 930
15.8.20.1.3.2 OPC UA Server Configuration ... 931
15.8.20.1.4.3 OPC UA Communications using External Software ... 931
15.8.20.1.5 OPC UA - Fieldbus Configuration ... 931
15.8.20.1.6.1 OPC UA - Create New Driver Configuration ... 931
15.8.20.1.7.2 OPC UA - Insert an Endpoint ... 934
15.8.20.1.8.3 OPC UA - Insert a Group ... 936
15.8.20.1.9.4 OPC UA - Insert a Variable ... 938
15.8.20.1.10.5 OPC UA - Add Multiple Groups and Variables to the Driver Configuration ... 941
15.8.20.1.11.6 OPC UA - Move Variables ... 945
15.8.20.1.12.7 OPC UA - Remove Variables ... 946
15.8.20.1.13 OPC UA - Additional Features ... 947
15.8.20.1.14 OPC UA - Troubleshooting ... 948
15.8.20.1.15 OPC UA - Add TAG(s)* dialog box ... 948
15.8.20.1.16.1 Add TAG(s)* dialog box buttons ... 950
15.8.20.1.17 OPC UA - Endpoint dialog box ... 953
15.8.20.1.18 OPC UA - Group: dialog box ... 955

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.8.20.1.19 OPC UA - Insert Variable dialog box ... 957
15.8.20.1.20 OPC UA - Server dialog box ... 960
15.8.20.2 OPC UA - External Software Communications ... 964
15.8.20.2.1 OPC UA - Establish a Connection using UaExpert ... 965
15.8.20.2.2.1 OPC-UA - UaExpert Installation ... 965
15.8.20.2.3.2 OPC-UA - Establish a Server Connection with Certificates - UaExpert ... 965
15.9 1. PCMM2G Configuration (with Certificates) ... 965
15.10 2. UaExpert Configuration (with Certificates) ... 968
15.11 3. UaExpert Connection to the PCMM2G (with Certificates) ... 977
15.11.0.0.1.1 OPC-UA - Establish a Server Connection without Certificates - UaExpert ... 980
15.12 1. PCMM2G Configuration (without Certificates) ... 981
15.13 2. UaExpert Configuration (without Certificates) ... 983
15.14 3. UaExpert Connection to the PCMM2G (without Certificates) ... 989
15.14.0.0.1 OPC UA - Establish a Connection using Exor HMI ... 992
15.14.0.0.2.1 OPC UA - Exor HMI Installation ... 992
15.14.0.0.3.2 OPC UA - Establish Server Connection without Certificates - Exor HMI ... 993
15.14.1 Modular EtherCAT Concept ... 1000
15.14.1.1 Examples of Modular Machine Design Uses ... 1001
15.14.1.1.1 Machine with Multiple Variants ... 1001
15.14.1.1.2 Module Maintenance ... 1001
15.14.1.1.3 Share Modules across Machines in a Factory Line ... 1001
15.14.1.2 Modular Design Guidelines ... 1001
15.14.1.3 Backing up Modular EtherCAT Devices ... 1002
15.14.1.4 Limitations ... 1002
15.14.1.5 Create an Application with Modular EtherCAT ... 1003
15.14.1.5.1 Create a Machine Variant with the Basic Machine ... 1003
15.14.1.5.2 ST Language Example ... 1004
15.14.1.5.3 Add a Module ... 1004
15.14.1.5.4 ST Language Example ... 1004
15.14.1.5.5 Remove a Module ... 1005
15.14.1.5.6 ST Language Example ... 1005
15.14.2 Controller Performance Guidelines ... 1006
15.14.2.1 EtherNet/IP and PROFINET ... 1006
15.14.2.1.1 PCMM2G ... 1006
15.14.2.1.2 Example: EtherNet/IP Performance ... 1006
15.14.2.2 OPC UA ... 1007
15.14.2.2.1 Recommendations to Reduce Missed Receive Frames and PLC Cycles ... 1007
15.15 Project Structure Guidelines ... 1007
15.15.1 External Files ... 1007
15.15.2 Application Software Structure - Definitions ... 1008
15.15.2.1 Modules to Build up the Structure ... 1008
15.15.2.1.1 Structure Overview ... 1008
15.15.2.1.2 Main Module ... 1009
15.15.2.1.3.1 Module Controller ... 1009
15.15.2.1.4.2 Functional Part ... 1010
15.15.2.1.5 Axis Module ... 1010
15.15.2.1.6.1 Axis Module Controller ... 1010
15.15.2.2 State and Function Definitions ... 1010

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.15.2.2.1 State, State Transitions and Functions 1011
15.15.3 Application Software Structure - Implementation 1011
15.15.3.1 SFC Children Building up the Software 1011
15.15.3.1.1 Parent SFC 1011
15.15.3.1.2 Main Module SFC Children 1012
15.15.3.1.3 Axis Module SFC Children 1012
15.15.3.2 Functions Linked to States 1012
15.15.3.2.1 Functions Treatment 1012
15.15.3.2.2.1 Function (Function Step) 1012
15.15.3.2.3 Add a New Function 1013
15.15.3.3 Main Module Implementation 1013
15.15.3.3.1 bErrorReset 1014
15.15.3.3.2 M1_AckState 1014
15.15.3.3.3 M1_CmdState 1014
15.15.3.3.4 M1_ErrorHandling 1014
15.15.3.3.5 M1_ModStatus, (Ai_ModStatus) 1015
15.15.3.3.6 M1_ModuleController 1015
15.15.3.3.7.1 M1_ReqState 1016
15.15.3.4 States and Errors 1016
15.15.3.4.1 States and Errors Treatment 1016
15.15.3.4.2 Add a New State 1017
15.15.3.5 Variables for the Interface 1017
15.15.3.5.1 List of Variables 1017
15.15.3.5.2 List of Output Variables 1017
15.16 Project Templates 1018
15.16.1 Coordinated Motion 1018
15.16.2 KAS Runtime 1018
15.16.3 KDN Download 1018
15.16.4 Pipe Network 1018
15.16.5 PLCopen 1019
15.16.6 Template Inclusions 1019
15.16.7 Coordinated Motion 1020
15.16.7.1 Coordinated Motion - 2-Axis Template 1020
15.16.7.1.1 Programs 1020
15.16.7.1.2 Motion Procedure 1020
15.16.7.1.3 Control Panel 1020
15.16.7.2 Coordinated Motion - 3-Axis Template 1021
15.16.7.2.1 PLC Programs 1021
15.16.7.2.2 Motion Procedure 1021
15.16.7.2.3 Control Panel 1022
15.16.7.3 Coordinated Motion - 3-Axis (3D) Template 1022
15.16.7.3.1 PLC Programs 1023
15.16.7.3.2 Motion Pattern Procedure 1023
15.16.7.3.3 Control Panel 1023
15.16.8 Pipe Network Axis Templates 1024
15.16.8.1 Pipe Network 1-Axis Template - FFLD 1025
15.16.8.2 Pipe Network 1-Axis Template - ST 1025
15.16.8.3 Pipe Network 2-Axes Template - FBD, FFLD, SFC, and ST 1026

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.16.8.3.1 Control Panel ... 1026
15.16.8.3.2 HMI ... 1027
15.16.8.3.3 Motion ... 1028
15.16.8.3.4 PLC Programs ... 1028
15.16.8.4 Pipe Network 2-Axes Template - ST ... 1031
15.16.8.4.1 Control Panel ... 1031
15.16.8.4.2 HMI ... 1032
15.16.8.4.3 Motion ... 1033
15.16.8.4.4 PLC Programs ... 1033
15.16.8.5 Pipe Network 2-Axes Template - FFLD ... 1034
15.16.8.5.1 Control Panel ... 1034
15.16.8.5.2 HMI ... 1035
15.16.8.5.3 Motion ... 1036
15.16.8.5.4 PLC Programs ... 1036
15.16.9 PLCopen Axis Templates ... 1037
15.16.9.1 PLCopen1-Axis Template - FFLD ... 1037
15.16.9.2 PLCopen 1-Axis Template - ST ... 1038
15.16.9.3 PLCopen 2-Axes Template - FFLD and SFC ... 1039
15.16.9.3.1 PLC Programs ... 1039
15.16.9.3.2.1 Control Panel Screens ... 1041
15.16.9.3.3.2 Control Panel Sections ... 1041
15.16.9.4 PLCopen 2-Axes Template - ST ... 1043
15.16.9.4.1 PLC Programs ... 1043
15.16.9.4.2.1 Control Panel Screens ... 1044
15.16.9.4.3.2 Control Panel Sections ... 1044
15.16.9.5 PLCopen 2-Axes Template - FFLD ... 1047
15.16.9.5.1 Control Panel ... 1048
15.16.9.5.2.1 Control Panel Screens ... 1048
15.16.9.5.3.2 Control Panel Sections ... 1048
15.16.9.5.4 HMI ... 1049
15.16.9.5.5 Motion ... 1050
15.16.9.5.6 PLC Programs ... 1051
16 Appendices: Lists of Manuals, Content, and Support Information ... 1052
16.1 PCMM2G - File Naming Conventions ... 1052
16.1.1 PCMM2G Firmware Files ... 1052
16.1.2 PCMM2G Operating Source (OS) and License Files ... 1052
16.1.3 Operating System (OS) Build Sources - File Naming Convention ... 1052
16.1.3.1 File Contents ... 1052
16.1.3.2 Usage ... 1052
16.1.3.3 File Format and Example ... 1052
16.1.4 Operating System (OS) License Sources - File Naming Convention ... 1053
16.1.4.1 File Contents ... 1053
16.1.4.2 Usage ... 1053
16.1.4.3 File Format and Example ... 1053
16.1.5 Operating System (OS) + KAS Runtime (RT) - File Naming Convention ... 1053
16.1.5.1 File Contents ... 1053
16.1.5.2 Usage ... 1053
16.1.5.3 File Format and Example ... 1053

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

16.1.5.4 Model Code ... 1054
16.1.6 Recovery Operating System (OS) + KAS Runtime (RT) - File Naming Convention ... 1054
16.1.6.1 File Contents ... 1054
16.1.6.2 Usage ... 1054
16.1.6.2.1 Reset Changes ... 1054
16.1.6.3 File Format and Example ... 1055
16.1.6.4 Model Code ... 1055
16.1.7 Boot Loader - File Naming Convention ... 1056
16.1.7.1 File Contents ... 1056
16.1.7.2 Usage ... 1056
16.1.7.3 File Format and Example ... 1056
16.1.7.4 Model Code ... 1056
16.1.8 KAS Runtime (RT) Only - File Naming Convention ... 1056
16.1.8.1 File Contents ... 1056
16.1.8.2 Usage ... 1057
16.1.8.3 File Format and Example ... 1057
16.1.8.4 Model Code ... 1057
17 List of Documentation ... 1058
17.1 AKD, AKD2G, S300, and S700 Documentation ... 1058
17.2 HMI Documentation ... 1058
17.3 IO Documentation ... 1059
17.3.1 AKT (K-Bus) Terminals ... 1059
17.3.2 AKT2G-IO (EtherCAT) Terminals ... 1060
17.4 KAS Documentation ... 1061
17.5 KVB Documentation ... 1062
17.6 PCMM2G Documentation ... 1063
18 List of How-Tos ... 1064
18.1 Advanced Motion How-Tos ... 1064
18.2 EtherCAT Fieldbus How-Tos ... 1064
18.3 Hardware How-Tos ... 1064
18.4 PLC Code How-Tos ... 1064
18.5 Run the Application How-Tos ... 1064
19 Kollmorgen Documentation Legal Information ... 1065
19.1 KAS - Copyrights, Trademarks, Patents, Licenses, and Disclaimers ... 1066
19.2 Kollmorgen Trademarks ... 1066
19.3 Kollmorgen Patents ... 1066
19.4 PCMM2G ... 1067
19.5 Other Trademarks ... 1068
19.6 3rd-party Licenses ... 1069
19.7 Disclaimers ... 1071
Support and Services ... 1073

Record of Document Revisions

Edition	Software Version	Date	Description
JJ	5.00	July 2026	PCMM2G Axis Scalable Models Support, Recipe Management using Function Blocks, Action Blocks Support for PLC Programs, PCMM2G Optimized Code Generation, PCMM to PCMM2G Migration Support Improvements, KAS-IDE Windows Scaling Improvements on High-Resolution Monitors, Secure KAS-IDE Connections to KAS / PCMM2G Runtime, Copy Retained variable runtime values for arrays and structures, PLC cycle load monitor and diagnostics support via function blocks, PxMM End-Of-Life Changes: Removed AKD PDMM / PCMM controller selection for KAS projects.
HH	4.05 R1	May 2026	Bug fixes.
GG	4.05	February 2026	OPC UA Server, Web Diagnostic Panels, Single-Core Compatibility Mode for PCMM2G Migration, Kollmorgen Essentials Drive Integration, MKD-N Drive Integration, KAS-IDE-WorkBench Express Setup View, EtherCAT PDO Data Read/Write Function Blocks. EtherCAT PDO Mapping to Array Variables, PCMM2G OS Security and Stability Enhancements.
FF	4.04 R2	December 2025	Added MKD-N Drive Integration Support.
EE	4.04 R1	October 2025	Kollmorgen Essentials initial implementation.
DD	4.04	June 2025	Installer improvements, non-admin installer option, PDO selection mapping, import oscilloscope trace file, export and import CoE Init-Commands.
CC	4.03 R1	January 2025	Upload of KAS-IDE crash report files to Kollmorgen server, coordinated motion engine now correctly handles rollover.
BB	4.03	December 2024	PCMM2G and KAS-IDE security enhancements, AKD2G limit switch handling, digitizing axis multi-turn encoders support, oscilloscope enhancements.
AA	4.02 R1	October 2024	KAS-IDE compiler now correctly checks the firmware version of the AKD devices, controller now tracks the actual feedback position of the axis, PCMM2G runtime has been updated to properly initialize UDP communications, MC_Power state machine updated to abort the drive power-on sequence, digitizing axes accurately report the actual position, exported file includes data for all the traces present in the Channels List, default single-turn position bit count is 31.
Y	4.02	June 2024	Release of PCMM2G.
W	4.01	December 2023	Updated documentation to support PCMM2G.
V	4.00	December 2022	Add compatibility for PCMM2G; Remove obsolete File Management functions.
U	3.07	December 2021	Modular EtherCAT topology, AKT2G-IO-BRC-000 Brake Chopper, saving previous versions of KAS files, link to templates on KDN.
T	3.06	June 2021	Integration of stepper I/O slices and support for stepper motors. Added ECATDevReadParam, MCFB_DriveFault, MLFB_DriveFault, MCFB_ECATRestart, and MLFB_ECATRestart. Access to Bits of an Integer increased to 64-bits.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Edition	Software Version	Date	Description
R	3.05	December 2020	Integration of BBH SCU-1-EC with KAS-IDE and BBH SafePLC2 software; Integration of E-bus AKT2G-IO I/O devices; WorkBench connection improvements; MLAxisReadVel.
P	3.04	June 2020	TCP/IP functions, eitadapter, file management functions, SafePLC support, WorkBench improvements, support AKT2G-IO slices, DriveParamStrRead, multi-select PDO objects for mapping.
N	3.03	December 2019	FSoE; T-Bot / H-Bot / Gantry robotics: Axes Groups, MC_KIN_REF structure, MC_SetKinTra; AKD2G Backup and Restore, MDP support.
M	3.02	June 2019	Map a digitizing axis, AKD2G, multiple instances of KAS-IDE, ElasticSearch, reorder programs.
L	3.01	December 2018	Multiple installed versions, updated UDFBs, Backup Import/Export on webserver, UI improvements, Softscope enhancements, security enhancements.
K	3.0	January 2018	AKD 1.17, AKD2G integration, MC_CreatePLCAxis, MLPNAxisCreate.
J	2.12	June 2017	AKD2G support, AKD 1.16, KVB 2.20, AKIX2 panels, EtherCAT Diagnostics (webserver diagnostics tab, A31, ECATWCStatus, ECATMasterStatus, ECATDeviceStatus, ECATCommErrors), MC_ MoveContVel, FFLD Editor Comments, PN Editor Improvements, SSH Security.
H	2.11	September 2016	Parabolic cam profile support, send data to/from a controller via UDP, access KDN from the menu, improved EtherCAT diagnostics, Find All command, find variable instances, access shared directories from the webserver, configurable Modbus slave data exchange rate, double-click editing of FBD and FFLD, F_Seek function, configurable circle center point tolerance.
G	2.10	January 2016	PCMM support, Code generation types: Normal and Optimized, MC_ TouchProbe: Time or Position-based capture, MCFB_ StepLimitSwitchFastInput, HTML5 help.
F	2.9	May 2015	AKD-C/N support, flexible PDOs, multiple E/IP assembly files, export-compliant firmware, new functions and function blocks, GUI improvements including easier EtherCAT device setup, improved IO and variable mapping, improved IO viewing.
E	2.8	May 2014	Coordinated Motion (3D linear w/ blending and transition; nD w/o blending and amp transition), HTTP variable interface, UDP socket interface, EtherCAT MDP management.
D	2.7	October 2013	Coordinated Motion; EtherCAT device configuration; KVB 2, Softscope; generic Modbus configuration; Dictionary improvements; Control panel changes.
C	2.7	June 2013	Initial doc.
B	2.6	December 2012	EtherCAT devices backup and restore; PLCopen Registration; Superimposed Axes; Cam on the fly; EtherNet/IP; PROFINET; Reset to factory settings; Save project on controller; Password protection.
A	2.5	May 2012	Initial release of KAS-IDE and AKD PDMM.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Edition	Software Version	Date	Description
0.2		September 2010	First release for KAS R1.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1 Preface

Welcome to the Kollmorgen Automation Suite help and documentation.

This document covers the use and programming of the KAS-IDEand PCMM2G information about KAS components, and much more.

This section explains how to use the online help provided with KAS.

1.1 System Requirements

KAS-IDE and KAS Simulator are compatible with any PC having this minimum hardware.

These are the minimum system requirements for the KAS-IDE.

Component	Description
Supported Operating Systems	• Microsoft Windows 11 (64-bit). • Microsoft Windows 10 (32-bit or 64-bit). • For optimal performance, verify the operating system is fully updated with the latest patches.
Connectivity	• 1 Ethernet port or WiFi connectivity, at either 100 MB or 1 GB. • See Note #2.
Display	• WXGA+ (1440 x 900) or higher-resolution monitor with 24-bit, 16.7M colors. • See Note #1.
Hard Disk	Free space required on hard disk: • 32-bit hard disk: 16GB • 64-bit hard disk: 20GB
Network	Internet connection for downloading and updating WebView2 Runtime.
Processor	Intel Pentium M or equivalent processor at 1.5GHz or greater.
RAM	• 32-bit: 1GB RAM • 64-bit: 2GB RAM or greater • Recommended for complex applications.
Software	• Microsoft Edge WebView2 Runtime (Evergreen) version 133.0.3956.59 or newer. • .NET Framework 4.8.1 or newer.
Web Browser	• A modern web browser is required to access the controller Web server and online help. • We recommend Microsoft Edge.

NOTE

1. Better results are achieved with OpenGL and 3D cards.
2. A 100MB or 1GB network is required to allow the KAS-IDE to Runtime communication to work in all conditions.
   The AKD WorkBench AutoTuner and Scope both require 100MB of bandwidth to function properly.

TIP

- See Connect Remotely about the ports used by the KAS.
- The ports may need to be opened to support connecting to a controller from an external network.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1.1.1 Digital Signatures

The KAS-IDE Installer and the executables are digitally signed to protect the integrity of these files and identify the distributor of the software components.

The digital signature appears as part of the file properties.

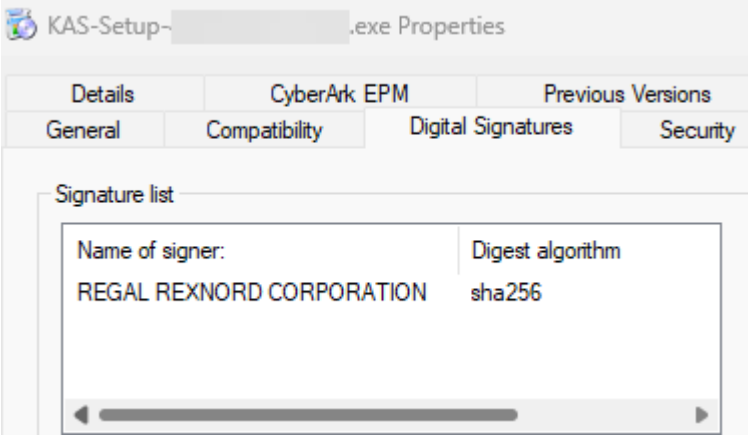


Figure 1-1: Example: KAS Setup Properties Digital Signature

Kollmorgen | kdn.kollmorgen.com | July 2026
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

2 PCMM2G Quick Start

Use this PCMM2G quick start to get the PCMM2G running quickly.

1 Grounding

EN: Ground the controller with a copper braided ground strap.

DE: Erden Sie den Steuerung mit einem kupferumflochtenen Erdungsband.

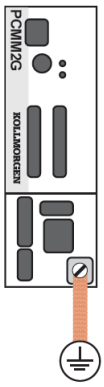
ES: Conecte a tierra el controlador con una correa de tierra trenzada de cobre.

FR: Mettez le contrôleur à la terre à l'aide d'une sangle de mise à la terre tressée en cuivre.

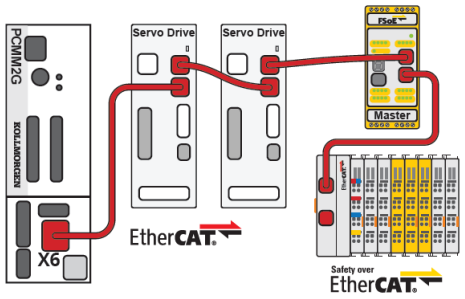
IT: Collegare il controller a terra utilizzando un cavo di messa a terra intrecciato in rame.

PO: Ligue o controlador à terra com uma fita de ligação à terra trançada de cobre.

ZH: 使用铜编织接地带将控制器接地。

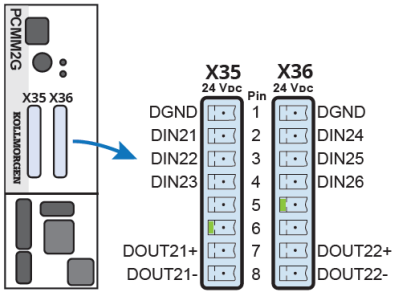


2 EtherCAT Devices



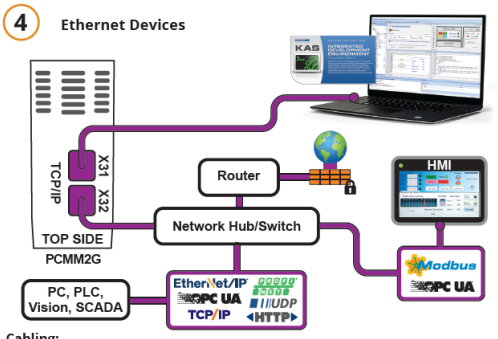
Cabling:
Standard RJ45 CAT5e/CAT6 Shielded Ethernet Cables

3 IO Connections



Digital Inputs: ON > 15 V (max 15 mA), OFF ≤ 5 V (max 15 mA)
Digital Outputs: Max. 30 Vdc, 100 mA

4 Ethernet Devices



Cabling:
Standard RJ45 CAT5e/CAT6 Shielded Ethernet Cables

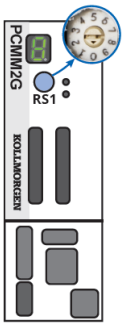
Continued →

5 Rotary Switch (RS1)

Adjust IP address if required

RS1	IP Address
0	DHCP / Auto-IP (169.254.x.x)
1	Static (user configurable) (default 192.168.1.101)
2-9	192.168.0.10n (n = RS1 position)

IP Address: Displayed on the 7-segment LED during startup.

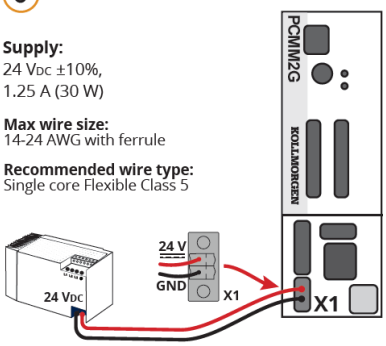


6 Power

Supply:
24 Vdc ±10%,
1.25 A (30 W)

Max wire size:
14-24 AWG with ferrule

Recommended wire type:
Single core Flexible Class 5

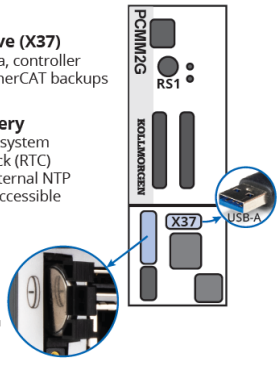


7 Optional Steps

USB Flash Drive (X37)
Use to store data, controller backups, and EtherCAT backups

Coin Cell Battery
1. Maintains the system Real-Time Clock (RTC)
2. Required if external NTP server is not accessible

Battery Type:
CR2032,
3 V, ≥ 190 mAh



8

KOLLMORGEN Automation Suite
SOFTWARE DOWNLOADS

Note: Requires registration and login

https://www.kollmorgen.com/en-us/developer-network/kas-ide-distribution-and-downloads-0

9

Firmware and Manuals

https://www.kollmorgen.com/en-us/developer-network/pcmm2g

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

3 Migrate from an AKD PDMM / PCMM (PxMM) to a PCMM2G Controller

This procedure is used to:

- Replace replacing an existing PxMM controller with a PCMM2G controller.
- Migrate the associated KAS project.

IMPORTANT

- This procedure is intended for users familiar with PxMM controllers.
- Starting with KAS v5.00, support for PxMM controllers is discontinued.
  - KAS v4.05 is the last officially supported KAS release for these controllers.

3.1 Migration Overview

These are the basic steps to complete the migration from a PxMM to PCMM2G controller:

A. "Prepare for Migration" (→ p. 46)
B. "Optional: Prepare the Existing .KAS Project for Retained Variable Transfer" (→ p. 46)
C. "PCMM2G Hardware Preparation" (→ p. 48)
D. "PCMM2G Configuration" (→ p. 49)
E. "Migrate the .KAS Project" (→ p. 50)

3.2 Prepare for Migration

Before migration, download the required software and create a backup of your project.

3.2.1 Download the Required Software

1. Download and install the latest KAS-IDE.
   - See KAS-IDE Distribution.
   - Registration and login is required for this site.
2. Download the latest PCMM2G firmware.
   - See KSN Downloads - PCMM2G Firmware.
3. Optional: Identify the firmware version running on the existing PxMM controller.
   a. Install the corresponding KAS-IDE version.

   NOTE

   - Installing the matching KAS-IDE version is only required if you plan to transfer retained variables values from the existing PxMM controller to the new PCMM2G controller.

3.2.2 Backup the Project

Before making any changes, create a backup copy of the existing .KAS project.

This backup preserves the original project and provides a recovery point, if needed, during migration.

3.3 Optional: Prepare the Existing .KAS Project for Retained Variable Transfer

This procedure prepares the existing PxMM controller so retained variable values can be transferred to the PCMM2G controller during migration.

NOTE

- Skip this section if transferring retained variable values is not required.

Procedure

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. Start the KAS-IDE version that matches the firmware installed on the existing PxMM controller.
   See Start the KAS-IDE.
2. Open the existing project:
   a. Select **File** > **Open**.
   b. Browse and open the .KAS project.
3. Enable the PLC Variable Remote Access:
   a. In the Project View panel, right-click Controller and select Properties.
   The Controller Properties dialog box opens.
   b. Select the Enable PLC Variable Remote Access check box. (Figure 3-1)

IMPORTANT

- **Why is this required?**
  - Enabling PLC Variable Remote Access allows retained variable values to be read from the existing controller and transferred to the PCMM2G during the migration process.

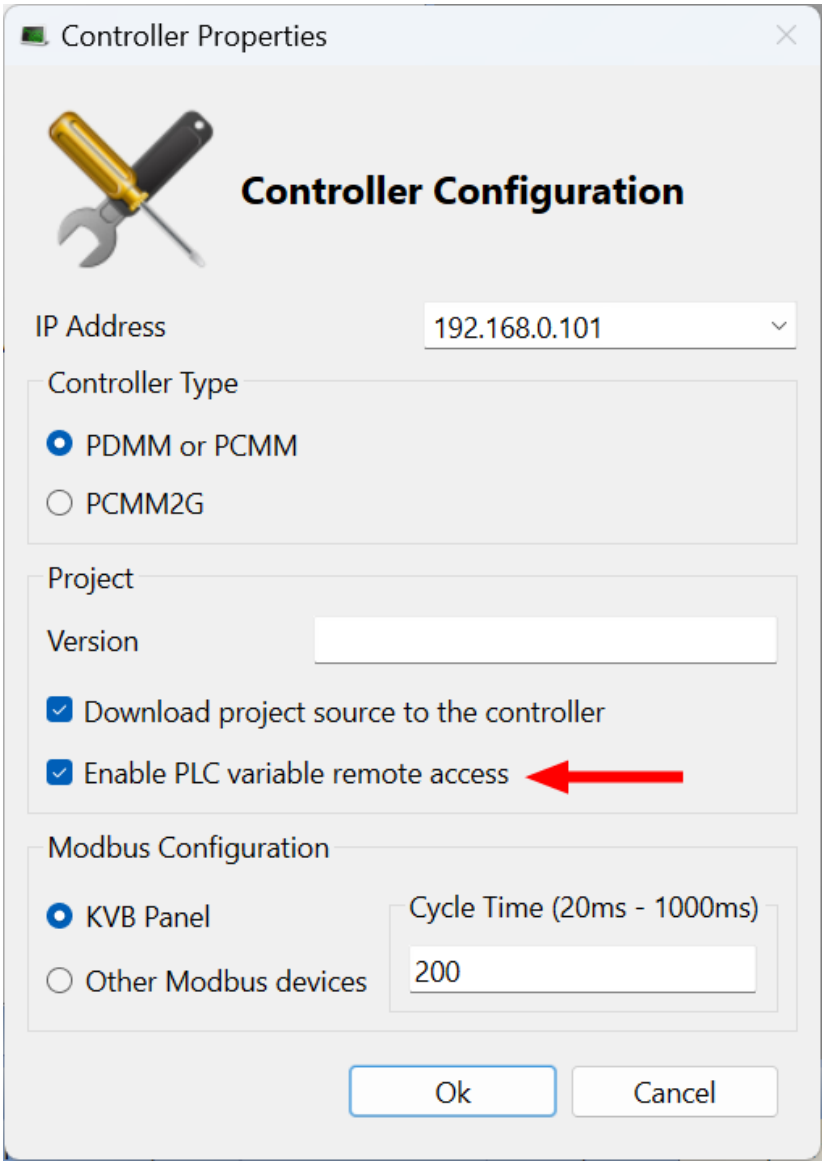


Figure 3-1: Controller Properties dialog box

   c. Click **OK** to save the changes or selections and close the dialog box.
4. On the toolbar, click the **Connect** button ( ) to establish connection with the target controller.
5. Compile the project.
6. Download the .KAS project to the PxMM controller.

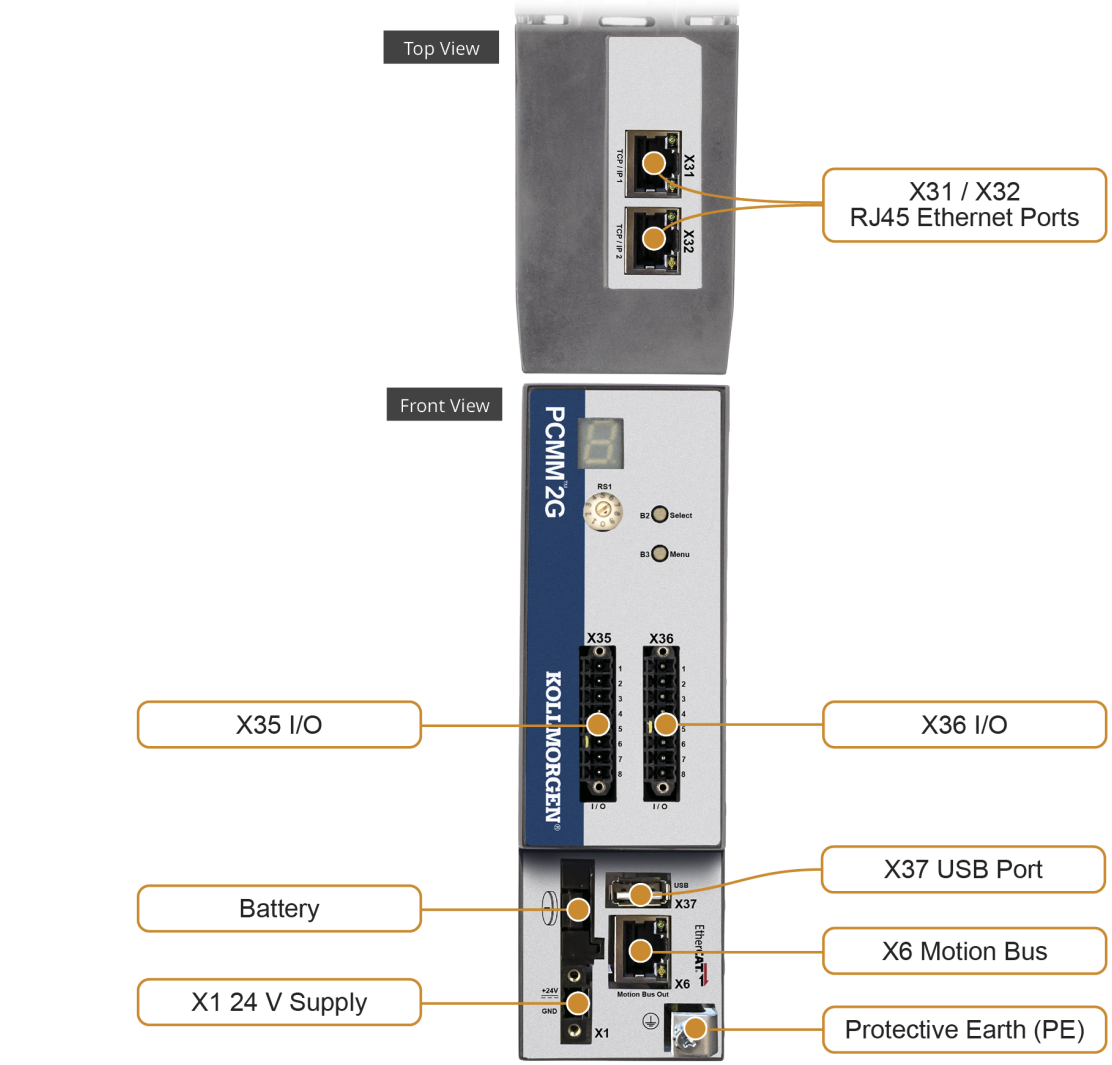
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

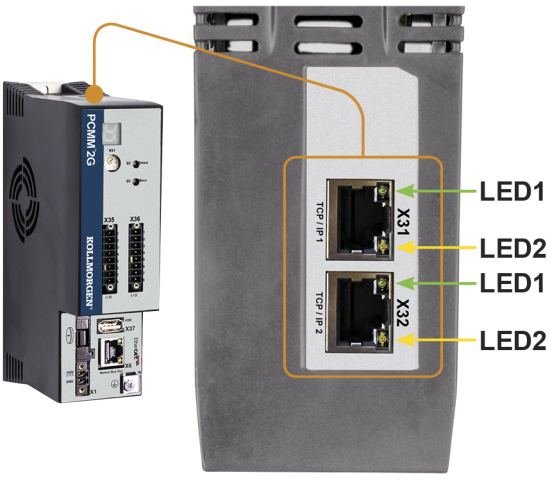
7. Run the project and verify it operates as expected.
8. Save the verified project.
9. Continue with "PCMM2G Hardware Preparation" (→ p. 48).

3.4 PCMM2G Hardware Preparation

Use this procedure to physically install and power the PCMM2G controller.

Procedure





See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Figure 3-2: PCMM2G Hardware Connections

1. **Verify the PCMM2G is turned off.**
2. PCMM2G-specific settings:
   a. Connect the service interface (X31 or X32) of the PCMM2G to an Ethernet interface on the PC directly or use a network hub/switch.
   Use standard Cat. 5e / Cat. 6 Ethernet cables for connection.
   b. Insert the USB flash drive into X37.
3. Hardware settings are similar to the PxMM:
   a. Set the Rotary Switch RS1 to the applicable position.
   See the PCMM2G Installation Manual for additional setup information.
   b. Move these cables from the PxMM to the PCMM2G:
      - X35 and X36 I/O port.
      - X6 EtherCAT Motion Bus port.
   c. Attach a 24 VDC power supply connection to X1 to power on the PCMM2G.
   The "PCMM2G Boot Sequence" (→ p. 597) starts.
4. Confirm the X31 or X32 indicator on the PCMM2G are green.
5. Continue with "PCMM2G Configuration" (→ p. 49).

3.5 PCMM2G Configuration

Use this procedure to configure the controller and update it to the required firmware version.

Procedure

1. Download the latest PCMM2G firmware from the KSN page here: KSN Downloads - PCMM2G Firmware.
2. Identify the PCMM2G's IP address.
   See "View the Controller IP Address" (→ p. 602).
3. Open a web browser and enter the controller's IP address.
   The PCMM2G's Web server opens.
4. Click the **LOGIN** link at the top of the window.
5. Enter the password to log in.
   The factory default password is `administrator`.
   This can be changed after logging in.
6. Click the "Settings tab" (→ p. 640). (Figure 3-3)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

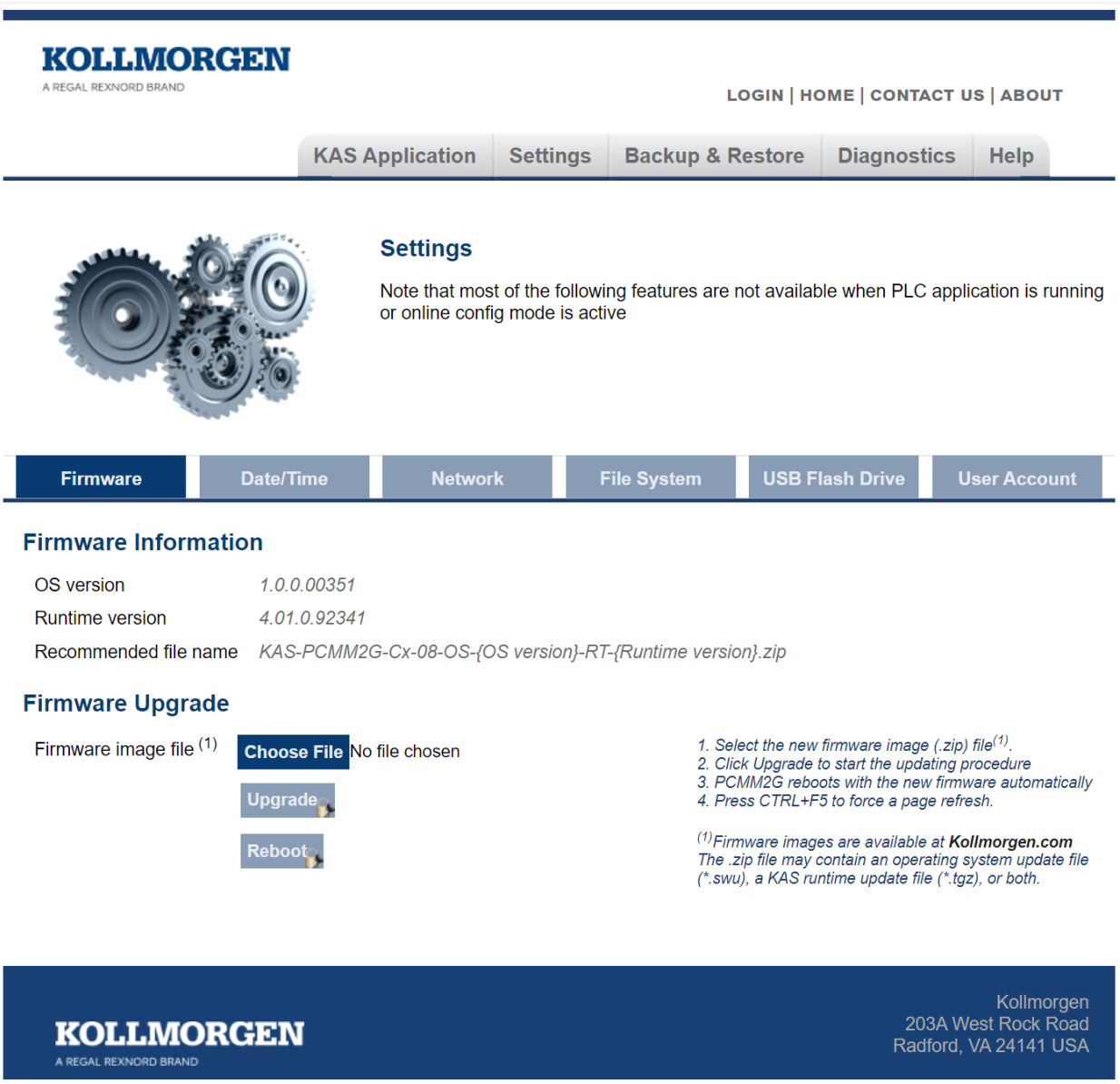


Figure 3-3: Settings tab

7. On the "Firmware tab" (→ p. 667), complete the "Upgrade the Controller's Firmware" (→ p. 668) procedure.
8. On the "Date/Time tab" (→ p. 662), edit the date, time, and time zone.

   Notes about the Date/Time tab

   - By default, the controller uses an NTP server for time synchronization.
     - The Date/Time can be manually changed using the "Update Date/Time" (→ p. 664) procedure.
   - A Time Zone must be assigned to the PCMM2G.
9. If the PCMM2G Rotary Switch RS1 is set to 1, use the "Network tab" (→ p. 675) to manually define the IP address.
10. Continue with "Migrate the .KAS Project" (→ p. 50).

3.6 Migrate the .KAS Project

Use this procedure to convert and run the PxMM project on the PCMM2G.

Procedure

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. Start the KAS-IDE version that matches the firmware installed on the PCMM2G controller.
2. In the KAS-IDE, click **File > Open** to search for and open the .KAS project saved in the "Prepare for Migration" (→ p. 46) procedure.
   The **Project Compatibility Warning** dialog box opens. (Figure 3-4)

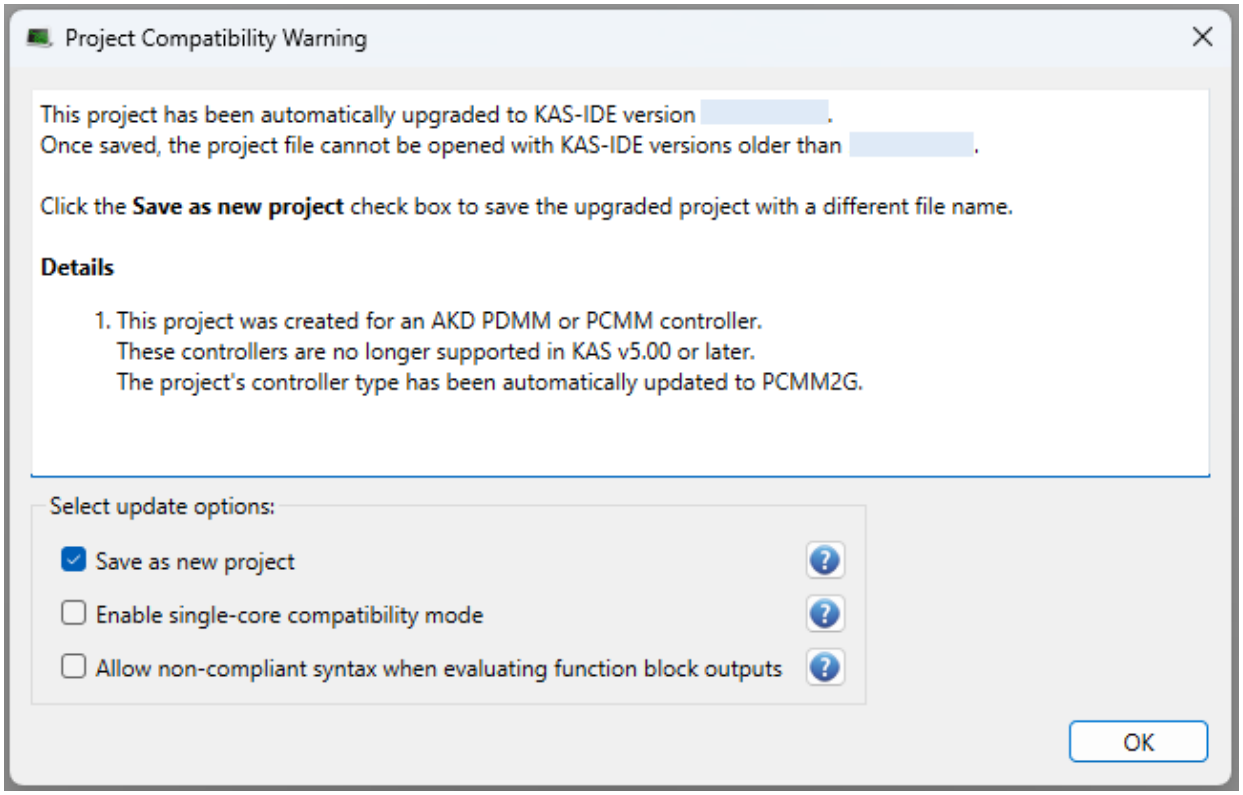


Figure 3-4: Project Compatibility Warning dialog box

3. In the dialog box, configure the applicable update options.
   a. Optional (but recommended): Accept the default selection of the **Select Save as new project** check box to make a copy of the newly converted project.
      - This preserves the original project file.
      - If this check box is selected, the Project Save-As dialog box opens.
      - You are prompted to specify the file location and name for the updated project.
   b. Select the **Enable Single-Core Compatibility Mode** check box if:
      - The application does not follow multi-core controller programming guidelines.
        - See "Program a Multi-Core Controller" (→ p. 386).
      - Reduced performance on the PCMM2G is acceptable for the application.

      IMPORTANT

      The limitations listed in "Single-Core Compatibility Mode" (→ p. 161) are acceptable.
   c. Select the **Allow non-compliant syntax to evaluate function block outputs** check box if:
      - The application was developed using a version of KAS earlier than v4.00.
      - You prefer not to update the KAS project to use compliant syntax for the corresponding function blocks.
        - See "Unexpected complex data in expression." (→ p. 798).
   d. Click **OK** to save the changes or selections and close the dialog box.
4. Optional: Transfer the retained variables values from the existing PxMM controller:
   a. Open the Dictionary view and select the Variables tab.
   b. Right-click **RETAIN variables** and select **Copy variable value to init values from another controller**. (Figure 3-5)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

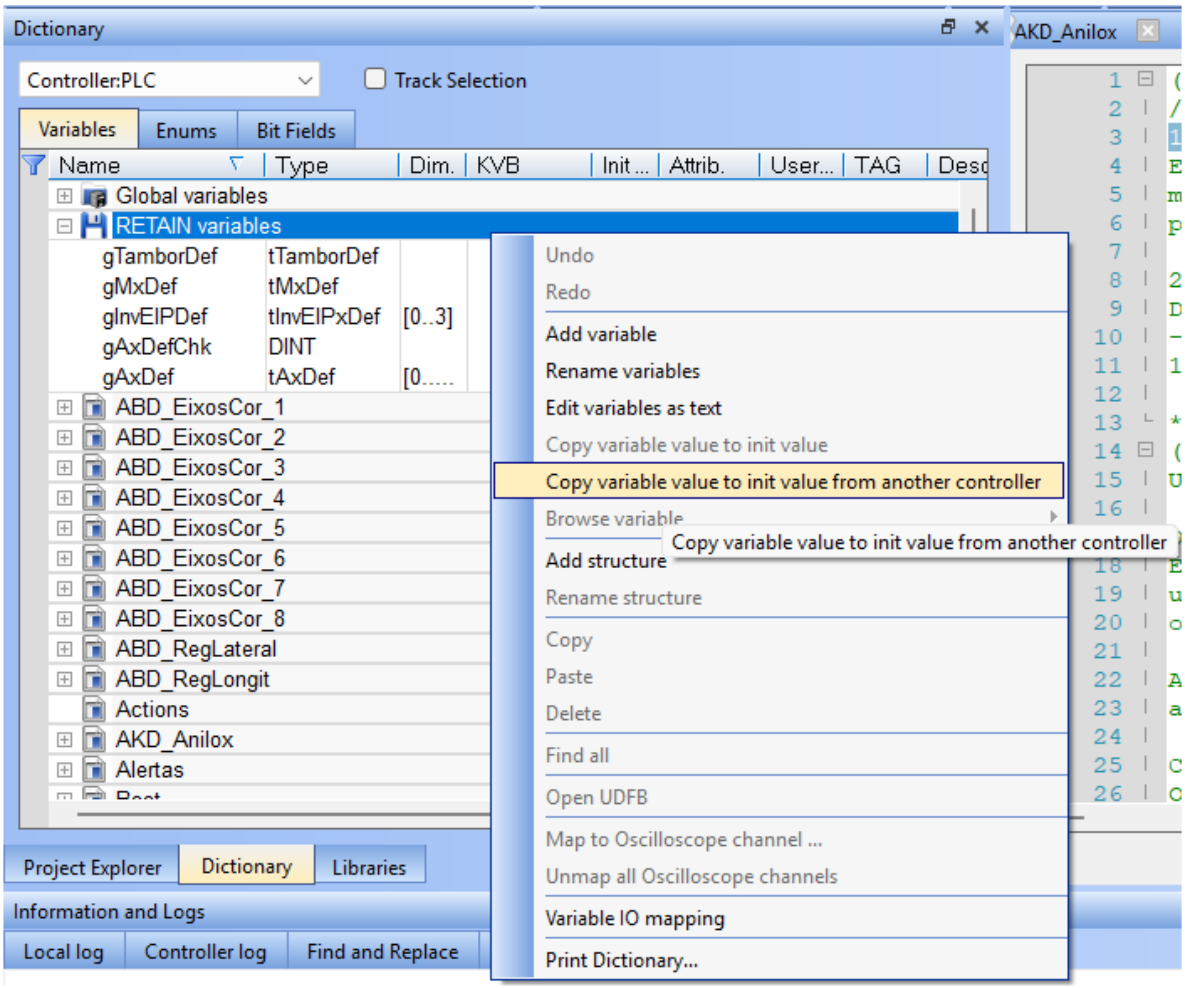


Figure 3-5: Select **Copy variable value to init values from another controller**

The **Copy Variable Values to Init Values** dialog box opens. (Figure 3-6)

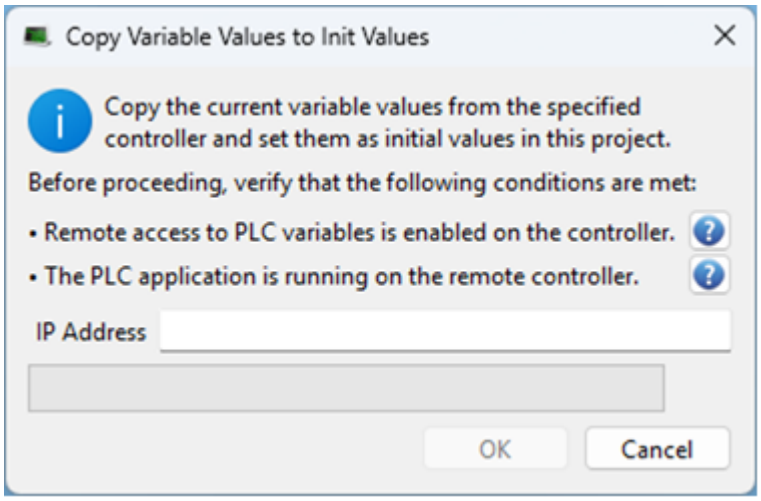


Figure 3-6: Copy Variable Values to Init Values dialog box

c. In the dialog box, enter the IP address of the existing PxMM controller. (Figure 3-7)

IMPORTANT

- The source PxMM controller must be online and PLC Variable Remote Access must be enabled.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

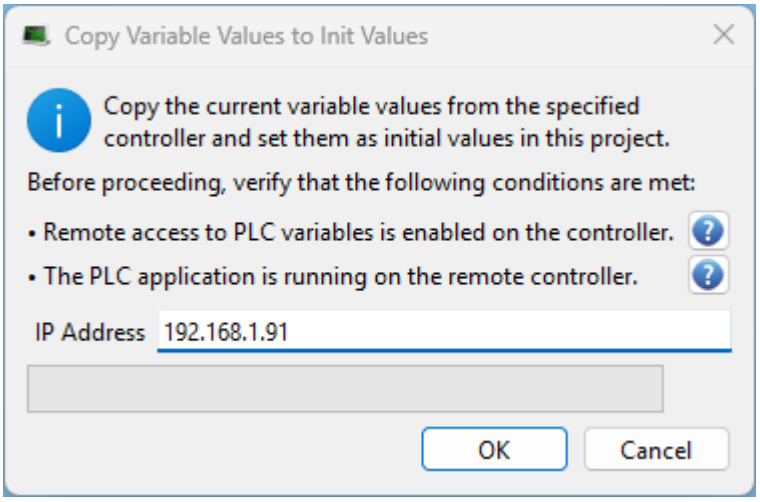


Figure 3-7: Entered IP address

d. Click **OK** to save the changes or selections and close the dialog box.
The transfer starts.
A confirmation message appears. (Figure 3-8)

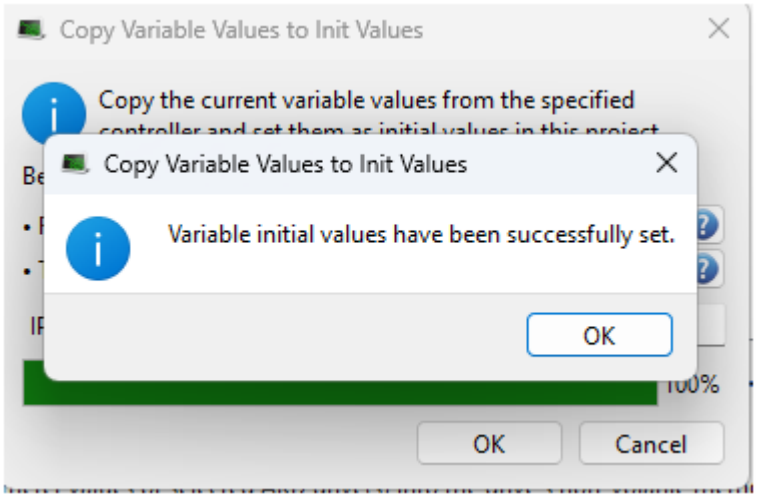


Figure 3-8: Confirmation message

When the operation completes successfully, the retrieved retained variables values are stored as initial values in the project.

e. Click **OK** to close the confirmation message and dialog box.

5. Update the Project Controller settings:
   a. In the **Project View** panel, right-click **Controller** and click **Properties**.
   The **Controller Properties** dialog box opens. (Figure 3-1)

b.

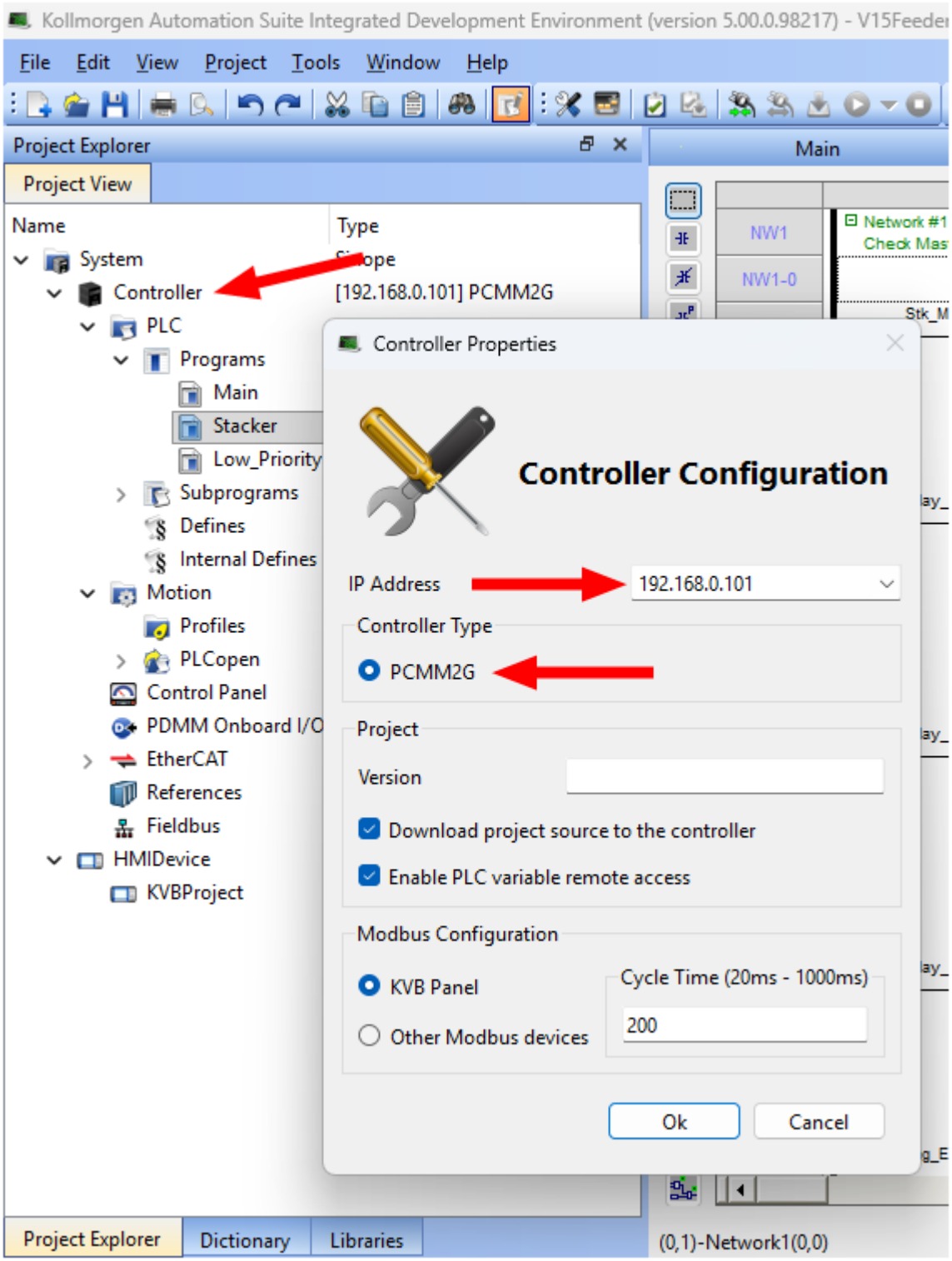


Figure 3-9: Controller Properties dialog box

c. In the **IP address** drop-down, enter the PCMM2G IP address.
d. Click **OK** to save the changes or selections and close the dialog box.

6. Connect to the PCMM2G:

a. On the toolbar, click the **Connect** button () to establish connection with the target controller. An **KAS Controller Login** dialog box opens. (Figure 3-10)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

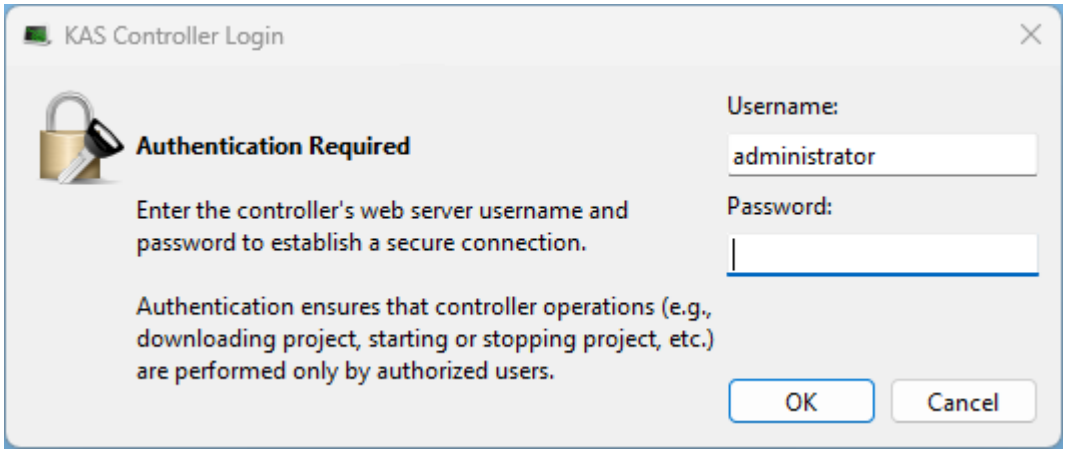


Figure 3-10: KAS Controller Login dialog box

b. Enter the PCMM2G Web server Username and Password to authenticate the connection. The default password is `administrator`.
c. Click **OK** to save the changes or selections and close the dialog box. The KAS-IDE connects to the controller.

7. Build and validate the application:
   a. Compile the project.
   b. Download the .KAS project to the PxMM controller.
   c. Run the project and verify it operates as expected.
      - If retained variable values were copied from the existing PxMM controller, perform a Cold Start to initialize the application using the transferred values.
      - To perform a Cold Start:
        i. Click the Start button drop-down.
        ii. Select Cold Start - Use Initial values for retain variables. (Figure 3-11)

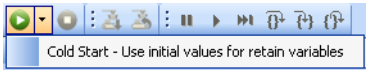


Figure 3-11: Cold Start - Use Initial values for retain variables

   d. Save the **migrated** .KAS project.
8. Create a backup: After successful validation, use the Web server to complete the "PCMM2G Controller and EtherCAT Devices Backup" (➔ p. 683) procedure.

3.7 Migration Tips

- Always keep a working PxMM backup.
- Start with the Single-Core Mode when migrating legacy applications.
- After validation, optimize the application for multi-core execution (if applicable).
- Verify the operation at each stage before continuing to the next step.

3.8 Migration Validation Checklist

Before considering migration complete, verify these are completed:

☐ The migrated project compiles without errors.

☐ The project downloads successfully to the PCMM2G controller.

☐ Retained variable values transferred correctly (if applicable).

☐ EtherCAT devices reach Operational state.

☐ I/O functions correctly.

☐ Motion functions operate as expected.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- ☐ Date, time, and network settings are configured.
- ☐ Controller and EtherCAT backups have been created.
- ☐ The application has completed a full operational test cycle successfully.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4 Kollmorgen Automation Suite Introduction

This section introduces Kollmorgen Automation Suite (KAS) with a product overview that lists the features, components, and different implementations.

- KAS is intended for engineers to design and build high-performance motion control and automation systems.
- KAS is designed to quickly and easily compose a motion application.
- This is achieved with all of the re-use and flexibility of the KAS libraries in conjunction with the IEC 61131-3 programming languages.

KAS can cover a wide variety of applications:

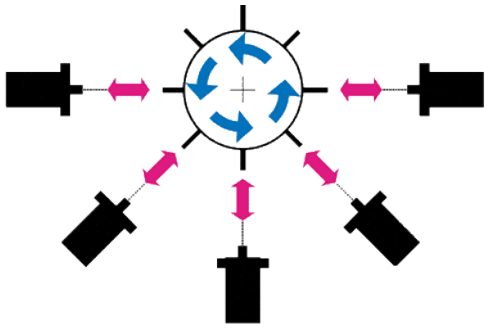


Figure 4-1: Synchronized Feeder

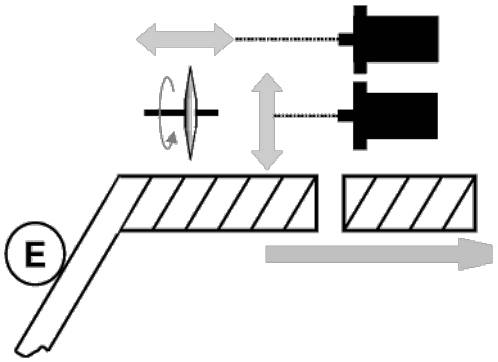


Figure 4-2: Spring Winding

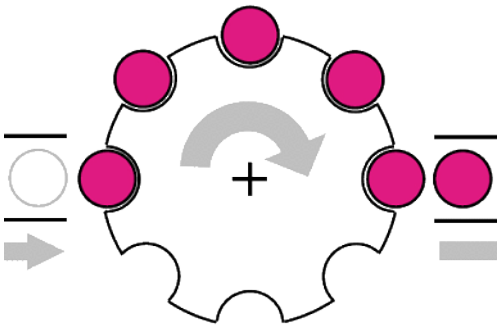


Figure 4-3: Synchronizer

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

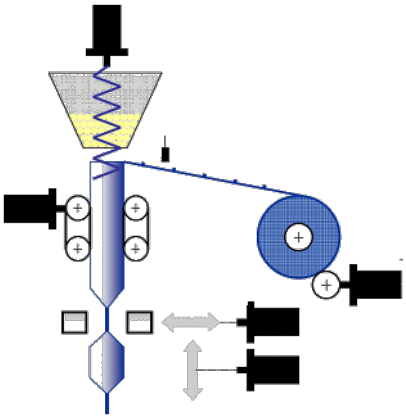


Figure 4-4: Form Fill Seal

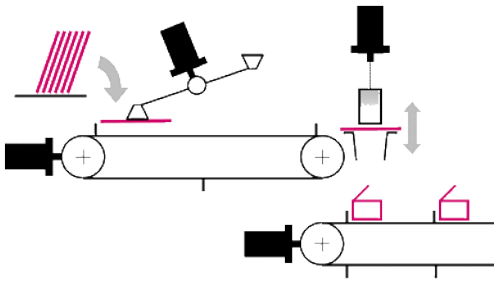


Figure 4-5: Carton Erector

4.1 Learning Kollmorgen Automation Suite

To learn Kollmorgen Automation Suite (KAS), you can:

- Navigate this online help and choose sections depending on your experience.
- Read the KAS Manuals (→ p. 59).

4.1.1 Where to Start

The KAS documentation includes information for readers from a variety of backgrounds.

TIP
To get the most out of the documentation, we recommend you start reading the sections that are most relevant to you. Within each section, read through the topics in sequence.

4.1.1.1 Beginner

- Find basic information about KAS in "Kollmorgen Automation Suite Introduction" (→ p. 57).
- If you are not familiar with the concepts behind KAS, read the "Understanding KAS" (→ p. 286).
- An overview of the KAS-IDE User Interface is in "KAS Graphical User Interface Description" (→ p. 83).
- To get information on how to run and debug the project, read "Start the KAS Simulator" (→ p. 262) and "Test and Debug the Project" (→ p. 268).

4.1.1.2 Advanced User

- To design and create a project, see "Getting Started" (→ p. 148).
- For explanations about the tools used by the KAS-IDE, see "Tools" (→ p. 695).
- For in-depth information, see "Advanced Topics" (→ p. 820).

4.1.1.3 Online

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- KSN: The Kollmorgen Support Network (KSN) has many articles, resources, and a user community to learn, ask questions, and contribute to our community.
  - See KDN at https://www.kollmorgen.com/en-us/developer-network/
- YouTube: Kollmorgen has a channel dedicated to KAS.
  - See the KAS Online Learning Collection.

4.1.2 Read the KAS Manuals

If you prefer to read a PDF, these manuals are available:

Document Title	Description
Release Notes	The KAS Release Notes contain a summary of: • New features, fixed and known limitations, workarounds • Information on all hardware and software components that have been updated, changed, or added in the release. • See KAS Release Notes EN.
Getting Started	Contains these main steps to get the KAS system up and running. • **Hardware (HW) Installation** (Connection and Wiring): Wiring and hardware details, connectors, system diagrams. • **HW Configuration**: Basic configuration and settings needed to start the HW components (e.g., HMI, Industrial PC, Fieldbus, I/O). • **Software (SW) Installation**: KAS software setup. • See KAS IDE Getting Started EN.
30 Minutes to Motion	Contains the main topics to get started quickly with KAS-IDE. The objective is to familiarize you with the basic principles and the way the program works by creating a simple motion application project. See KAS IDE 30 Minutes to Motion EN about: • Key Features in KAS. • Explore the Workspace • See "KAS Graphical User Interface Description" (→ p. 83). • Build a Motion Project • **Start Projects** - Create a project from scratch or modify an existing project. See Create a Project. • **Add Components** - Add necessary elements to build the project and control the motion part of the system. This includes: "Create Variables" (→ p. 515), "Add Motion" (→ p. 187), and "Programs: Structure and Syntax" (→ p. 481). • **Build Output** - Select a device and generate the application to deliver to users. See "Run the Project" (→ p. 257). • **Run Output** - Make the output accessible to end-users.
KAS-IDE User Manual	Contains content to help with KAS-IDE except the topics included in the Reference Manuals. See KAS-IDE User Manual EN.
KAS-IDE PLC Library Reference Manual	Contains Technical References on **PLC** Programming Languages and Library. See KAS-IDE PLC Library Reference Manual EN.
KAS-IDE Motion Library Reference Manual	Contains Technical References on **Motion** Library for Pipe Network and PLCopen. See KAS-IDE Motion Library Reference Manual EN.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TIP

The KAS-IDE allows "References" (➜ p. 96) to external files (e.g., the PDF files).
See "Use the Reference Folder" (➜ p. 284).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4.2 Key Features

The purpose of KAS is to include, in a single software package, all the tools you need (i.e., a soft PLC, configuration tools, and a motion controller) to create an automation system.

An overview of an investment in Kollmorgen for Motion Control and Automation Systems solution is shown in Figure 4-6:

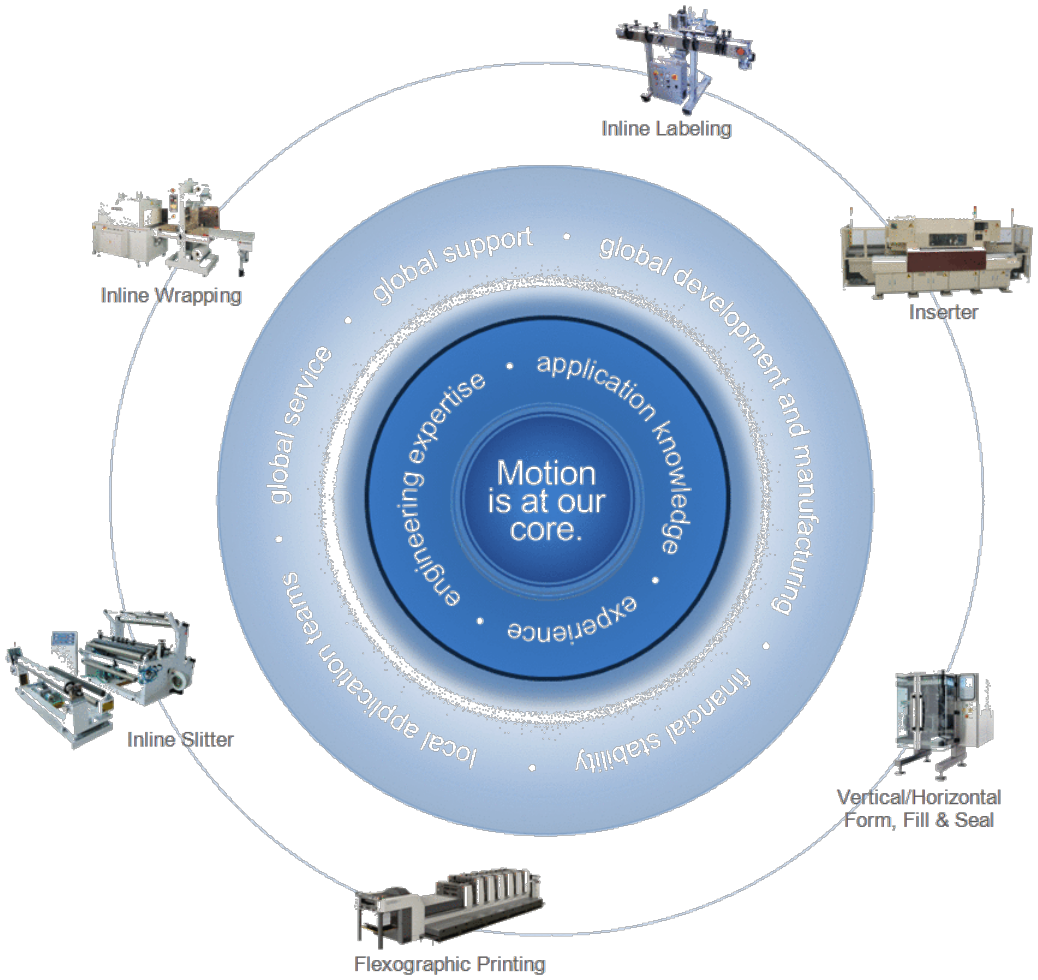


Figure 4-6: Motion at our Core

- KAS is Kollmorgen's all-in-one solution for designing, developing and maintaining automation systems.
  - As a solution offering, it brings many years of Motion Control experience to the market and this is coupled with technical expertise and experience, global delivery capability, and strong financial performance.
- KAS is a set of software packages designed to run and take advantage of Kollmorgen's extended set of these integrated hardware products:
  - Programmable Automation Controllers
  - Programmable Drives
  - AKD drive family
  - AKM motor family, gear boxes, I/O terminals and Human Machine Interaction terminals (or Operator Interfaces.)
- KAS provides machine builders with a high-performance, cost-effective, and easy to use solution for building machines.
- KAS achieves this by integrating in a coherent, intuitive, and flexible way these main functionalities of a machine:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Precise control of all moving parts (Motion control)
- Interface with machine operators (HMI)
- PLC programming of the machine (IEC 61131-3 Soft PLC)

KAS is made of these software components:

- KAS-IDE - the Integrated Development Environment allowing development and monitoring of complex machine automation systems.
- KAS Runtime - the Runtime engine offers the functionally of both **a High-Performance Motion and a PLC Engine**.

4.2.1 Kollmorgen Automation Suite-Integrated Development Environment (KAS-IDE)

KAS comes with a powerful Integrated Development Environment.

- The KAS-IDE provides machine builders with all the necessary tools for designing, programming, configuring, debugging, and maintaining machine applications.
- KAS uses the same interface, tools, and libraries to create applications for various types of KAS controllers (PCMM2G).

With the KAS-IDE, system engineers can:

- Create new application projects using predefined or custom application templates.
- Define the machine hardware architecture ("Fieldbus" (→ p. 79), "Motion Bus" (→ p. 79), "Controllers" (→ p. 73), "Drives" (→ p. 772), and motors) and the machine program (HMI panels, IEC 61131-3 programs and function blocks, motion block, profiles and axes) from a centralized Project Explorer which is based on a tree-structure.
- Develop PLC programs, functions and function blocks using the:
  - IEC 61131-3 programming languages (FBD, FFLD, IL, SFC, and ST).
  - IEC 61131-3 standard library.
  - KASFunction Block libraries dedicated to motion, communication and monitoring.
- Centrally manage all IEC 61131-3 variables with KAS variable dictionary on the "Variables tab" (→ p. 102).
- Create and organize libraries of functions and function blocks.
- Easily set up **HMI panels** using the "Human-Machine Interface Editor" (→ p. 709) that is part of the HMI control library and map graphical objects to IEC 61131-3 variables.
- Graphically design advanced multi-axis relations using **Kollmorgen's graphical motion programming** environment, the Pipe Network, with its tool generating code automatically.
- Use ultra-fast IEC 61131-3 compiler to validate the syntactical correctness application code.
- Configure hardware devices via an integrated set of configuration tools (e.g., AKD drives, EtherCAT I/O terminals, etc.)
- Access controller devices to download, start and stop the application, watch log messages and send shell commands to the target device.
- Debug the application by inserting break points and stepping into the code or by monitoring internal values (IEC 61131-3 variables, motion positions, drive's internal values) directly in the editors or with KAS advanced "Soft Oscilloscope" (→ p. 737) tool.
- Access the full online documentation.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4.2.2 KAS Runtime

Kollmorgen Automation Suite Runtime (commonly named **KAS Runtime**) offers, in a single software package, the functionally of both a soft PLC and a motion controller.

KAS Runtime (virtual machine) is a high-performance deterministic environment designed to run on different hardware platforms ranging from low-cost **programmable drives** to **high-end Programmable Automation Controllers**. This gives machine builders all the flexibility when designing their machines.

KASsupports many configurations when integrating machines:

- Ranges from **single-axis** to more than **200 tightly coordinated axes**.
- With a **centralized** (Programmable Automation Controllers), **distributed** (Programmable Drives) or **mixed** (Programmable Automation Controllers + programmable drives) control architecture.
- Runs on a **single or multiple controllers**.
- Communicates via Ethernet, EtherCAT, PROFINET, EtherNet/IP, or Modbus.
- Uses the high-performance Pipe Network or the standard PLCopen function blocks.
- Controls Kollmorgen's drives (AKD, some of the Servostar Sxxx drive family), AKM motors, and AKT terminals for I/Os products).

The KAS Runtime can be used in these contexts:

- Controller implementation (PCMM2G)

See "Implement Multi-Axis with PCMM2G" (➔ p. 81).

4.2.3 KAS Simulator

Thanks to the **KAS Simulator** tool, development of the machine software can start before (or parallel with) bringing in production. Thus developers can already begin to develop and test their application without any hardware available. Once ready, the application can be deployed on the targeted controller.

By allowing developers to run the full application (HMI, soft PLC, motion and I/Os) totally independently from the final targeted machine, KAS helps reducing:

- **Development time**, by limiting down time waiting for other engineers (mechanical, electrical…) to test, fix and tune the real machine
- **Cost**, by limiting broken hardware due to software bugs.

See "Using the KAS Simulator" (➔ p. 586) for more information.

4.2.4 KAS Architecture

This diagram shows an architectural view of what KAS is from a hardware and software perspective, or Real-time and not Real-time components.

- Hardware is shown in blue shaded boxes and, minimally consists of a computer, a controller, one or more drives, I/O, and optionally an HMI device.
    - Each of these components has a software aspect, many of which communicate with other components.
- Communication may be done over:
    - Ethernet or HTTP (in blue)
    - EtherCAT (in green)
    - Modbus (in red)
    - Several other protocols

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

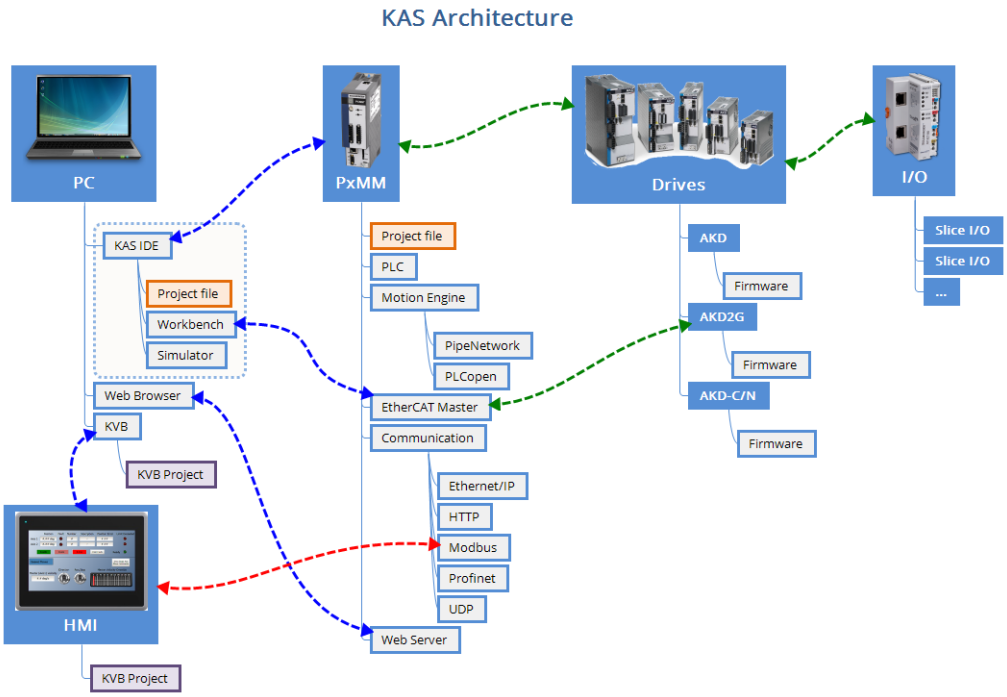


Figure 4-7: Architectural view of KAS components

4.2.4.1 Real-Time

The real-time components, including those running the KAS Runtime engine, consists of:

Item	Description
Virtual Machine (IEC 61131-3)	Responsible for managing an IEC 61131-3 application with its programs and variables.
Motion Manager	Manages motion engines, axis objects and motion bus drivers. The KAS Runtime comes with these motion engines: "Pipe Network Concept" (➔ p. 326) and "PLCopen" (➔ p. 341). • The motion engine implements different motion algorithms and functions to create, access, and delete pipes, pipe blocks, and axes (e.g., MLPNAxisCreate, MLGearInit, MLPipeAct). • It provides a set of Functions and Function Blocks that IEC 61131-3 applications can use to control the behavior of these algorithms
I/O Manager	Manages I/Os and I/O drivers. It works closely with the VM Manager instances to map transparently all IEC 61131-3 variables declared as input or output.
Motion Bus	A plug-in giving access to the EtherCAT network.
Custom Function Blocks	A plug-in implementing to "Create and Use Custom Libraries" (➔ p. 244) with function blocks.

4.2.4.2 Non-Real-Time

The non-real-time part is composed of:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Item	Description
KAS-IDE	The development tool used to create the project (i.e., design, create and run virtually).
KAS Runtime	...
KVB	The development tool used to create an HMI interface.
HMI	Available when integrated on a Programmable Automation Controller platform. Not present when integrated a programmable drive.

The interface between the real-time and non-real-time systems is done via shared memory buffers.

The KAS Runtime communicates with the KAS-IDE during operation to:

- Receive further instructions from the KAS-IDE such as a direct motion command.
- Provide status information to the KAS-IDE for motion and operation of the application program.
- Provide information displayed on the KAS-IDE scope.
- Provide Log information to the KAS-IDE.

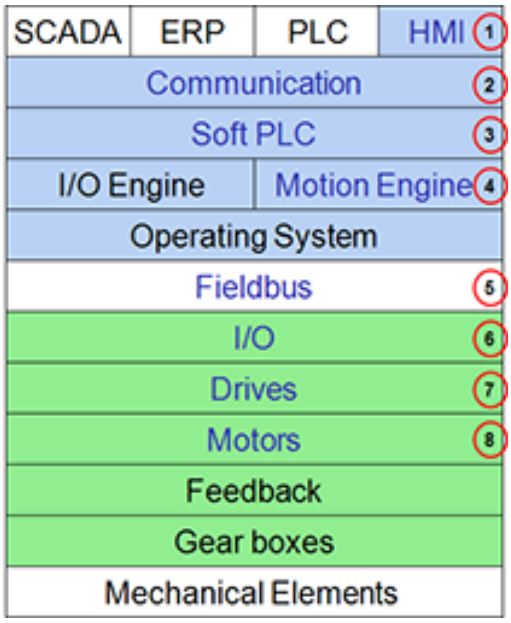
NOTE

When the KAS Runtime is implemented with a programmable drive, the interface between the real-time and non-real-time parts is done via Ethernet based on TCP/IP protocol.

4.2.5 Logical View

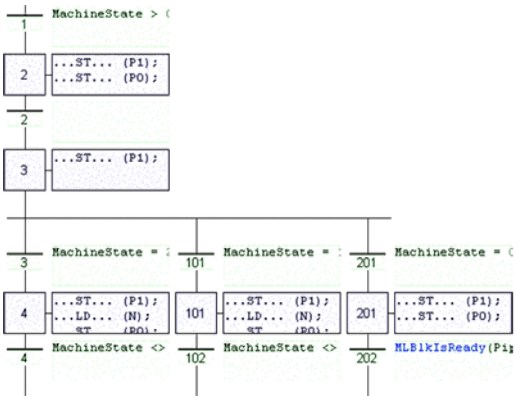
An automation system usually needs an organized hierarchy of controller systems to function.

This system usually includes these items:



Call out#	Item	Description
1	"Human-Machine Interface" (→ p. 73)	At the end-user top level, the Human Machine Interface is where the operator can monitor or operate the system. • It is usually composed of an AKI panel. • An HMI is usually employed to communicate with PLCs for further automated control or emergency response. • PLCs usually have built-in communications ports.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Call out#	Item	Description
2	"Communication and Fieldbus" (→ p. 79)	HMI is linked to the middle layer via a non time critical communication system (e.g., Modbus TCP protocol on Ethernet).
3	"PLCopen State Machine" (→ p. 383)	• A Programmable Logic Controller is a digital computer used for automation of industrial processes (e.g., control of machinery on factory assembly lines). • It is used to synchronize the flow of inputs from physical sensors and events with the flow of outputs to actuators and events.  **Figure 4-8:** Example: PLC
4	"Motion Concepts" (→ p. 320)	These are the Motion Engines available: • "Pipe Network Concept" (→ p. 326) • "PLCopen" (→ p. 341) • The Motion Engine implements different motion algorithms and functions to create, access and delete pipes, pipe blocks and axes. • It provides a set of Functions and Function Blocks that IEC 61131-3 applications can use to control the behavior of these algorithms.
5	"Fieldbus" (→ p. 79)	A fieldbus is an industrial network system for real-time distributed control (e.g., CAN). • It is a way of connecting instruments in an industrial network system that controls the motion. • Communicating frames requires high performance and reliability to ensure real-time distribution. • The fieldbus is the way to connect instruments in a plant design by linking the PLC to the external systems.
6	"Input/Output Bus" (→ p. 79)	Input/Output refers to the communication between an automation system and the outside world. • Inputs are the signals or data received by the automation system. • Outputs are the signals or data sent from it.
7	"Drives" (→ p. 772)	A Drive is an electronic device that provides power to a motor or servo.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Call out#	Item	Description
8	"Kollmorgen Motors" (➔ p. 80)	At the bottom of the control chain is the servo motor which actually does the work. • Servo motors are mostly used when circular motion is needed. • They can also be used for linear applications by transforming circular to linear motion with a bolt and screw transducer. • Some actuators, however, are intrinsically linear.

4.2.6 Physical View

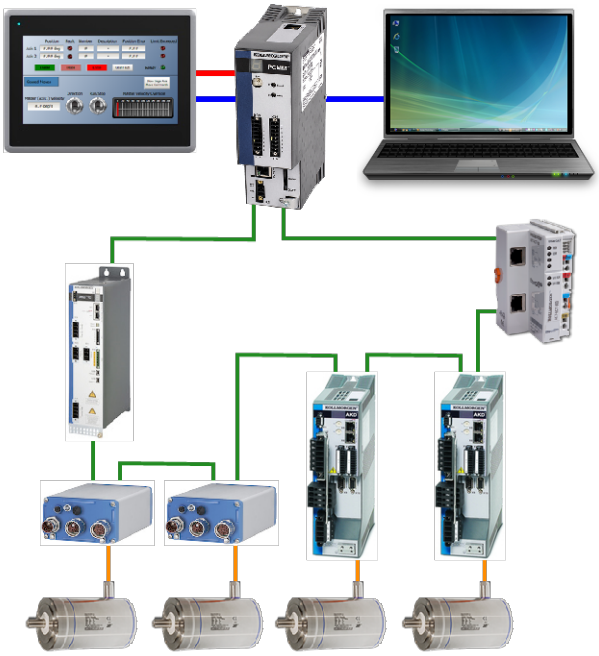


Figure 4-9: Example of Automation System

Color	Method of communication
Blue	Ethernet
Green	EtherCAT
Red	Modbus
Orange	Motor cable

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

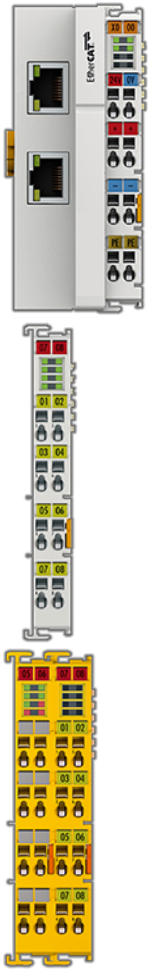
4.3 KAS Breakdown

Domain	Product	Product Example Image	Concept (Technology)	Installation	Tasks (Tools)	Reference
Controller	PCMM2G		Programmable Drive Multi-axis Master: "Controllers" (→ p. 73)	• Install the PCMM2G controller. • See Install Controllers.	• "Add the Controller" (→ p. 153) • "Configure the Controller" (→ p. 156)	"Controllers" (→ p. 767)
Drives	AKD AKD-C AKD-N AKD2G S300 S700		• "Drives" (→ p. 772)	• Install a AKD or AKD2G Drive. • See either the: • AKD Installation Manual • AKD Quick Start • Available from: AKD Downloads.	• "Add and Configure a Servo Drive" (→ p. 220) • "AKD Drive Configuration" (→ p. 448) • Update Kollmorgen Servo Drives Firmware	"Drives" (→ p. 772)
Fieldbus	EtherCAT		EtherCAT	• Set up EtherCAT Motion Bus Communication. • See AKD®EtherCAT Communication.	"Configure EtherCAT Motion Bus" (→ p. 164)	"Motion Bus and Fieldbuses" (→ p. 886) Cables

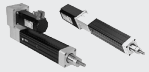
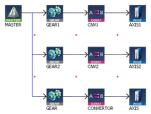
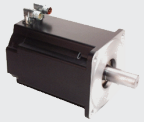
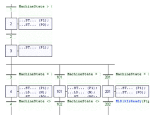
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Domain	Product	Product Example Image	Concept (Technology)	Installation	Tasks (Tools)	Reference
HMI	Human-Machine Interface			• Install the graphic operator interface. • See HMI - Graphic Operator Interface.	• "Use the Kollmorgen Visualization Builder (KVB)" (→ p. 709) • "Add an HMI Device" (→ p. 215)	"HMI" (→ p. 776)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Domain	Product	Product Example Image	Concept (Technology)	Installation	Tasks (Tools)	Reference
I/O Terminal			EtherCAT	• Install the remote Input/Output Terminal. • See Remote Input/Output - I/O Terminal.	• "Add and Configure I/O Slices" (➔ p. 231) • "Map Input and Output to Variables" (➔ p. 526)	"Remote Input/Output Terminals" (➔ p. 777)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Domain	Product	Product Example Image	Concept (Technology)	Installation	Tasks (Tools)	Reference
Mechanical						Linear Actuators Gearheads
Motion Engines		 	Motion Concept: • "Pipe Network Concept" (→ p. 326) • "PLCopen" (→ p. 341)		• Design Pipe Network • Pipe Network Editor • Create Cam Profile • Cam Profile Editor • "Soft Oscilloscope" (→ p. 737)	
Motors	AKM AKM2G		Kollmorgen Servomotor	• Install the AKM or AKM2G Motor. • Mechanical and Electrical installation. • Available from: • AKM Downloads • AKM2G Downloads		• AKM • AKM2G • Cartridge DDR • Direct Drive Linear • Housed DDR • KBM • TBM
PLC			"IEC 61131-3" (→ p. 286)		• "Free Form Ladder Diagram (FFLD) Editor" (→ p. 496) • "Function Block Diagram (FBD) Editor" (→ p. 487) • "Sequential Function Chart (SFC) Editor" (→ p. 505) • "Soft Oscilloscope" (→ p. 737) (Softscope) • "Structured Text (ST) / Instruction List (IL) Editor" (→ p. 511) • "Variables tab" (→ p. 102) for Dictionary	• FBD Language • FFLD Language • IL Language • SFC Language • ST Language

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Domain	Product	Product Example Image	Concept (Technology)	Installation	Tasks (Tools)	Reference
Safety			"Safety over EtherCAT" (→ p. 421)	"SafePLC2 Projects" (→ p. 219)	1. "SafePLC2 Projects" (→ p. 219) 2. "Set Up a Variable to Monitor Safety Data" (→ p. 558) 3. "AKD2G Safety Parametrization Using FSoE with SCU-1-EC and PxMM / PCMM2G" (→ p. 893) 4. "Set Up FSoE Master and an AKD2G with SafeMotion Monitor" (→ p. 891)	• "BBH FSoE Master" (→ p. 767) • "Related Documents" (→ p. 772) • AKT2G-SDI-004-000 • AKT2G-SDO-004-000

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4.3.1 Human-Machine Interface



Figure 4-10: Hardware to Display the Human-Machine Interface

A Programmable Logic Controllers (PLC) interacts with people for the purpose of configuration, alarm reporting or everyday control. A Human-Machine Interface (HMI) is employed for this purpose.

- A simple system uses buttons and lights to interact with the end-user.
- Text and graphic displays are available on the touch panels.

Most modern PLCs can communicate over a network to some other systems, such as a computer running a SCADA system.

The communication between the HMI and the PLC is based on Modbus over TCP/IP (Modbus TCP is the Ethernet version of Modbus) by means of a standard Ethernet cable that connects the two devices.

This communication is done in the background, asynchronously, at the cycle time (20-1000 milliseconds) specified in the Controller Properties. See "Configure the Controller" (➔ p. 156) for more information.

- Variables defined in the HMI to describe the interface are passed to the controller this way.
  - See "Map Variables to HMI" (➔ p. 216) for more information.
- This means there is no data coherency in the data exchange because the variables read by the Modbus do not come from the same PLC cycle.
- This data has a rather low priority.
  - It is interpreted by human feedback and should never be noticed by the user.

NOTE

Kollmorgen HMIs are limited to communicating no more frequently than every 100 milliseconds.

4.3.2 Controllers

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



Figure 4-11: Kollmorgen Controllers

These are the Kollmorgen Automation Suite controllers:

- PCMM2G - **P**rogrammable **C**ontroller **M**ulti-Axis **M**aster **2**nd **G**eneration.

4.3.2.1 PCMM2G Hardware

4.3.2.2 Controller Rotary Switch



The rotary switch on a Kollmorgen controller is set from a 0 to 9 position.

Switch Position	Description
Position 0	• The controller tries to get an IP address from a DHCP server. • If the DHCP fails, the controller uses AutoIP to get a usable IP address.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Switch Position	Description
Position 1	Either: • The default static IP address of 192.168.0.101. • A custom IP address.
Position 2-9	• The controller is pre-configured with static IP addresses. • The IP addresses range is from: • 192.168.0.102 (Position 2) to 192.168.0.109 (Position 9)

NOTE

If a DHCP server is not present, the drive assumes an Automatic Private IP Address of the form 169.254.x.x.

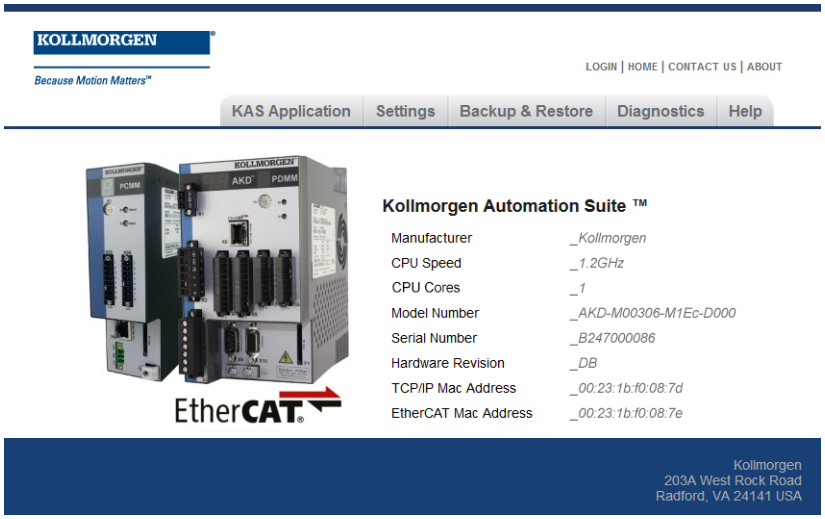
NOTE

- On the PCMM2G, if a network cable is not attached or no IP address is assigned, the controller shows an IP address of 0.0.0.0.

4.3.2.3 Controller Web Server

The controllers contain a Web server used to perform these operations:

- Read information about the controller (e.g., model type, firmware version, version of the KAS application).
- Diagnose the system (e.g., CPU speed and usage, total and free storage space, list the EtherCAT devices).
- Configure some parameters (e.g., change the IP address, upgrade the firmware).
- Interact with the application (e.g., Start and Stop the KAS application, see the logs).



See "Using the KAS Web server" (→ p. 619) for more information.

4.3.2.4 Controller Performance

Kollmorgen offers several controllers with varying performance. This topic compares the relative performance to help make a decision about what is best for your needs.

The PCMM2G improves performance by running PLC code and motion code at the same time.

- Typical performance gain varies from 130% to 300% over the single core 1.2GHz PxMM , depending on the type of application being run.
- The greatest performance gain is obtained when PLC code execution time and motion code execution time, measurable using the trace times feature, are roughly the same.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- If one of these two values dominate the execution time, the performance gain is on the smaller side.
- See "Practical Application: Using Trace Time To Measure CPU Load" (→ p. 756).

This topic has these sections:

- "Performance Graphs" (→ p. 76)
- "Pipe Network and PLCopen Applications" (→ p. 78)
- "Application Complexities" (→ p. 77)

4.3.2.4.1 Performance Graphs

These graphs are for guidance in initial controller selection for development.

- "PCMM2G Controllers" (→ p. 76)
- "PxMM Controllers" (→ p. 76)

Notes

- During development, the performance of an application should be measured to ensure that the selected controller meets the performance requirements.
- Measurements can be done with either the:
  - KAS's digital oscilloscope using the trace times traces.
  - GetCtrlPerf function block.
  - Observation of the CPU usage on the "Diagnostics tab" (→ p. 688) tab of the KAS controller's web page interface.

4.3.2.4.2.1 PCMM2G Controllers

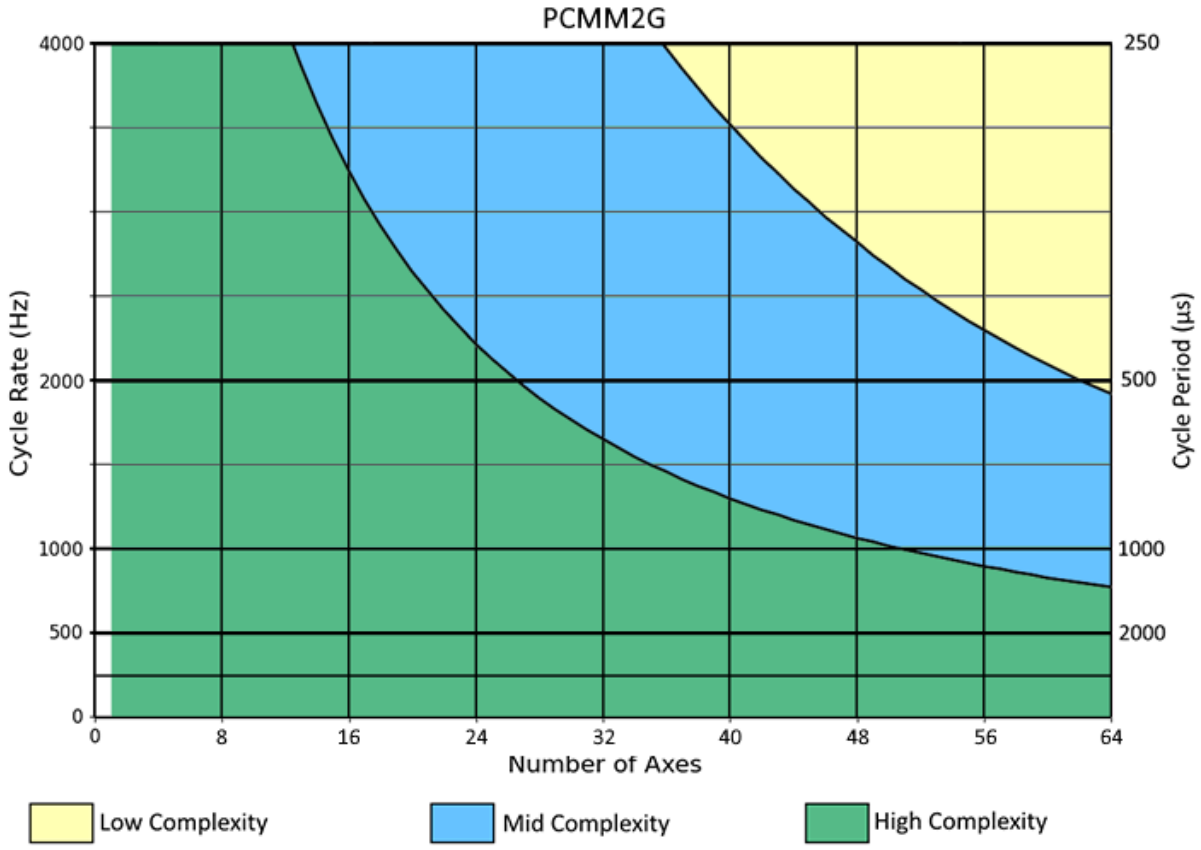


Figure 4-12: PCMM2G Controllers

4.3.2.4.3.2 PxMM Controllers

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

IMPORTANT

- AKD PDMM and PCMM are not available for new applications.
- Kollmorgen recommends the PCMM2G for new designs.

Single Core

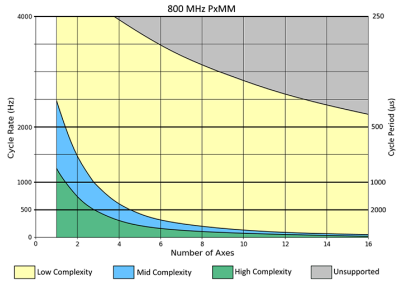


Figure 4-13: 800MHz Single-core

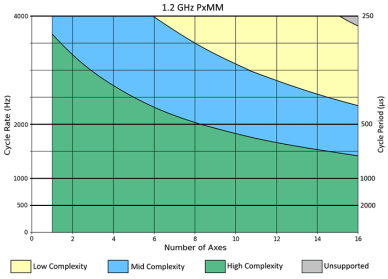


Figure 4-14: 1.2GHz Single-core

Dual Core

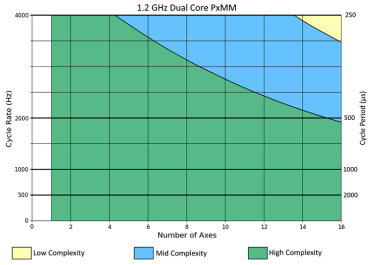


Figure 4-15: 1.2GHz Dual-core

4.3.2.4.4 Application Complexities

4.3.2.4.5.1 High Complexity

A **high-complexity** application is characterized by:

- Complex setups.
  - Most axes being slave axes.
  - Cammed or geared to a master axis.
- Frequent moves.
- Frequent reading and setting of external I/O points, and complex PLC logic.

Click for a High Complexity Example

A 4-axis high-complexity application running at 1kHz (1000us) can be handled by a single-core 1.2 GHz PxMM.

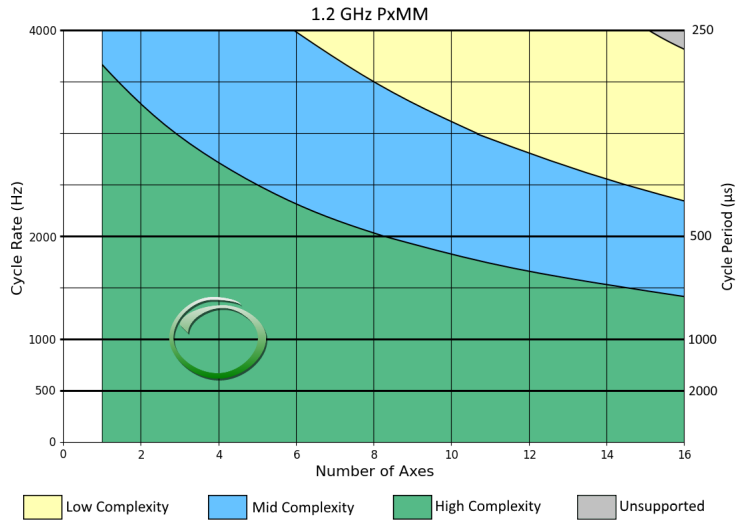


Figure 4-16: Example: High Complexity

4.3.2.4.6.2 Middle Complexity

A **middle-complexity** application is characterized by:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Being in between High and Low Complexity.
- Most axes are point-to-point or velocity moves.
  - Some axes are cammed or geared to a master.
- PLC logic is split so parts of a complex logic run at given times.

Click for Middle Complexity Example

A 12-axis middle-complexity application running at 4kHz (250us) requires the dual-core 1.2 GHz PxMM.

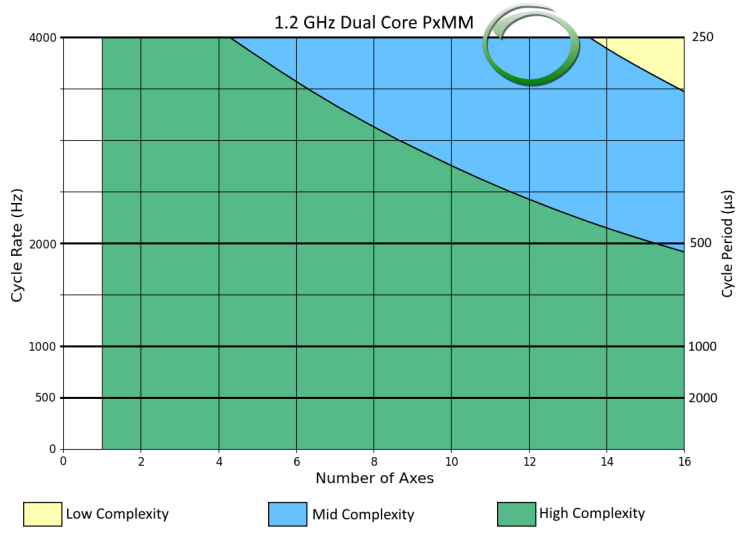


Figure 4-17: Example: Middle Complexity

4.3.2.4.7.3 Low Complexity

A **low-complexity** application is characterized by:

- Point-to-point and velocity moves.
- Occasional reading of external I/O points, and simple PLC logic.

Click for a Low Complexity Example

A 10-axis low-complexity application running at 2kHz (500us) can be handled by an 800 MHz PxMM.

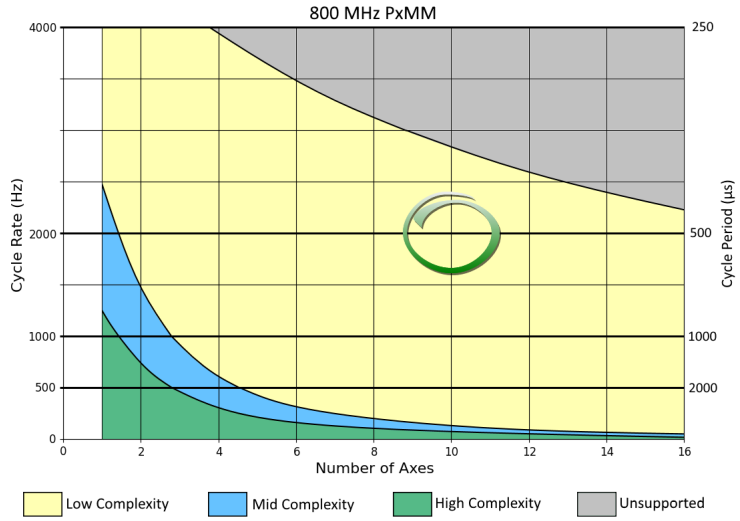


Figure 4-18: Example: Low Complexity

4.3.2.4.8 Pipe Network and PLCopen Applications

Pipe Network applications usually use less of the controller's CPU.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

When using the graphs:

- A Pipe Network application acts like a slightly "Low Complexity" (➔ p. 78) application.
- A PLCopen application acts like a slightly "High Complexity" (➔ p. 77) application.

4.3.3 Communication and Fieldbus

4.3.3.1 Fieldbus

Fieldbus allows a machine to be connected to other machines in an automation systems network.

Typically, this connection is referred to as a factory automation network connection.

4.3.3.2 Motion Bus

Motion requires the controller to frequently update the drive with new trajectory setpoints.

The bus involved in the motion control requires it to be able to handle rigid jitter and timing demands including high data throughput and low latency.

4.3.3.3 Ethernet

Ethernet is certainly the most popular communications bus today because it is used in most computer networks.

- Motion control devices using Ethernet allow high-speed connections to computers without requiring special hardware.
- This reduces the cost and time required to make high-speed connections.

4.3.3.3.1 EtherCAT

The EtherCAT technology overcomes the system limitations of other Ethernet solutions.

- The Ethernet packet is no longer received, then interpreted and copied as process data at every connection.
    - Instead, the Ethernet frame is processed on the fly.
- Each slave node reads the data addressed to it, while the telegram is forwarded to the next device.
- Similarly, input data is inserted while the telegram passes through.
    - The telegrams are only delayed by a few nanoseconds.

4.3.4 Input/Output Bus

Input/Output refers to the communication and acquisition of data between your automation system and the outside world (possibly a human, or another information processing system).

- Inputs are the signals or data received by the automation system.
- Outputs are the signals or data sent from it.

Automation systems built with KAS are interrupt-driven.

Typical interrupt uses include: system timers, disks I/O, power-off signals, and exceptions handling.



Figure 4-19: I/O Modules

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

I/O modules provide a convenient modular package which is simple to wire and add or change slice types.



Figure 4-20: Standard I/O Couplers and Slices

4.3.5 Drives

These websites provide detailed information about drives:

- AKD Series Servo Drives
- AKD2G Servo Drive
- AKD-N Decentralized Servo Drive

4.3.5.1 Related Documents

See these manuals for additional information on drives:

Drive Manuals	Description
AKD EtherCAT Manual	Describes the installation, setup, range of functions, and software protocol for the EtherCAT AKD product series.
AKD Installation Manual	• Installation manual for AKD and AKD PDMM drives. • Describes the AKD series of digital drives and includes mechanical, electrical, and software installation information needed to safely install AKD.
AKD, AKD2G, S700 (in NA) Accessories Manual	Describes the accessories for Kollmorgen digital drive systems and servo drive motors.
AKD2G EtherCAT Manual	Describes the installation, setup, range of functions, and software protocol for the AKD2G product series.
AKD2G Installation Manual	• Describes the AKD2G series of digital drives. • Includes mechanical, electrical, software, and functional safety options.
MKD Installation Manual	• Installation manual for MKD drive system modules: • MKD-C power supply. • MKD-N servo drives. • It includes mechanical, electrical and software installation information needed to safely install the MKD system modules.
S300 Reference Documentation	Kollmorgen website with access to all S300 manuals.
S700 Reference Documentation	Kollmorgen website with access to all S700 manuals.

4.3.6 Kollmorgen Motors

TIP

Click the product image to open the respective product page on Kollmorgen.com.

4.3.6.1 Servomotors

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.


AKM2G


AKM


AKMH

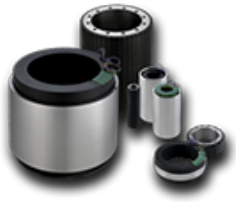

Goldline

4.3.6.2 Cartridge Motors

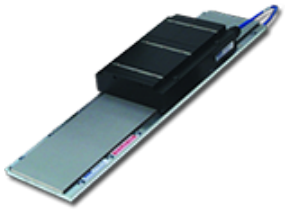

Cartridge DDR

4.3.6.3 Direct Drives


Housed DDR


KBM Frameless


TBM Frameless

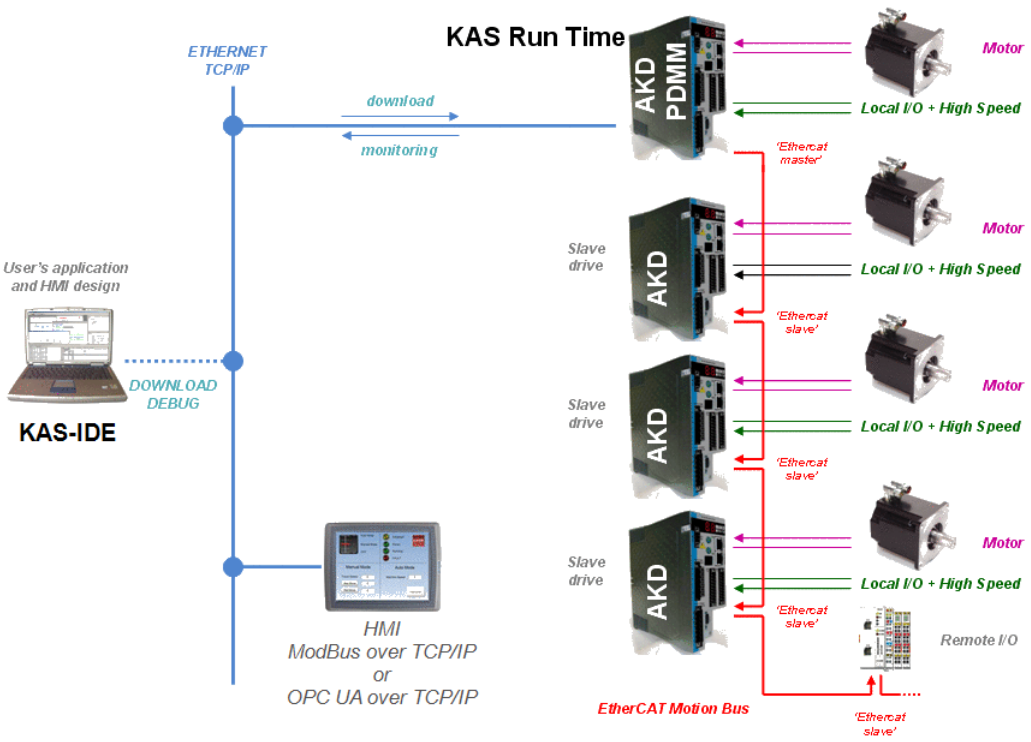

Direct Drive Linear

4.4 Implement Multi-Axis with PCMM2G

The controller can be scaled to control additional drives w/ motors and I/O devices via EtherCAT.

This configuration can manage up to 32 axes.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

5 KAS Graphical User Interface Description

This section provides an overview of the common elements of the KAS-IDE GUI.

5.1 Windows and Panels Overview 84

5.1.1 Main Window 84

5.1.2 Toolboxes 85

5.1.3 Project Explorer 85

5.1.4 Libraries 100

5.1.5 Dictionary 101

5.1.6 Data Type: STRING 116

5.1.7 Read-only Attribute 116

5.1.8 Declare an Array 117

5.1.9 Single Variable Input Value 117

5.1.10 Information and Logs window 123

5.1.11 Watch Window 138

5.1.12 Kollmorgen AKD Series Servo Drive 143

5.1.13 Status Bar 144

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

See:

- "KAS-IDE Reference" (➔ p. 453) for in-depth discussion of interface elements.
- "Using the KAS Simulator" (➔ p. 586) for KAS Simulator GUI.
- "Kollmorgen AKD Series Servo Drive" (➔ p. 143) for AKD drive GUI View.

5.1 Windows and Panels Overview

5.1.1 Main Window

The KAS-IDE interface provides an all-in-one-window integrated workspace.

The main view in the Integrated Development Environment (IDE) is a Multiple Document Interface (MDI) environment. This provides an easy-to-use and customizable view, including the ability to hide, enlarge, or overlap windows in order to optimize visibility.

The main view is saved when you exit KAS-IDE. This ensures that your workspace remains the same each time you open and use the KAS-IDE.

The KAS-IDE main window contains these items:

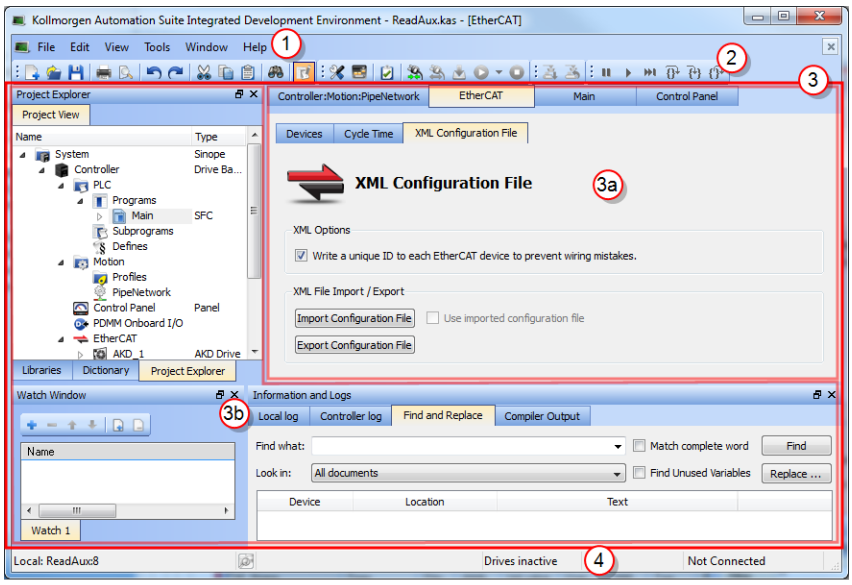


Figure 5-1: KAS Main Window

Item Number	Description
1	Menus
2	Toolbars
3	Workspace
3a	This area is specifically used for the workspace children windows.
3b	Several toolboxes A toolbox is a child window that provides some functions to perform specific tasks.
4	Status Bar - Shows the current state of the target.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

5.1.2 Toolboxes

These are the available toolboxes:

- "Project Explorer" (➔ p. 85)
- "Libraries" (➔ p. 100)
- "Information and Logs window" (➔ p. 123)

TIP

Hide or show each toolbox and toolbar directly from the contextual menus in any title bar (e.g., menu, toolbar, or toolboxes).

5.1.3 Project Explorer

The Project Explorer toolbox window displays machine application information in a tree-structure representation.

This window contains all the items used to design, implement, test, and document the application.

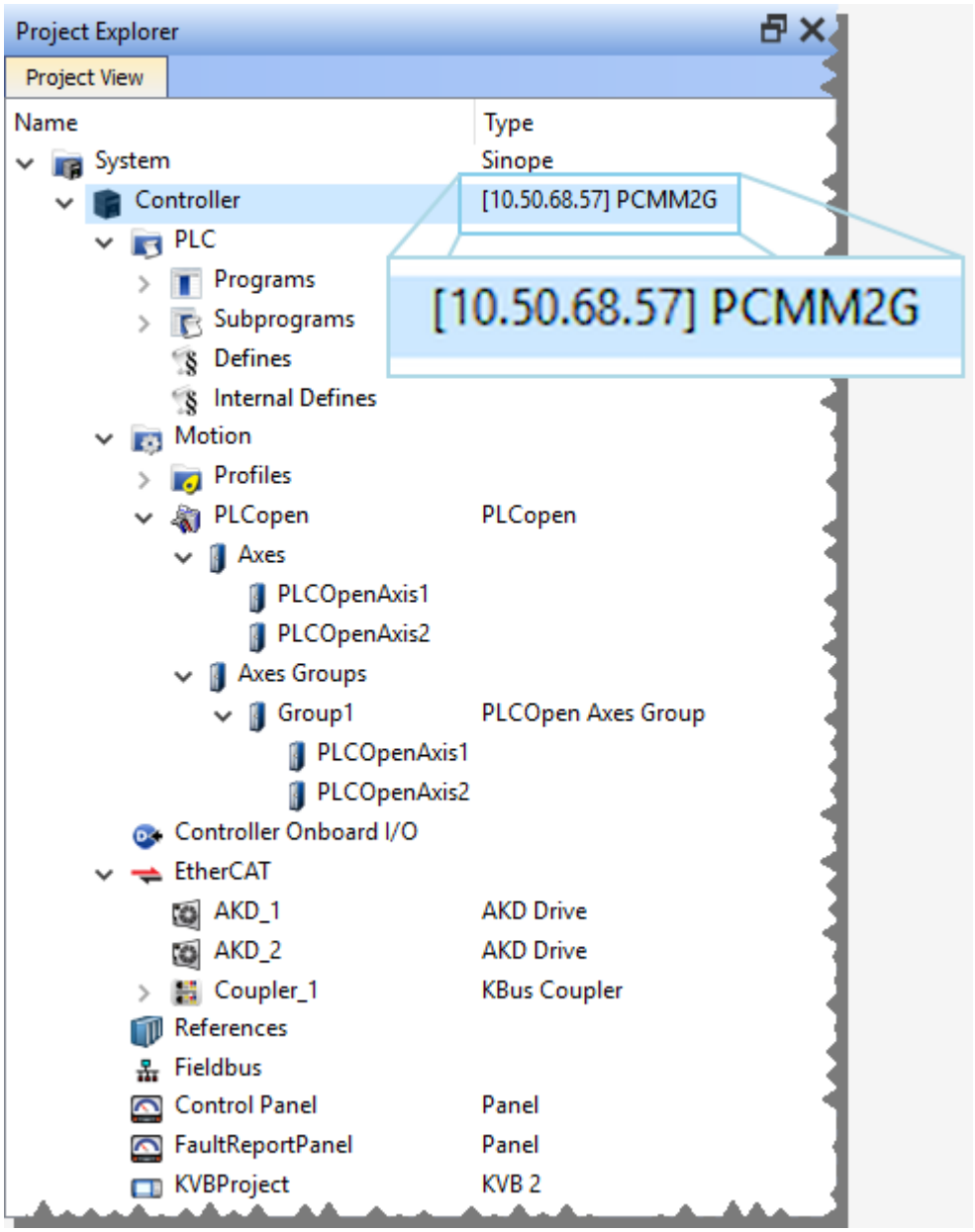


Figure 5-2: KAS Project Explorer

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Item	Description
Hardware	Devices that make up the system such as Controllers, EtherCAT Motion Bus, servo and stepper drives, HMI devices, I/O Terminals, etc.
Motion	• Pipe Networks or PLCopen • Axis objects • Cam profiles
PLC (IEC 61131-3)	• Programs that control the system. • User-defined Functions and Function Blocks.

TIP

Navigate in the project-tree by entering the item's initial letter or using the arrow keys.

A project is made of these items:

- "System" (→ p. 86)
- "Controller" (→ p. 86)
  - "PLC" (→ p. 88)
    - "Programs" (→ p. 89)
    - "Subprograms" (→ p. 90)
    - "Defines" (→ p. 91)
    - "Internal Defines" (→ p. 91)
  - "Motion" (→ p. 91)
    - "Profiles" (→ p. 91)
    - "Pipe Network" (→ p. 92) or "PLCopen" (→ p. 92)
  - "Controller Onboard I/O" (→ p. 93)
  - "EtherCAT" (→ p. 93)
    - "AKD Drive" (→ p. 94)
    - "AKT (K-Bus) I/O Coupler" (→ p. 95)
    - "Device" (→ p. 95)
      - "Module" (→ p. 96)
  - "References" (→ p. 96)
  - "Fieldbus" (→ p. 96)
  - "Control Panel" (→ p. 93)
  - "KVB Project" (→ p. 96)
  - "SafePLC2 Project" (→ p. 97)
- "HMI Device" (→ p. 96)
  - "KVB Project" (→ p. 96)

NOTE

Node Names: Allowed characters for node names include A-Z, a-z, 0-9, . (period), - (hyphen), and _ (underscore).

5.1.3.1 System

This item concerns the whole project. A right-click opens its menu that provides this option:

Command	Description
Add HMI Device	• Add a new HMI device with a KVB panel. • See "HMI Device" (→ p. 96).

5.1.3.2 Controller

A controller is composed of:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- a "PLC" (→ p. 88) item
- a "Motion" (→ p. 91) item
- a "Control Panel" (→ p. 93)
- an "EtherCAT" (→ p. 93) Motion Bus
- some "References" (→ p. 96)

This item displays the controller's IP address and controller hardware type.

- It is used to "Access the Web server From the KAS-IDE" (→ p. 97).
  - The webserver functionality may be used directly in the KAS-IDE.
- See "Using the KAS Web server" (→ p. 619).

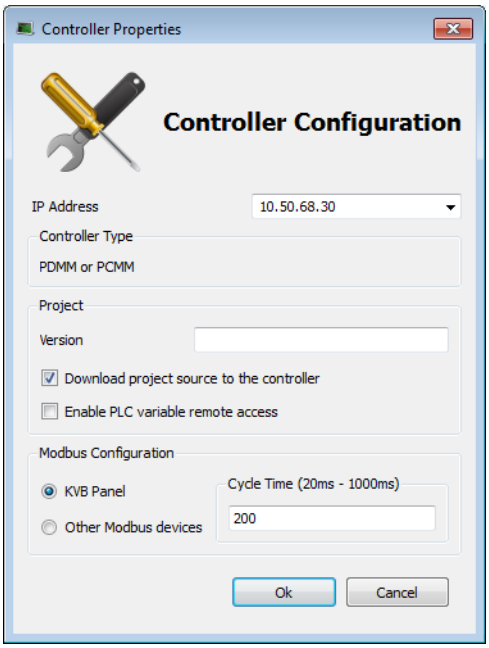
NOTE

The IP address is shown as `127.0.0.1` if the system is in simulation mode.

5.1.3.3 Controller Node - Context Menu

Command	Description
Add Control Panel	Add a new control panel to the controller. See "Control Panel" (→ p. 93).
Import Control Panel	Import a pre-configured control panel for use in the project. See "Import a Control Panel" (→ p. 99).
Add KVB Project	Add a new KVB panel which is embedded into the controller. See "KVB Project" (→ p. 96).
Import KVB Project	• Import a compressed ("zipped") KVB project, which may be created in KAS or KVB. • The system validates the compressed KVB project and adds the panel.
Add SafePLC2 Project	Add a node to the tree to access a SafePLC2 project directly from the KAS-IDE. See "SafePLC2 Projects" (→ p. 219).
Import SafePLC2 Project	Select a `.SPL2` file to import. • There may only be a single instance of a SafePLC2 project in the KAS-IDE. • If none exists, a SafePLC2 Project is added to the end of the Project tree. • If there is a project, you are prompted to overwrite the project or cancel the import. • The import cannot occur if SafePLC2 is running.
Add Fieldbus	Add a node to access the Fieldbus Editor.
Access Web Server	This command opens the web server interface in the GUI. See "Access the Web server From the KAS-IDE" (→ p. 97) and "Using the KAS Web server" (→ p. 619).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Command	Description
Properties	Open a dialog box to configure the controller. See "Configure the Controller" (→ p. 156).  **Figure 5-3:** Configure the Device **Parameters Description** • IP Address or COM: Allows for connecting to and downloading your application to the controller. Changing this value modifies the IP address for the Controller. • The last 10 IPs entered are accessible and are stored with the project. • Select an item in the list and press <Delete> to clear the entry from the list. • When you click  to choose the simulation mode, this address is disabled. • See "Device toolbar" (→ p. 470). **IMPORTANT** Do not use leading zeroes when entering an IP address. The connection to the controller automatically fails if leading zeroes are used. Example: Use **10.2.3.40** instead of **010.002.003.040**. • A version number is used to ensure both versions of your application on the KAS-IDE and the KAS Runtime are the same. • Enabling **Download project source to the controller** allows for comparing source on the controller to the source on your computer. • The **Enable PLC variable remote access** is disabled by default. • Enabling it allows users to read/write variables using an HTTP connection. • **Modbus Configuration** sets what the Modbus connects to. • **Cycle Time** specifies the update rate.

5.1.3.3.1 PLC

This item contains all the PLC (Virtual Machine) part of the controller.

These items can be present in the PLC:

- "Programs" (→ p. 89)
- "Subprograms" (→ p. 90)
- Some "Defines" (→ p. 91)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Command	Description
Libraries	Import new libraries

5.1.3.3.2 Programs

Command	Description
Cycle	Configure the cycle of the virtual machine. See "Define the PLC Cycle" (→ p. 253).
Import	Import a saved program.
New Program	Add new program items: • Free Form Ladder Diagram (FFLD) Editor • Function Block Diagram (FBD) • Instruction List (IL) • Sequential Function Chart (SFC) • Structured Text (ST)

TIP

Programs and Subprograms can be reordered by drag-and-drop.
Dragging an item up the list places it above the program you drop it on, while dragging down places it below.

5.1.3.3.3 Program Item - Contextual Menu

Command	Description
Add Action Block	• Adds an action block to this program. • See "Action Blocks (POU)" (→ p. 310). **NOTE** This option is **not** available for the SFC programs.
Import Action Block	• Import a saved action block to this program. • See "Action Blocks (POU)" (→ p. 310). • Action Blocks share the parent program's variables. • Local and global variables are **not** copied during import. • Local variables **are copied** when the parent program is imported. **NOTE** This option is **not** available for the SFC programs. Click to import all children from one project to another 1. Export each Action Block program one at a time from the existing project. 2. Save the Action Block program. • Specify a location and a name. • Do NOT use spaces in the file name. 3. Close the project. 4. Open the project to be updated. 5. Import each saved Action Block program into the project tree. 6. If applicable, rename the program.
Add Child SFC	Add a child program to this program. **NOTE** This is reserved for only the **first** SFC program.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Command	Description
Import Child SFC	Import a saved SFC program to the current program. **NOTE** This is reserved for only the **first** SFC program. How to import all children from one project to another? 1. Export each program one at a time from the existing project. 2. Save the program (specify a location and a name, avoiding spaces in the file name). 3. Close the project. 4. Open the project to be updated. 5. Import each saved program in the project tree. 6. Rename the program if needed. Only local variables are copied (not the global variables).
Export	Save the selected program to your file server, avoiding spaces in the file name.
Rename	Rename the selected program.
Delete	Delete the selected program.
Print SFC and All Level 2	Print all PLC programs. See "Print" (→ p. 282).

TIP

Double-click the program name to open the program in the workspace.

5.1.3.3.4 Subprograms

Command	Description
New Function (Subprogram)	Add a new subprogram item: • Free Form Ladder Diagram (FFLD) Editor • Function Block Diagram (FBD) • Instruction List (IL) • Structured Text (ST)
New UDFB	Add a new UDFB item: • Free Form Ladder Diagram (FFLD) Editor • Function Block Diagram (FBD) • Instruction List (IL) • Structured Text (ST)
Import	Import a saved program.

5.1.3.3.5 Subprogram Item - Context Menu

You can create your own functions as well as functional blocks that are called UDFBs (User-Defined Functional Blocks).

For each of them, use these commands:

Command	Description
Export	Save the selected subprogram onto your file server.
Rename	Rename the selected subprogram.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Command	Description
Create Unlocked Copy	Duplicate the selected, locked subprogram. The duplicate is NOT locked.
Delete	Delete the selected subprogram.
In/Out Parameters	Open the Program Properties dialog box to "Declare Functions or Function Blocks" (→ p. 521). This item is disabled for locked UDFBs.

5.1.3.3.6 Defines

- This item contains all the global definitions in the scope of the corresponding device.
  - Double-click this item to edit these global definitions.
- See "Use the Defines List" (→ p. 523).

5.1.3.3.7 Internal Defines

- This item contains all predefined definitions available for use in KAS projects.
  - Double-click this item to view these internal definitions.
- See "Internal Defines" (→ p. 524).

5.1.3.3.8 Motion

The motion item contains the motion-specific items (e.g., the Profiles and Pipe Network items).

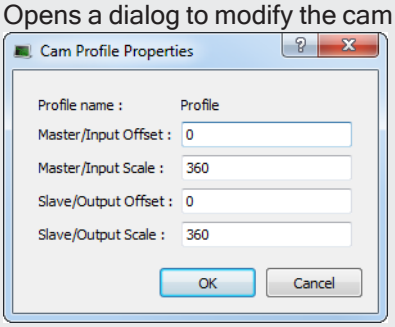
Command	Description
Motion Engines	Choose the motion engine for the application "Pipe Network or PLCopen" (→ p. 324).

5.1.3.3.9 Profiles

This item contains all the cam profiles in the project.

Command	Description
New Profile	Create a new cam profile and add it to this device (*.csv, *.cam). See "Add CAM Profiles" (→ p. 247).
Import	Import already existing cam profiles to a project.
Show Compiled Code	Show the code corresponding to the selected cam profile.

Right-click a cam profile to use these additional commands:

Command	Description
Rename	Change the cam profile to a different name.
Delete	Remove the cam profile from the list.
Export	Save the cam profile in (.CAM) format.
Properties	Opens a dialog to modify the cam profile's Master/Input Slave/Output Offset and Scale values. 

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

See "Add CAM Profiles" (→ p. 247) and "Cam Profile Editor" (→ p. 695).

5.1.3.3.10 Pipe Network

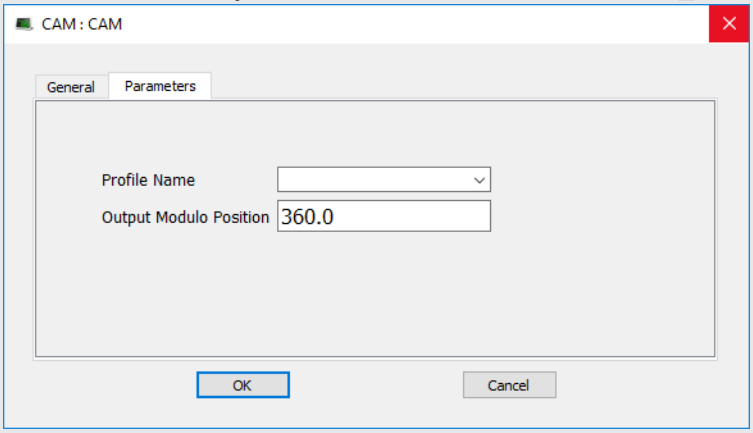
This menu applies to the Pipe Network in the project.

Command	Description
Import and Replace	This command replaces the existing Pipe Network with a pre-saved Pipe Network. A dialog box opens to locate the pre-saved file. The Pipe Network Editor is opened when the file is imported.
Export	Export the Pipe Network to a file for reuse.
Show Compiled Code	Show the code corresponding to the Pipe Network.

IMPORTANT

The existing EtherCAT axis mapping is lost when using **Import and replace**.
Profiles assigned to Cam blocks are cleared.

1. In the Project View, double-click **EtherCAT** to open the "EtherCAT Devices tab" (→ p. 536).
2. Use the tab to reassign the axes.
3. Double-click any cam of the blocks and set the **Profile_Name** parameter.



5.1.3.4 PLCopen

Command	Description
New Axis	Add a new axis to your project. See "Create a PLCopen Axis" (→ p. 195).
Show Compiled Code	Show the code corresponding to the PLCopen.

5.1.3.4.1 Axes

- Each axis is listed here, whether it is found via a scan of the EtherCAT network, or added manually.
- To add an axis manually, right-click and select **New Axis**.

For each PLCopen axis, use these commands:

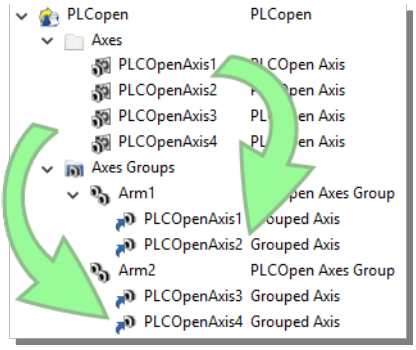
Command	Description
Properties	Open a dialog box to "Change a PLCopen Axis" (→ p. 196).
Delete	Delete the selected axis.

5.1.3.4.2 Axes Groups

Axes Groups are collections of related axes defined by the data type AXIS_GROUP_REF.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- They are typically used by coordinated motion and function blocks such as MC_SetKinTra.
- Right-click in this section to create a new Axes Group.
- Groups are populated by dragging an existing axis from the above section into the group.

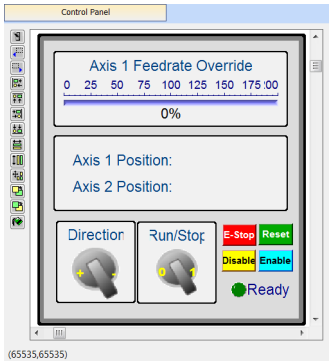


These commands are available by right-clicking on a group.

Command	Description
Delete	Delete the selected Axes Group.
Rename	Change the name of the Axes Group.

5.1.3.5 Control Panel

The Control Panel is used to provide a basic interface between you and the virtual machine.



- See "Design the HMI with the Internal Editor" (➔ p. 715).
- See "KVB Project" (➔ p. 96) for a more advanced tool to build HMI.

Command	Description
Rename	Rename the selected Control panel.
Delete	Delete the selected Control panel.
Export	Export the control panel for use in other projects. See "Export a Control Panel" (➔ p. 99).

5.1.3.5.1 Controller Onboard I/O

Command	Description
Properties	Open the Properties dialog box to configure the local I/O for the AKD PDMM or PCMM controller. See "Controller Onboard I/O" (➔ p. 616).

5.1.3.5.2 EtherCAT

This item gives access to all the devices linked to the EtherCAT Motion Bus.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Command	Description
Add Device...	In the Project view, add a Kollmorgen drive, power supply, or coupler, per a third-party EtherCAT device to the EtherCAT node.
Scan Devices	• TheKAS Runtime sends EtherCAT messages to discover the devices present in the network. • See "EtherCAT Devices tab" (→ p. 536).
Enable/Disable Online Configuration Mode	• Toggles Online Configuration Mode on and off. • See "Online Configuration Mode" (→ p. 471).
Properties	• Opens the Properties dialog box. • See "Configure EtherCAT Motion Bus" (→ p. 164).

See "Add and Configure Third-party EtherCAT Devices" (→ p. 234).

5.1.3.5.3 AKD2G Drive

Double-click an AKD2G to set its parameters.

See "Configure the Kollmorgen Servo Drive" (→ p. 221).

Command	Description
Rename	Rename the selected drive.
Delete	Delete the selected drive.
Configuration	Opens the Configuration tab for the AKD GUI.
Properties	Select the Properties menu to access the EtherCAT device's configuration views.

5.1.3.5.4 AKD Drive

Double-click an AKD to set its parameters.

See "Configure the Kollmorgen Servo Drive" (→ p. 221).

Command	Description
Rename	Rename the selected drive.
Delete	Delete the selected drive.
Configuration	Opens the Configuration tab for the AKD GUI.
Properties	Select the Properties menu to access the EtherCAT device's configuration views.

5.1.3.5.5 AKD-C Central Power Supply

Double-click an AKD-C to set its parameters.

See "Configure the Kollmorgen Servo Drive" (→ p. 221).

Command	Description
Rename	Rename the selected device.
Delete	Delete the selected device.
Properties	Select the Properties menu to access the EtherCAT device's configuration views.
Configuration	Opens the Configuration tab for the AKD GUI.

5.1.3.5.6 AKD-N Drive

Double-click an AKD-N to set its parameters.

See "Configure the Kollmorgen Servo Drive" (→ p. 221).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Command	Description
Rename	Rename the selected drive.
Delete	Delete the selected drive.
Properties	Select the Properties menu to access the EtherCAT device's configuration views.
Configuration	Opens the Configuration tab for the AKD GUI.

5.1.3.5.7 AKT (K-Bus) I/O Coupler

The Standard I/O Coupler node gives access to its I/O slices.

Command	Description
Add I/O Slice	Add a new slice (Digital or Analog Input and Output) to the selected Standard I/O Coupler.
Rename	Rename the selected coupler.
Delete	Delete the selected coupler.
Properties	Select the Properties menu to access the EtherCAT device's configuration views.

NOTE

These commands are disabled when the controller is running.
See "Troubleshooting" (➔ p. 781) about diagnosing the coupler LEDs.

5.1.3.5.8 AKT2G (EtherCAT) I/O Coupler

The Coupler node acts as a gateway for E-Bus I/O slices.

Command	Description
Rename	Rename the selected coupler.
Delete	Delete the selected coupler.
Properties	Select the Properties menu to access the EtherCAT device's configuration views.

NOTE

These commands are disabled when the controller is running.
See "Remote Input/Output Terminals" (➔ p. 777).

5.1.3.5.9 I/O Slice

- AKT (K-Bus) slices are children of the AKT I/O Coupler.
- AKT2G (E-Bus) slices are children of the EtherCAT node.

Command	Description
Rename	Rename the selected slice.
Delete	Delete the selected slice.
Properties	Open the Properties dialog box to configure the I/O slice. See "Map Input and Output to Variables" (➔ p. 526).

5.1.3.5.10.1 Device

Double-click a Device to access its EtherCAT device configuration views.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Command	Description
Rename	Rename the selected device
Delete	Delete the selected device
Add Module...	Add a module to an MDP device. See "Add Third-party EtherCAT Modules" (→ p. 236).
Properties	Access the EtherCAT device's configuration views.

5.1.3.5.11 Module

Command	Description
Rename	Rename the selected module.
Delete	Delete the selected module.

5.1.3.6 References

Used to **insert references** into a project.

Each reference is a user-defined reference that links any kind of deliverable to your project.

See "Use the Reference Folder" (→ p. 284).

Command	Description
Insert Reference	Link any kind of deliverable to your current project.
Delete	Delete the reference.
Properties	Open the referenced file in the workspace.

5.1.3.6.1 Fieldbus

This item holds the Fieldbus Editor to configure the Ethernet/IP or Profinet fieldbuses.

See "Fieldbus Editor" (→ p. 893).

5.1.3.7 HMI Device

This item holds the HMI (Human Machine Interface) item used to provide an advanced interface between you and the virtual machine.

Command	Description
Add KVB Project	• Add a new KVB panel to the controller. • This command is disabled when a KVB panel already exists. • See: • "HMI" (→ p. 776). • "Add an HMI Device" (→ p. 215) for mode details.
Import KVB Project	• Import a compressed ("zipped") KVB project, which may be created in KAS or KVB. • The system validates the compressed KVB project and adds the panel.
Rename	Rename the selected HMI device.
Delete	Delete the selected HMI device.

5.1.3.7.1 KVB Project

Command	Description
Open KVB	Opens the associated KVB project.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Command	Description
Rename	Rename the selected KVB panel.
Delete	Delete the selected KVB panel.
Export	Save a copy of the panel in a compressed (.ZIP) file.

5.1.3.8 SafePLC2 Project

Command	Description
Delete	Delete the SafePLC2 project.
Export	Select a location and name to export / create a SafePLC2 file.
Open SafePLC2	• Creates or opens a synchronized SafePLC2 project. • See "SafePLC2 Projects" (➔ p. 219).
Rename	Rename the SafePLC2 project.

5.1.3.8.1 Access the Web server From the KAS-IDE

Double-click the **Controller** to both:

- Expand/collapse the controller's components.
- Open the web server and automatically log into the administrator account.
- See "Using the KAS Web server" (➔ p. 619).

TIP

Right-click the Controller node and select **Access webserver** to access the Web server. (Figure 5-4)

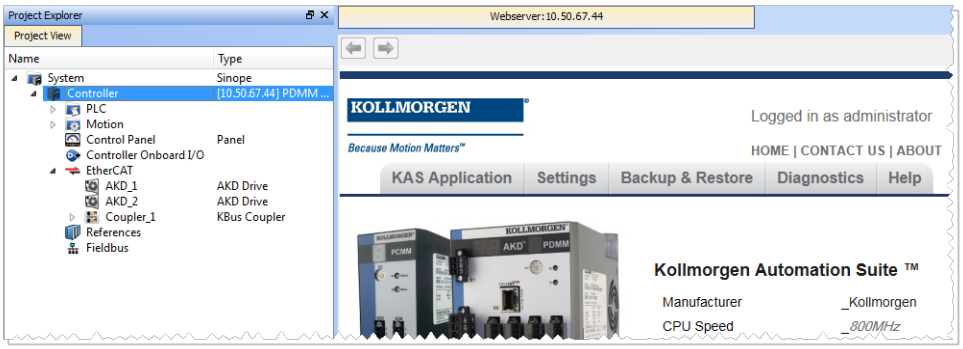


Figure 5-4: KAS-IDE Web server

IMPORTANT

KAS-IDE users are considered administrators and are automatically logged into the Web server.
You can change the password but there is no logout function.

By default, the localhost (127.0.0.1) is opened.

To set the IP address of the controller:

1. Right-click and select Properties.
2. Enter the proper **Address** and **Controller type** and click **OK**.
   The page is automatically refreshed.

If an invalid or wrong IP address is entered, this error appears (Figure 5-5):

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

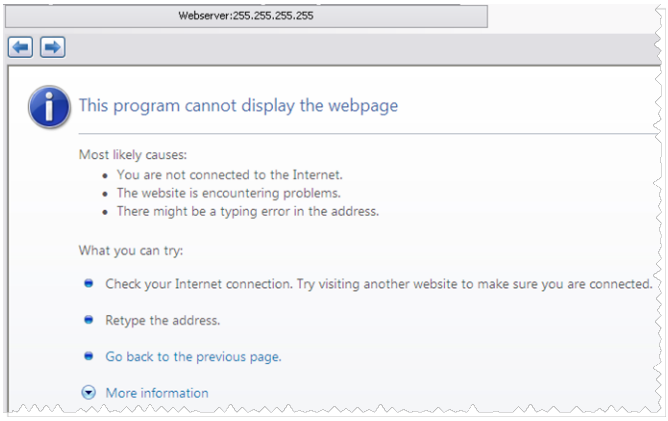


Figure 5-5: KAS-IDE Web server wrong IP address

IMPORTANT

- The KAS software uses Microsoft WebView2 Runtime software to display the documentation and web server pages from the KAS-IDE.

- This runtime software must be installed on the system to access the web server from KAS-IDE.
- If WebView2 is not installed, a message appears in the Web server window with instructions on how to download and install it.

Figure 5-6 shows the KAS-IDE Web server - WebView2 Runtime message when the runtime cannot be found.

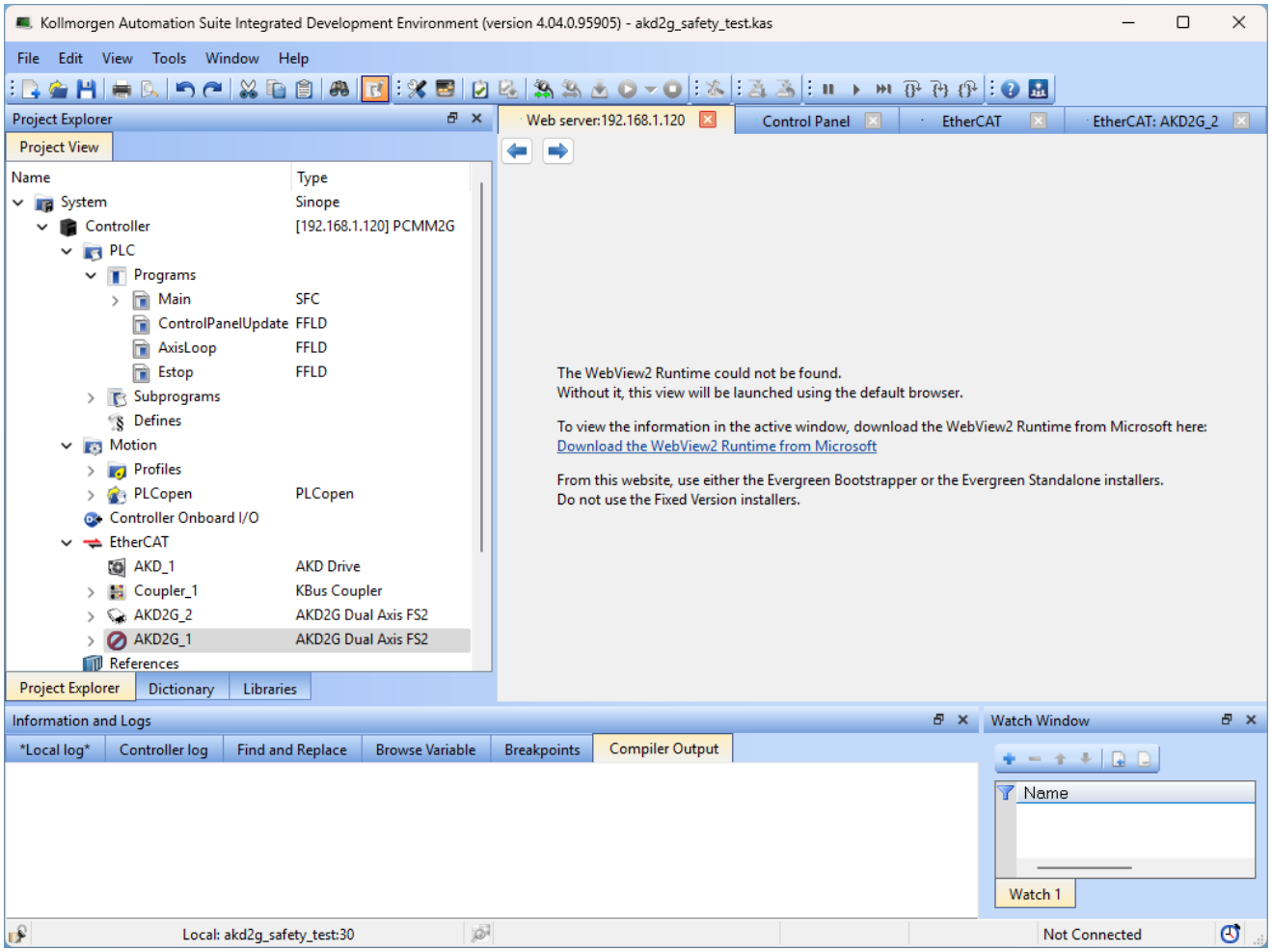


Figure 5-6: KAS-IDE Web server - WebView2 Runtime message

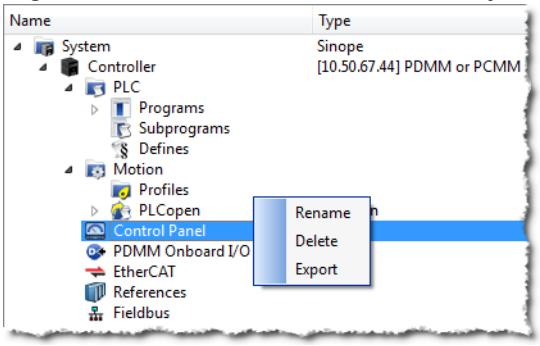
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

5.1.3.8.2.1 Export a Control Panel

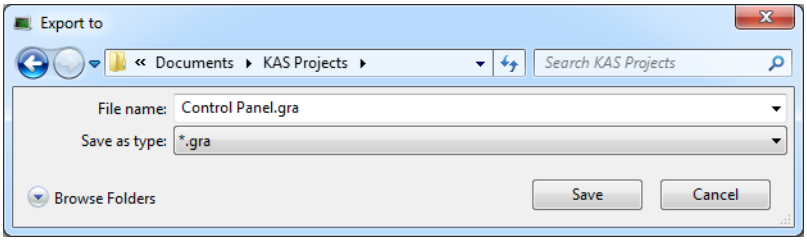
Control Panels may be exported and imported for easy reuse across projects.

Any variables associated with the control panel will be preserved across projects.

1. Right-click on the Control Panel in the Project Tree and select **Export**.



2. Name and save the .GRA file.



3. Click OK to confirm the export.

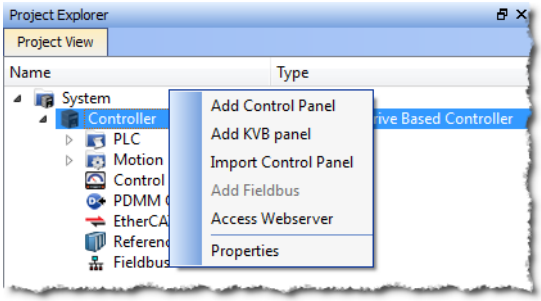
See Also

"Import a Control Panel" (→ p. 99)

5.1.3.8.3.2 Import a Control Panel

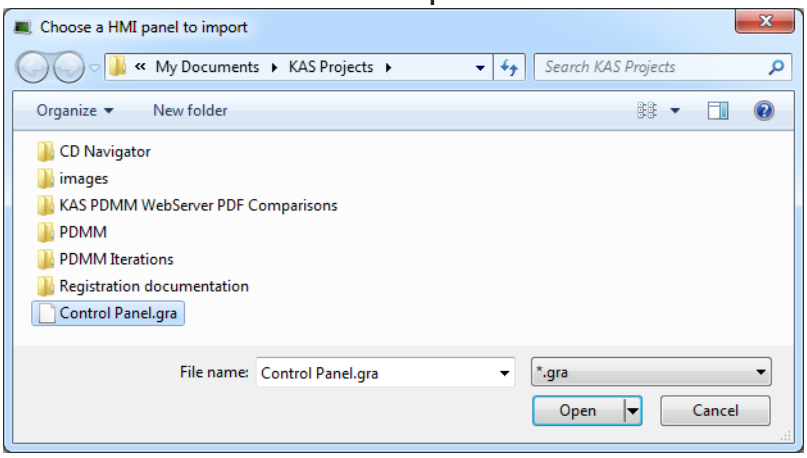
Control Panels may be exported and imported for easy reuse across projects. Any variables associated with the control panel will be preserved across projects.

1. Right-click on **Controller** in the Project View and select **Import Control Panel**.

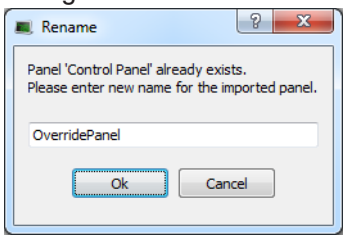


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

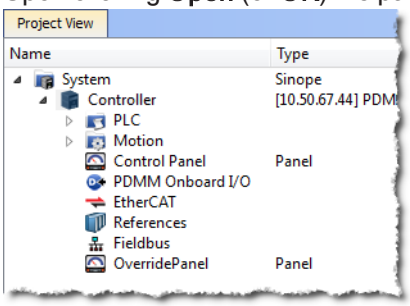
2. Locate a saved .GRA file and click **Open**.



- If the panel you are trying to import has the same name as an existing panel, you will be prompted to change the name or cancel the import.



3. Upon clicking **Open** (or **OK**) the panel will be placed in the Project View.



See also: "Export a Control Panel" (→ p. 99)

5.1.3.8.4.3 Fieldbus Editor

Usage

The Fieldbus Editor is used to configure various kinds of networked protocols.

- Each kind of fieldbus is shown as a top-level node in the configuration tree in the Project Explorer.
- Each fieldbus configuration is listed in a four level tree using this form:
  - Configuration (kind of fieldbus)
    - Communication port or master device
      - Slave device or data block
        - Connected variable

5.1.4 Libraries

This toolbox contains several tabs to access all the functions of the available libraries.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

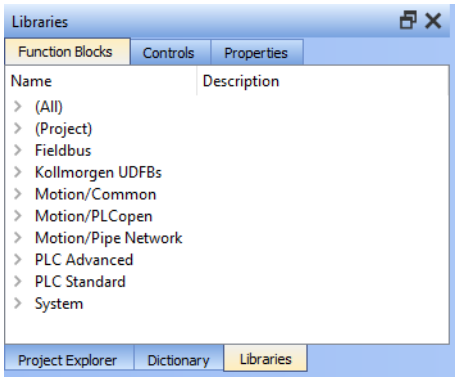


Figure 5-7: Libraries dialog box

5.1.4.1 Function Blocks tab

This tab displays all the available libraries shown in a tree-structure representation and grouped by categories. Expand a library to access all its functions. A short description of each function is available.

The **(All)** category at the top enables you to see the full list of available functions sorted in alphabetical order. The **(Project)** node contains all the UDFB and subprograms associated to the current project.

For more details about these libraries, see these library descriptions:

- Fieldbus Library
- Kollmorgen UDFBs
- Motion Library - Pipe Network
- Motion Library - PLCopen
- PLC Advanced Libraries
- PLC Standard Libraries
- System Library

TIP

It is possible to use the functions, UDFB or subprograms in PLC editors with a simple drag-and-drop operation.

NOTE

Dragging and dropping a Kollmorgen UDFB into the "Defines" editor has no effect.
However, if the Kollmorgen UDFB is already imported to the project, then it's prototype are seen in the editor.

5.1.4.2 Controls tab

This tab displays all the controls available for the HMI design.
See "Graphic Objects" (→ p. 720).

5.1.4.3 Properties tab

This tab displays all the properties of an HMI control currently selected in the HMI editor.

See (→ # 727, Graphic Objects Properties) to set the properties of an HMI widget.

5.1.5 Dictionary

The Dictionary toolbox is used to show all the identifiers (i.e., variables, data types, sub-routines, etc.) defined in the project.

These are the tabs in the Dictionary:

- "Bit Fields tab" (→ p. 119)
- "Enums tab" (→ p. 117)
- "Variables tab" (→ p. 102)

5.1.5.1 Variables tab

The Variables tab is used to show all the variables defined in the project.

The variable details are displayed in order to show the variable types, dimensions, attributes, etc.

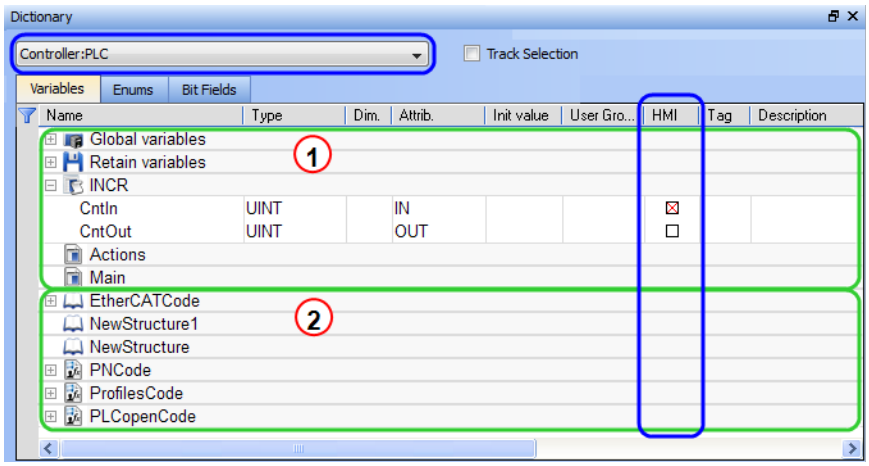


Figure 5-8: Variables tab

The list of variables is split into two parts:

Item Number	Description
1	All the Variables are at the top.
2	All the Structures are at the bottom.

NOTE

See "Call Functions or Function Blocks" (→ p. 523) about the procedure to create an instance of a structure.

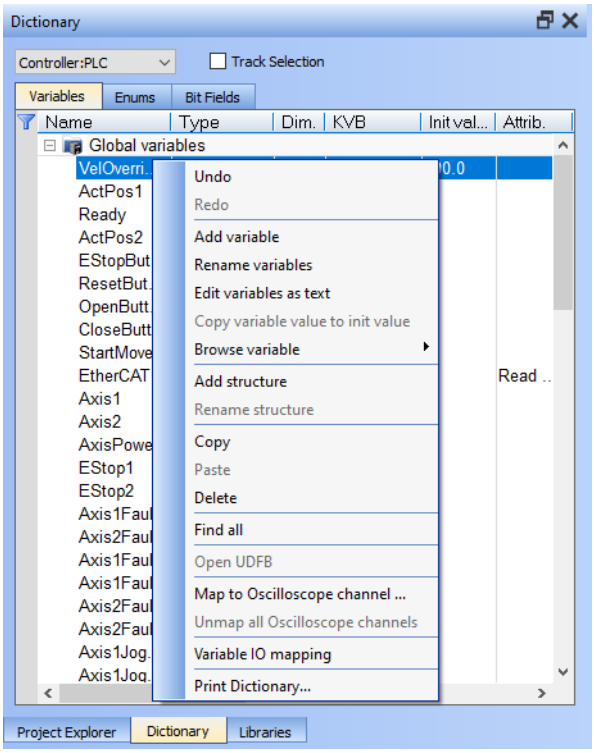
TIP

To show all the variables of all programs, select '**PLC**' in the project tree.

The Dictionary's Context Menu

Right-click in the Dictionary window to open the menu:

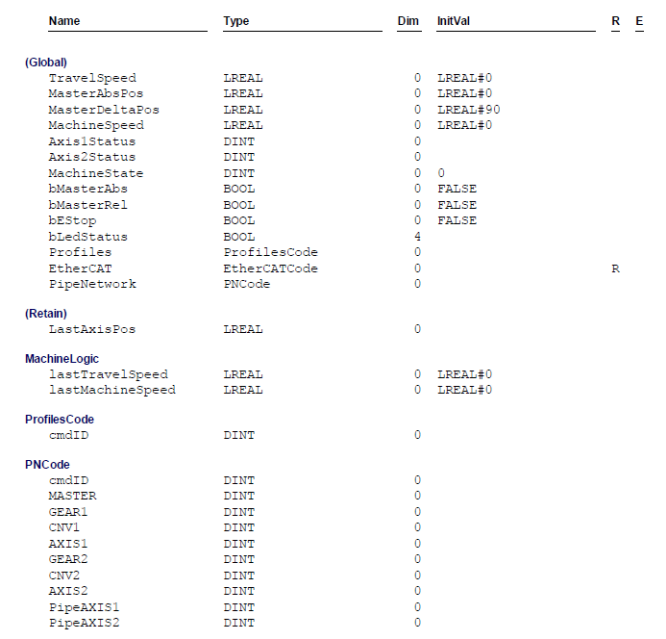
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



This menu is used to perform these actions:

Command	Description
Undo	Undo the last action performed on the Dictionary.
Redo	Redo the last undone action.
Add Variable	Add a new variable in the selected level (e.g., Global, Retain, program). • This automatically creates a new variable called NewVar with type BOOL. • For a Function or UDFB, you can specify input and output parameters. • See "Define Parameters and Private Variables" (➔ p. 522).
Rename variables	Used to either replace a section of matching variables or append text to the variable's names. See "Rename Variables" (➔ p. 111).
Edit variables as text	Opens a text editing dialog used to edit the variables found in the group as text using IEC 61131-3 syntax. See "Edit Variables as Text" (➔ p. 519).
Copy variable value to init value	Copies the current value of a retain variable to the variable's initial value. See "Retain Variables" (➔ p. 292) .
Browse variable	Used to browse all instances of a variable. See "Browse Variable tab" (➔ p. 133).
Add Structure	Used to add a new complex type. A structure named NewStructure is created and variables can be dragged into it. See "Complex Structures" (➔ p. 517).
Rename Structure	Used to rename the selected structure.
Copy	Used to copy a variable.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Command	Description
Paste	Used to paste the copied variable to the selected level.
Delete	Used to delete the selected variable. Press <Delete> on the keyboard.
Find all	Used to find all instances of the specified variable and opens the results in the Find and Replace tab.
Open UDFB	Used to open the selected UDFB instance. See "Monitoring UDFBs" (→ p. 279).
Map to Oscilloscope channel ...	Used to plug the selected variable on a channel. • This command opens a dialog used to configure the variable plug operation. • This command is enabled when the application is connected and running, and if the "Data Types" (→ p. 286) of variable is eligible for the Softscope as long as they are not in a UDFB instance: • Data Types: BOOL, BYTE, DINT, DWORD, INT, LINT, LREAL, LWORD, SINT, TIME, UDINT, UINT, ULINT, USINT, WORD.
Unmap all Oscilloscope Channels	Used to unplug all plugged probes from the Softscope.
Variable I/O mapping	Used to connect a variable to an I/O.
Print Dictionary ...	Used to print all the variables displayed in the Dictionary and sorted by programs. The columns show the Name, Type, Dimension, Initial Value, and Attributes. 

Name	Type	Dim	InitVal	R	E
(Global)					
TravelSpeed	LREAL	0	LREAL#0		
MasterAbsPos	LREAL	0	LREAL#0		
MasterDeltaPos	LREAL	0	LREAL#90		
MachineSpeed	LREAL	0	LREAL#0		
Axis1Status	DINT	0			
Axis2Status	DINT	0			
MachineState	DINT	0	0		
bMasterAbs	BOOL	0	FALSE		
bMasterRel	BOOL	0	FALSE		
bEStop	BOOL	0	FALSE		
bLedStatus	BOOL	4			
Profiles	ProfilesCode	0			
EtherCAT	EtherCATCode	0		R	
PipeNetwork	PNCode	0			
(Retain)					
LastAxisPos	LREAL	0			
MachineLogic					
lastTravelSpeed	LREAL	0	LREAL#0		
lastMachineSpeed	LREAL	0	LREAL#0		
ProfilesCode					
cmdID	DINT	0			
PNCode					
cmdID	DINT	0			
MASTER	DINT	0			
GEAR1	DINT	0			
CNV1	DINT	0			
AXIS1	DINT	0			
GEAR2	DINT	0			
CNV2	DINT	0			
AXIS2	DINT	0			
PipeAXIS1	DINT	0			
PipeAXIS2	DINT	0			

What is the purpose of the Track Selection check box?

The **Track Selection** filters the displayed items in the dictionary to show only items linked to the current PLC selected program:

- **Unchecked**: All project variables are displayed. This is the default setting.
- **Selected**: The variables in the Dictionary are filtered to show only those relevant to the PLC item currently selected in the project tree.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Along with the Global, retains and variables related to the selected program or UDFB, structure definitions are shown.
- The dictionary content changes accordingly if another PLC program is selected in the project tree.

The two editing modes for editing a variable.

These are the modes when editing a variable in the Dictionary:

- **Cell**: Only the selected cell is active. (Figure 5-9)

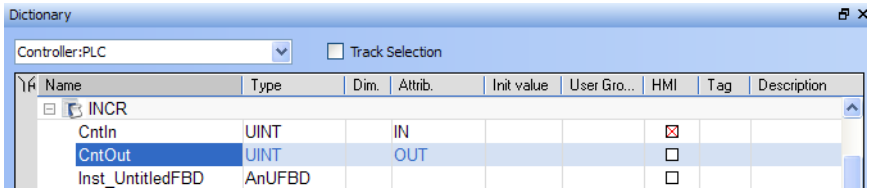


Figure 5-9: One active cell

- **Row**: The entire row is active. (Figure 5-10)

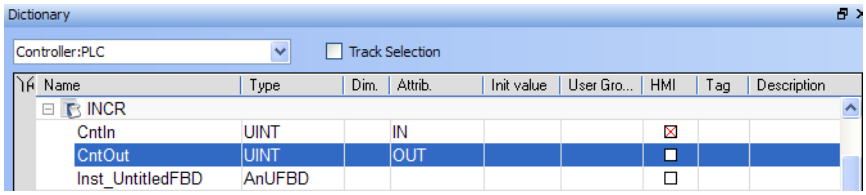


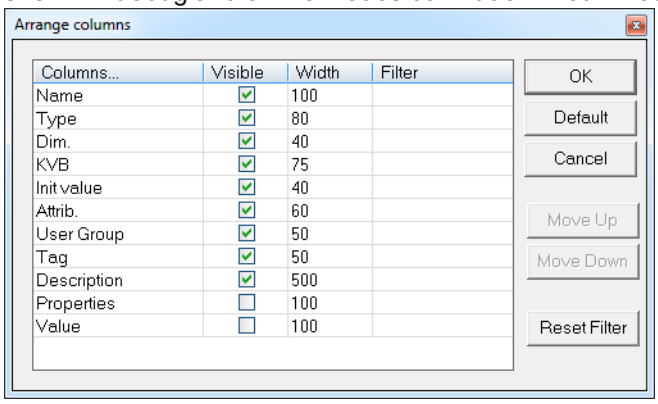
Figure 5-10: Entire active row

Press the Spacebar to toggle the selection mode from cell to row.

See "Table Keyboard Shortcuts" (→ p. 480).

How can variables be arranged and/or sorted?

- The columns in the Variables tab can be reordered, resized, and hidden by double clicking on the filter icon in top left corner of the table.
  - This opens a dialog box used to modify the table's appearance.
  - These settings persist until you change them.
  - The Visible box for the Value field cannot be changed because the Value column is automatically shown in debug and online modes but hidden in edit mode.



- Sort the list of variables in the table by either:
  - Verify you are in **cell** editing mode (press the **Spacebar** to toggle from one mode to the other).
  - Click the header of the column to use as the key sort order.

Modify Parameters of a Variable

Press the **Spacebar** to toggle to the relevant edition mode.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Mode	Description
One Parameter	In the **cell** edition mode, double-click on the parameter.
All the parameters are at the same time	In the **row** edition mode, double-click in any parameter to open the dialog box for variable configuration. See "Variables" (→ p. 106) about parameters.

NOTE

A variable cannot be changed when the KAS-IDE is connected to the controller.

5.1.5.1.1 Variables

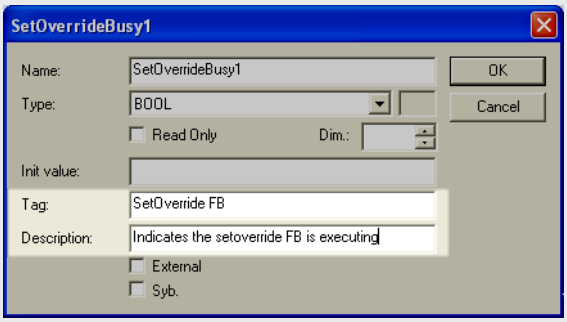
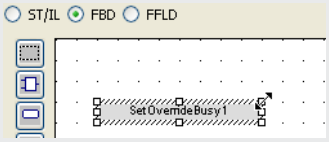
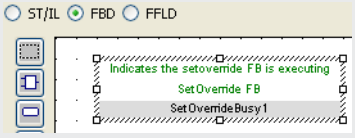
Variables in the entire project are grouped by:

Variable	Description
Global Variables	Lists all global variables used and accessible throughout the entire program.
Program Variables	Lists the variables related to your specific selected program.
"Retain Variables" (→ p. 292)	Lists all variables to be retained when the system is powered down.

For each variable, the Dictionary toolbox is used to set these parameters:

Field	Description
Attributes	The variable attributes are Read-only, External, IN, OUT. • Read-only: A variable set as Read Only is a constant. • It cannot be modified in the PLC code, but it can be manually forced. • See "Force a Variable" (→ p. 279). • Read-onlyvariables can be mapped to Inputs but not to Outputs. • Inputs can change state and a Read-only variable cannot change its value to match the input state. • External: this attribute is not used. • IN or OUT: Input or Output parameters of User Defined Function Blocks
Description	The variable description is a long comment text that describes the variable.
Dim.	• To declare an array, specify dimensions for an internal variable. • See "Arrays" (→ p. 287).
HMI	• Select variables to be used in HMI. • See "Map Variables to HMI" (→ p. 216).
Init value	• The variable initial value when you start your application. • See "Variable Initial Value" (→ p. 115).
Name	The variable name.
Syb.	Reserved.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

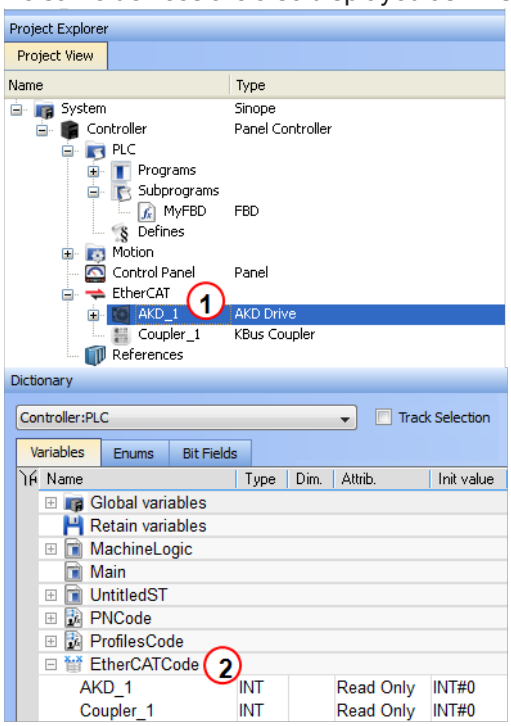
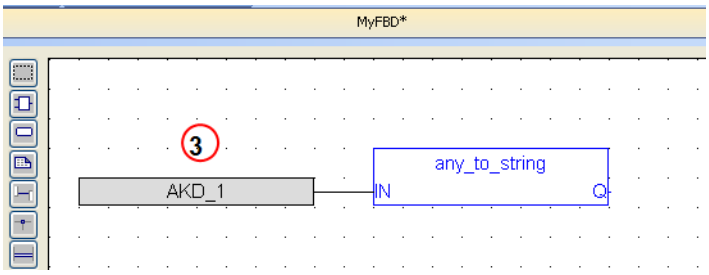
Field	Description
Tag	The variable tag is a short comment, that can be displayed together with the variable name in graphical editors. Edit the variable parameters.  Add the variable to your FBD program.  Resize the rectangle to make the Tag and Description visible. 
Type	• The variable type (can also be UDFB or complex structure). • See "Data Types" (➔ p. 286).
User Group	The variable user group (used for sorting variables).
Value	• All the variables in the Dictionary are animated with real-time values [1] • This column is only appears when the application is running. • See "Variable Monitoring" (➔ p. 278).

5.1.5.1.2 Structures

All the **structures** within the entire system project are grouped as follows:

[1] To better track variables in Running mode, the KAS-IDE dynamically computes their value along with the application execution and display the result in this column.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Structure	Description
EtherCATCode	List all the devices as PLC variables so you can use their names in FBs instance • The devices are displayed under the EtherCAT node in the Project Explorer (see call out 1) • The same devices are also displayed as PLC variables in the Dictionary 2  • You can use them in your PLC programs with a drag-and-drop operation 3 
PNCode	List all the variables related to the specific Pipe Network Code
ProfilesCode	List all the variables related to the specific profiles
PLCopenCode	List all the variables related to the specific PLCopen Code
UDFB variables	List the variables related to a specific UDFB

5.1.5.1.3 Variable Editor

Variables are declared in the Dictionary of the KAS-IDE main window.

The variable editor is a table used to declare all variables of the application.

Variables in the editor are sorted by groups:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Global variables
- Retain non-volatile global variables
- I/O variables
  - Each I/O device is a group
- Variables local to a program
  - Including in and out parameters in case of a UDFB.
- See the description of "Variables" (→ p. 290) in the language reference for a more detailed overview.

Each group is marked with a gray header in the variable list.

The – or + icon on the left of the group header is used to expand or collapse the group. (Figure 5-11)

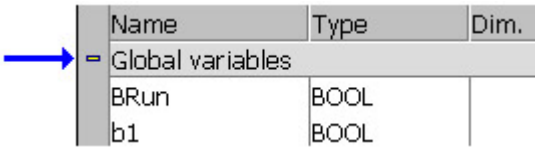


Figure 5-11: Expand - Collapse group

See how to:

- "Create a New Variable" (→ p. 109).
- Use the "Variable Table List" (→ p. 109).
- "Define Structures" (→ p. 110).
- Set "Bookmarks" (→ p. 481).

5.1.5.1.4.1 Create a New Variable

Click the **INSERT** key in the variable editor to create a new variable in the selected group.

- The variable is added at the end of the group. Variables are created with a default name.
- Use the Variable Editor to rename a new variable or change its attributes.

NOTE

You cannot insert a new variable in an I/O group.

In case of a group corresponding to local variables of a UDFB, clicking the **INSERT** key gives you the choice between:

- Add an IN (input) parameter.
- Add an OUT (output) parameter.
- Add a private variable.

IN and OUT parameters always appear at the beginning of a UDFB group.

5.1.5.1.5 Variable Table List

These are the modes when editing a variable in the Dictionary:

- **Cell**: Only the selected cell is active. (Figure 5-12)

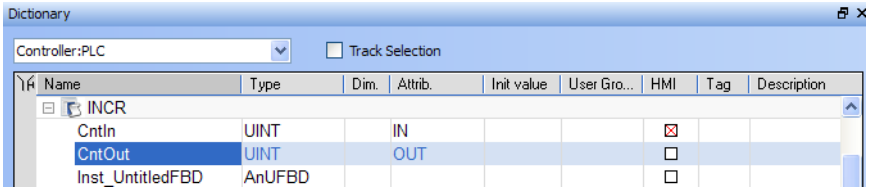


Figure 5-12: One active cell

- **Row**: The entire row is active. (Figure 5-13)

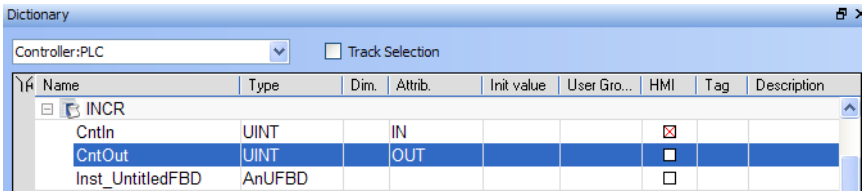


Figure 5-13: Entire active row

Press the Spacebar to toggle the selection mode from cell to row.

See "Table Keyboard Shortcuts" (➔ p. 480).

When the selection mode is on cell, the variable editor enables you to enter each piece of information directly in the cell.

Double-click or press <Enter> to open the dialog box.

Use the mouse to drag the column separators in the main table header to resize the column widths.

Press these keys for browsing groups of variables:

Shortcut	Description
Ctrl + Page Up	Move the selection to the head of the previous group.
Ctrl + Page Down	Move the selection to the head of the following group.

For Table manipulation, see "Windows Standard Conventions" (➔ p. 472).

5.1.5.1.6 Sort Variables

You can sort variables of a group according to their name, type, or dimension.

1. Move the cursor to the header of the group.
2. Click the name of the column to sort.

- The KAS-IDE always keeps the original order of declared variables.
  - This allows safe online change.
- Each time a new variable is inserted or expanded / collapsed in a group, the original sorting is re-applied.

5.1.5.1.7 Define Structures

To create a new type of data structure, use the **Add structure** command.

For more details of the full procedure, see (➔ # 517, Complex Structures).

- Each structure is represented as a group in the dictionary grid.
  - Enter the members of the structure in its group in the same way variables are entered in another group.
- New data structures are created with default names.
  - Use the **Rename structure** command to change its name.
- If a member of a structure is an instance of another structure, the nested structure must be declared BEFORE in the list.

5.1.5.1.8 Name a Variable

Use this procedure to change the name of the variable:

1. In the **Name** column of the table, select the cell to edit.
2. Press ENTER (or press the first character of the new name).
3. Enter the name in the small box.
4. Press ENTER to validate the name or ESCAPE to cancel the change.

A variable must be identified by a unique name within its parent group.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The variable name cannot:
  - Be a reserved keyword of the programming languages.
  - Have the same name as a standard or C function or function block.
- A variable must not have the same name as a program or a user-defined function block.
- The name of a variable must begin by a letter or an underscore (_), followed by letters, digits, or underscore marks.
  - Two consecutive underscores in a variable name is **not** allowed.
- Naming is case-insensitive.
  - Two names with different cases are considered as the same.

5.1.5.1.9.1 Name Physical I/Os

Each I/O channel has a predefined symbol that reflects its physical location.

- This symbol begins with:
  - **%I** for an input.
  - **%Q** for an output.
  - The symbol is followed by a letter identifying the physical size of the data.
    - See "Variables" (→ p. 290).

5.1.5.1.10.2 Change the Name of an I/O Variable

You cannot change the **%...** name of an I/O variable.

- This name is directly allocated according to the I/O devices defined in the I/O device list.
- You can give an alias (a readable name) to each I/O channel.
  - In that case, either the **%** name or the alias can be used in programs.
  - The alias must fit to the same rules as a variable name.
  - When an alias is defined for a variable, both **%...** name and alias are shown in the **Name** column of the grid.

5.1.5.1.11 Rename Variables

Variables may be renamed from within the Dictionary.

- The renaming function can either replace a section of the name or append text to variable names.
- The search will find and replace matches first within the Dictionary and will then continue the search within programs in the current project.
- You will have the option to propagate the changes to programs or not.

Procedure

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. Right-click on any variable in the Dictionary and select **Rename variables**. (Figure 5-14)

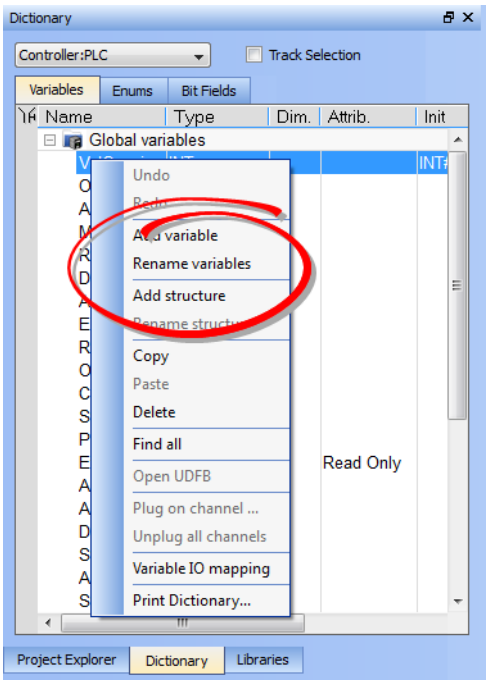


Figure 5-14: Rename variables option

The Rename Variables dialog box opens. (Figure 5-15)

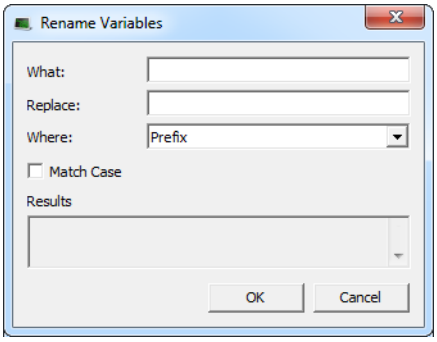


Figure 5-15: Rename Variables dialog box

2. In the **What** text box, enter the phrase to be replaced
3. In the **Replace** text box, enter the new phrase.
4. Click the **Where** drop-down arrow and select the replacement method.
   See "Where" (→ p. 114) for the options and their results.
5. Click OK to make the changes shown in the **Results** frame.

   When matches exist within programs, the **Replace in files** dialog box opens. (Figure 5-16)

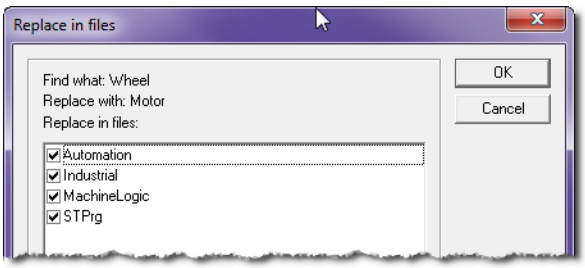


Figure 5-16: Replace in files dialog box

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6. Use the dialog box to choose which programs to modify.
7. Click **OK** to save the changes or selections and close the dialog box.
   The programs are modified in the order they appear in the list.

Controls in the Rename Variables dialog box

Control	What it Means
What	Enter the variable name/text to be replaced.
Replace	This new text replaces the current variable name text.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Control	What it Means
Where	Choose where the replace occurs:

Everywhere	Searches the entire variable name. (Figure 5-17)

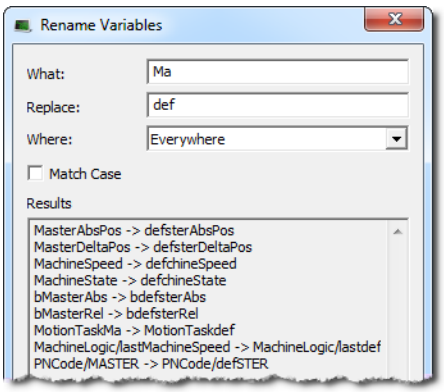


Figure 5-17: Rename Variables dialog box - Where = Everywhere

Prefix	The search/replace starts from the first character in the variable name.

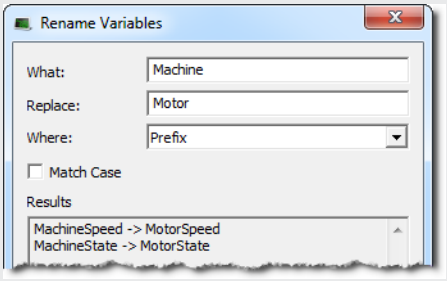


Figure 5-18: Rename Variables dialog box - Where = Prefix

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Control	What it Means
	Suffix — The **Replace** text is appended to the **What** text, completely replacing any matches. (Figure 5-19 and Figure 5-20)

IMPORTANT

Be aware that this search method completely replaces the search string.
Use this method with caution.

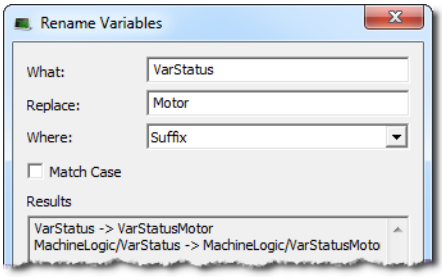


Figure 5-19: Example 1: Rename Variables dialog box - Where = Suffix

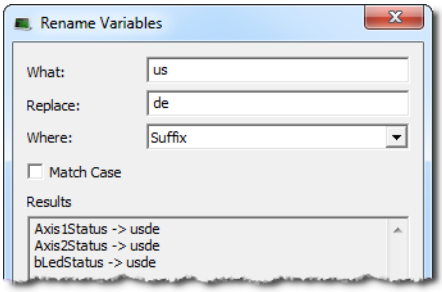


Figure 5-20: Example 2: Rename Variables dialog box - Where = Suffix

Match Case	The search is case sensitive.
Results	The search results and proposed changes are shown in this box.

5.1.5.1.12 Variable Initial Value

A variable can have an initial value.

- The value must have valid "Constant Expressions" (→ p. 301) that fits to the "Data Types" (→ p. 286) of the variable.
- The initial value is displayed in red if it is not a valid expression for the selected data type.
- There is no initial value for arrays and instances of function blocks.
- Change the initial value of a variable using the "Variables tab" (→ p. 102).

NOTE

A retain variable's value may be assigned to the variable's initial value.
See "Retain Variables" (→ p. 292) for more information.

5.1.5.1.13 Variable Tag and Description

For each variable, the KAS-IDE enables you to enter in the dictionary two strings that describe the variable:

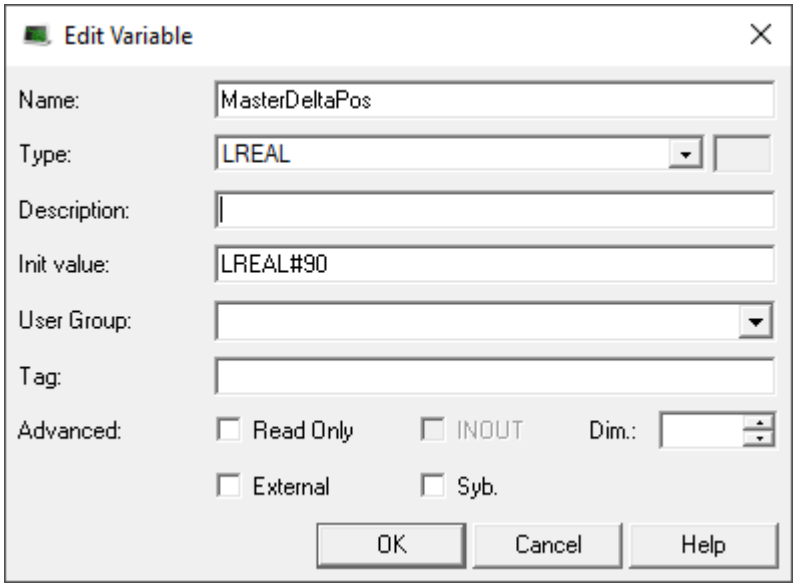
- The **Tag** is a short comment, that can be displayed together with the variable name in graphic languages.
- The **Description** is a long comment text that describes the variable.

5.1.5.1.14.1 Change the Tag or Description of a Variable

1. Enable the modification mode to **Row**.
   See "Variable Table List" (→ p. 109) for more information.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

2. Move the cursor to the corresponding cell.
3. Press <Enter> to save the new text.



5.1.5.1.15 Edit Variables as Text using IEC 61131-3 Syntax

Using IEC 61131-3 syntax, variables are declared within structured blocks.

Each block begins with `VAR`, `VAR_INPUT`, `VAR_OUTPUT`, or `VAR_EXTERNAL` keyword and ending with `END_VAR` keyword (with no semicolon after).

This table provides the meaning of each keyword:

Keyword	Meaning
VAR	• Memory variables. • Can be global, local, or retain depending on the edited group. • See "Retain Variables" (→ p. 292).
VAR_EXTERNAL	• External variables. • Can be global or local depending on the edited group.
VAR_INPUT	• Input parameters of a block. • Available only when the edited group is a UDFB.
VAR_OUTPUT	• Output parameters of a block. • Available only when the edited group is a UDFB.

5.1.5.1.16.1 Basic Syntax for Declaring a Variable

To declare a variable, enter its symbol followed by a colon (:) and its data type.

5.1.6 Data Type: STRING

If the data type is STRING, it must be followed with the maximum length in parentheses (e.g., `STRING (255)`).

```
MyVar : BOOL;
MyString : STRING(255);
```

5.1.7 Read-only Attribute

To indicate that a variable has the **read-only** attribute, insert the `CONSTANT` keyword at the beginning of the variable declaration:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
CONSTANT VarName : DataType;
```

5.1.8 Declare an Array

To declare an array, the data type must be preceded by `ARRAY [ dimensions ] OF`.

- There are at most three dimensions, separated by commas.
- Each dimension is specified as **0 .. MaxBound** (e.g., `0 .. 9`).

```
Array1 : ARRAY [0 .. 99] OF DINT;
Matrix : ARRAY [0 .. 9, 0 .. 9, 0 .. 9] OF REAL;
```

5.1.9 Single Variable Input Value

You can specify an initial value for single variables.

- The initial value is entered after the data type and is preceded by `:=`.
- The initial value must be a valid constant expression that fits the data type.

```
MyBool : BOOL := TRUE;
MyString : STRING(80) := 'Hello';
MyLongReal : LREAL := lreal#1.0E300;
```

5.1.9.0.1.1 Additional Information and Description Texts

Variables may have additional properties and comment texts in the KAS-IDE.

Special directives are entered as IEC comments AFTER the declaration of the variable, to specify additional info.

These directives are available:

Directive	Description
(*$desc=Text*)	Variable description
(*$embed=Text*)	Variable embedded properties (the syntax is the one shown in the variable grid, in the "Property" column)
(*$profile=ProfileName*)	Variable embedded profile
(*$tag=Text*)	Variable tag (short comment)

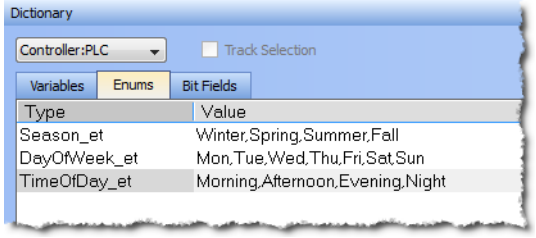
Optional: Use `//` single line comments to enter the directives:

```
//$tag=Text
//$desc=Text
//$profile=ProfileName
//$embed=Text
```

5.1.9.1 Enums tab

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This tab allows you to define enums (Enumerated Types). An enumerated type allows you to define a data type and assign a specific set of accepted values.



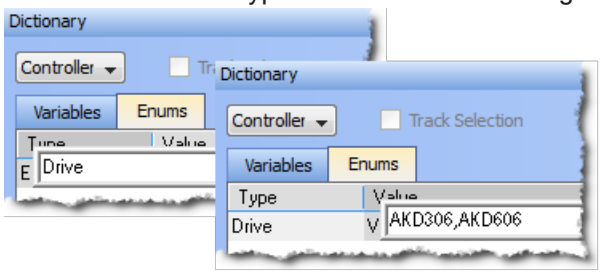
5.1.9.1.1 Adding Enums

An enum may be created in one of these ways:

- Right-click within the Enum tab and select **Insert Enum**.
- Press the **Insert** key while the Enum tab is active.

This creates a default enum labelled as "`EnumTypen`" with the Value of "`V0,V1`".

Double click on the Type or Value to make changes.

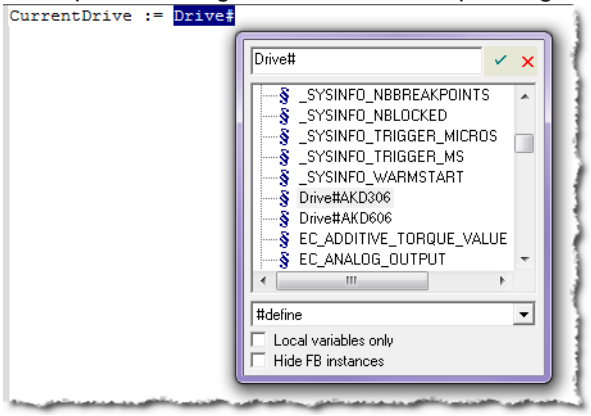


NOTE

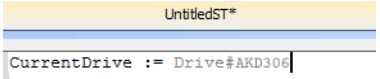
- An enum must contain two or more values.
- If only one value is defined, it is not considered and the previous value is retained.
- An enum cannot contain special characters (e.g., #, @, etc.)
- If a special character is defined in the value field, it is not considered and the previous value is retained.

5.1.9.1.2 Create Enums

1. Enter the enum type and hash (#), the press Ctrl+Space.
   This opens a dialog to select the corresponding value.



2. Select the value and click the check mark.
   The value is loaded into the program.

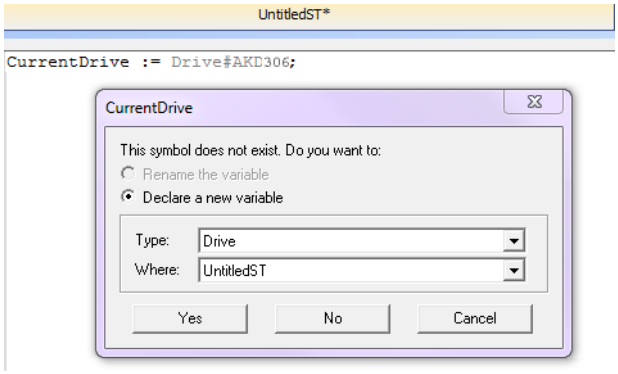


5.1.9.1.3 Declaring Enums

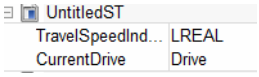
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Enums may be declared in the same way as variables; at the end of the line press Enter.

As CurrentDrive is not a part of the dictionary, a dialog is opened to add the variable. Set the Types as the enum name.



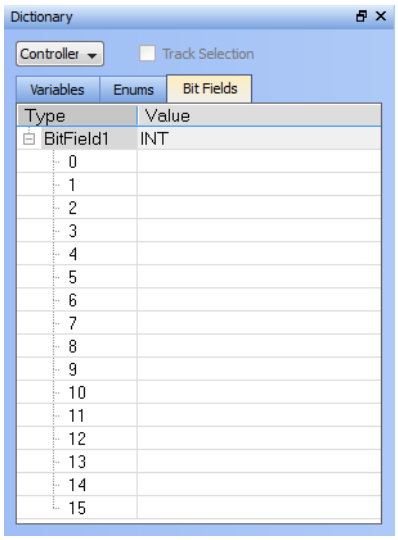
Once this is added, the variable will be displayed with the selected type in the Dictionary.



5.1.9.2 Bit Fields tab

A bitfield is a user-defined data type that combines multiple Boolean or integer values into a single variable.

- Bit Fields are used to define custom variable types and values.
- A bitfield packs multiple pieces of data together in one variable.
- Each field represents an independent piece of information.
  - These fields can be referenced individually within the application.
- Bitfields provide a compact and organized way to group related status and control information while maintaining easy access to individual fields.



5.1.9.2.1 Adding Bit Fields

A Bit Field is created in one of these ways:

- Right-click in the Bit Fields tab and select **Insert BitFields**.
- Press the **Insert** key while the Bit Fields tab is active
- Double-click in the Bit Fields tab.

This creates a default Bit Field labeled as `BitFieldn` with a Value type `INT`.

- The Bit Field type **Value** is changed by double-clicking on value, allowing a selection from a list.

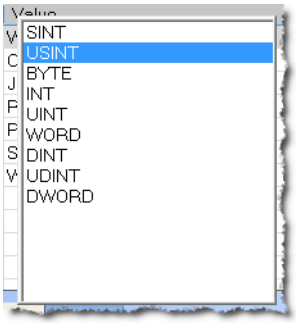


Figure 5-21:

Once the Type is defined you may populate the bits.

- Double click on a bit value to change the text.

IMPORTANT

To save the changes, you must press **Enter**, **Tab**, the Up, or the Down arrow key.

5.1.9.2.2 Using Bit Fields

This example sets up and tests a Bit Fields entry.

1. Define the Bit Field Type and Values.
   In this example, the bit is defined as **ConveyorState** with five values. (Figure 5-22)
2. To define a multibit field, select multiple value cells and type the name of the new field.

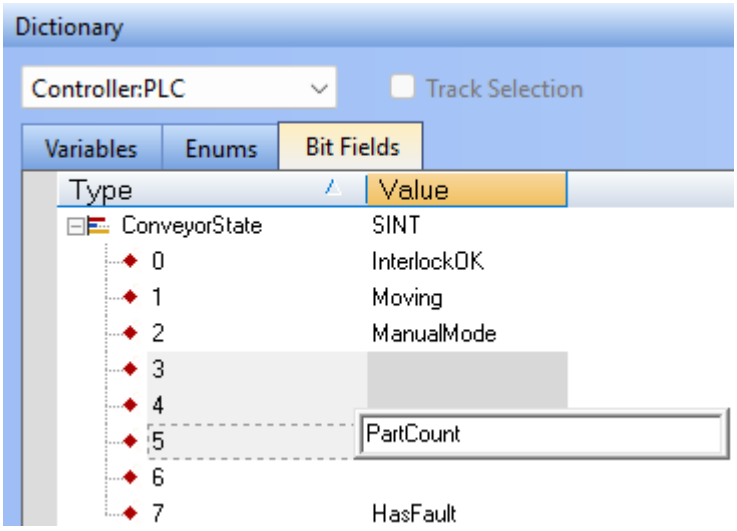


Figure 5-22: Example: ConveyorState

The name of the multibit field appears in all bits the field occupies. (Figure 5-23)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

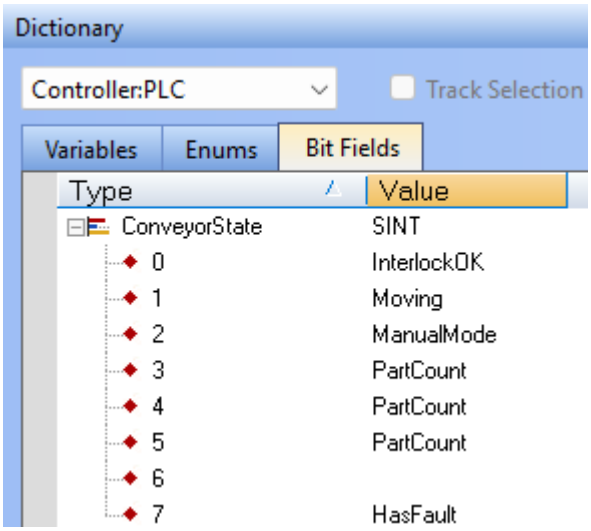


Figure 5-23: Multibit field

NOTE

- Not all bits need to be assigned to a field.
- There may be unassigned bits between fields.

3. Create a variable whose Type that matches the Bit Field.

   In this example, a variable is defined as **Converyor1State**.

4. Select the **ConveyorState** type from the drop-down menu. (Figure 5-24)

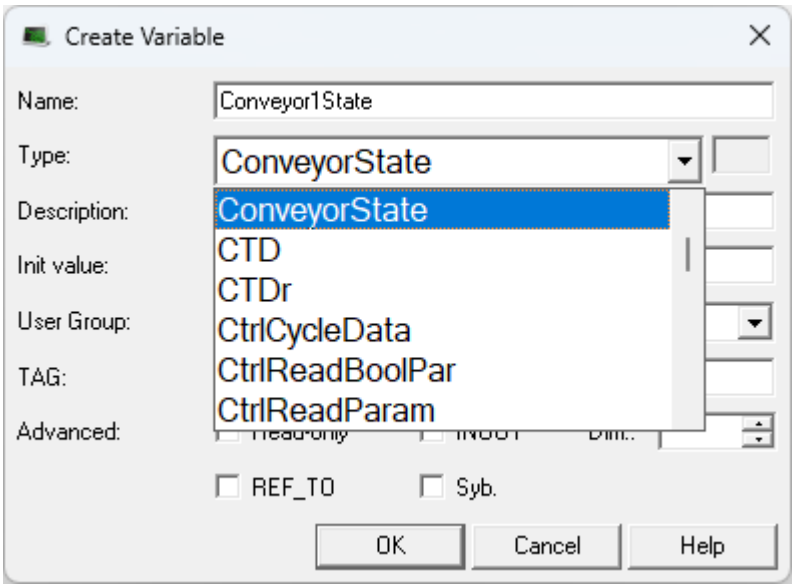


Figure 5-24: Selected ConveyorState

5. To use a field, enter the VariableName.FieldName in any editor.
   - In this example, the Structured Text editor is used.
   - **Conveyor1State.** is entered as the VariableName.
   - When the dot is typed, the user-defined fields dialog box opens. (Figure 5-25)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

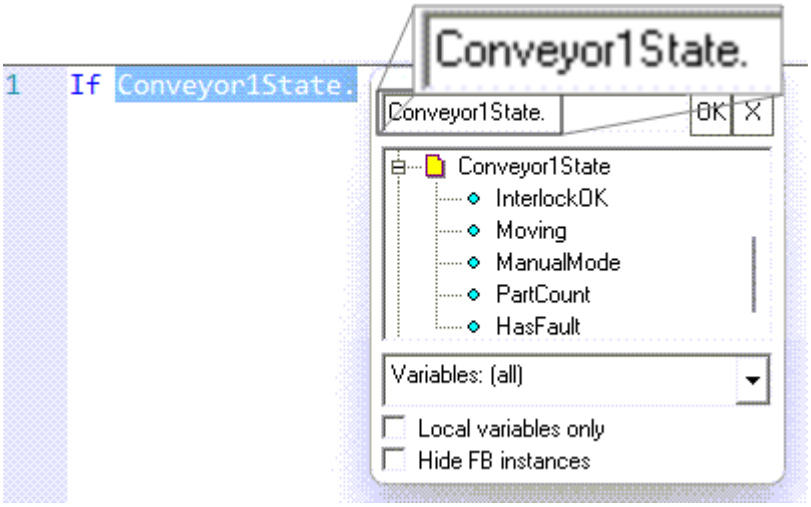


Figure 5-25: Conveyor1State user-defined field dialog box

6. In the dialog box, select the applicable field for the variable (e.g., PartCount).
7. Click **OK** to save the changes or selections and close the dialog box.
8. To assign or compare the value of a multibit field against a constant, specify the bitfield base type followed by # before the numerical constant.

   In this example, since ConveyorState has a base type of SINT, the **SINT#** prefix is used. (Figure 5-26)

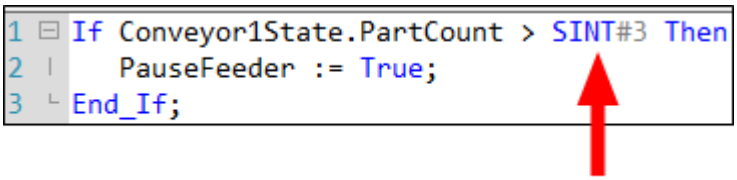


Figure 5-26: Multibit Field Comparison

9. To assign or compare the value of a field comprising of a single bit, use True, False, 0, or 1. Using the base type prefix generates a compiler error. (Figure 5-27)

```
Conveyor1State.ManualMode := True;     // OK
Conveyor1State.ManualMode := 1;        // OK

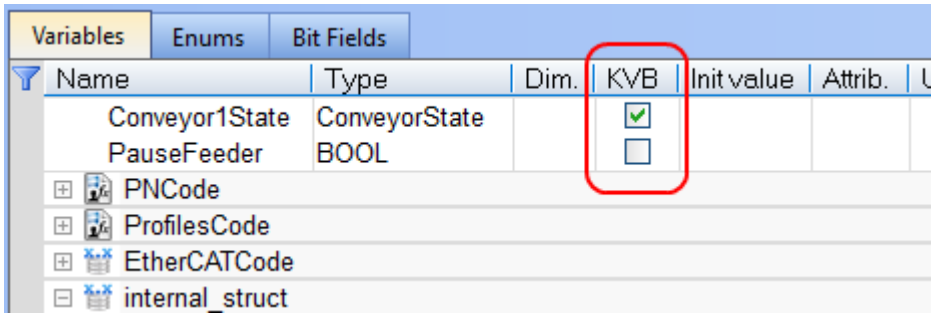
Conveyor1State.ManualMode := SINT#1;   // ERROR
```

Figure 5-27: Single-bit Assignment

5.1.9.2.3 Modbus Support

Bit Fields are supported over Modbus.

- Figure 5-28 is a variable with a custom Type defined in the Bit Fields list which is flagged for KVB.
- It will be exported to Modbus.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Figure 5-28: Custom Type

5.1.10 Information and Logs window

The Information and Log window is used to identify current state status and can be used to identify operational errors, compilation errors, and also to quickly assist you in finding areas of the workspace or program variables.

This window contains these tabs:

- Breakpoints tab with a list of breakpoints.
- Browse Variable tab.
- Compiler Output tab with state reports.
- Find and Replace tab for a system search.
- Log Messages tabs (Local or Controller).

5.1.10.1 Log Messages tabs

TIP

- Log messages provide important information when troubleshooting with KAS-IDE.
- When reporting an issue to Kollmorgen Support, copy/paste the logs in your report.

5.1.10.1.1 Log Messages - Local Log

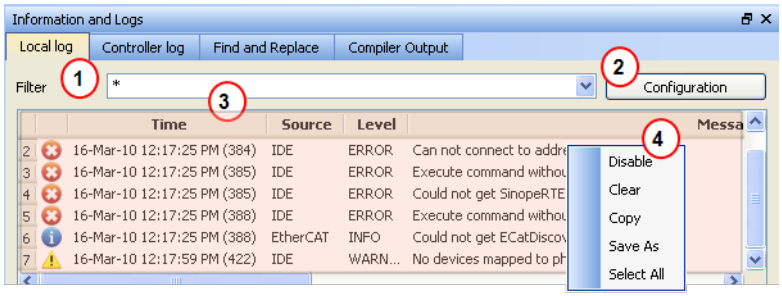


Figure 5-29: Log Messages - Local Log

Log messages are displayed on two tabs, **Local log** and **Controller log**.

- The **Local Log** tab shows all messages managed by the KAS-IDE to explain the current state of the system and to help identify any operation errors encountered when developing your system.
- The **Controller Log** shows all messages managed by the KAS Runtime.

Based on the configuration settings (see call out 2), only messages that are recorded and that match the filter 1 are displayed.

TIP

The Configuration button is only available on the Controller log tab when the KAS-IDE is connected to a controller.

5.1.10.1.2 Log Messages - List of Fields

Every log message in the table widget 3 has this information:

Field	Description
Time	Time when the log was recorded with the format: DD-MMMM-YY hh:mm:ss (millisecond)
Source	Identifies a software or hardware component issuing the messages. Each source is configured with a specific Level.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Field	Description
Level	Each message has one of the following levels with importance in ascending order: DEBUG > INFO > WARNING > ERROR > CRITICAL
Message	Text of the message issued from the source

5.1.10.1.3.1 Log Messages - List of Buttons

The table contains a contextual menu (see call out 4) with these commands:

Command	Description
Disable/ Enable	You can stop the log recording at any time, so that no more messages are added.
Clear	Empty the list by erasing all the messages already recorded.
Copy	Copy the text of the selected messages to the clipboard (you can perform multi-selection with the Ctrl or Shift keys).
Save As	Save all the messages in a log file.
Select All	Select all the messages that are displayed in the table.

5.1.10.2 Log Messages Settings

The KAS-IDE manages all messages according to these gates:

- "Configuration Settings" (→ p. 124) define what is recorded in the database.
- "Filtering" (→ p. 128) defines which messages are displayed in the table widget.

5.1.10.2.1 Configuration Settings

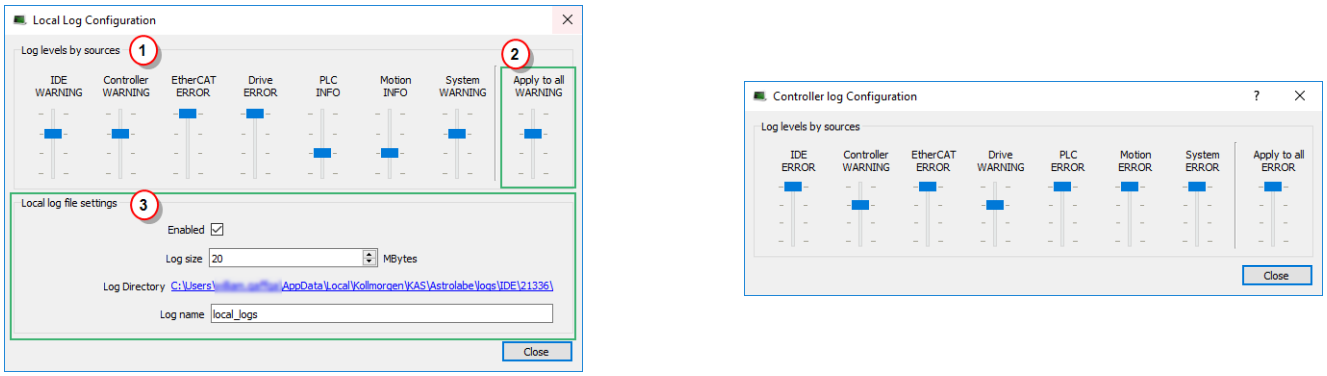


Figure 5-30: Configuration of the Local and Controller Log Messages

Call Out#	Description
1	Each log source can be set with its own notification level. • It is possible to get a maximum of log details for the selected source without getting a flood of irrelevant messages from the other sources by controlling the level of each source. • See: • "Message Sources" (→ p. 125) • "Log Message Levels" (→ p. 125)
2	You can set or reset all the sliders with the same level value

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Call Out#	Description
3	Logs can be recorded on the local machine as circular files. • On the controller, the recording of the logs is enabled by default. • See "Local Log File Settings" (➔ p. 126).

5.1.10.2.2 Message Sources

Click for Message Sources

Source	Apply to...
Controller	For the KAS Runtime items: • Drivers • IOEngine • SinopEngine
Drive	Messages from AKD drive
EtherCAT	For all kinds of EtherCAT items: • Motion bus • I/Os
IDE	Win32 Applications: • KAS-IDE • KAS Runtime Server (aka KAS Runtime Front-end)
Motion	Messages coming from the Motion engines: • PLCopen • Pipe Network • VM
PLC	For application engineers to create custom log in the PLC programs. See "Printf Function" (➔ p. 272).
System	For common API and libraries. Includes messages issued from the operating system.

5.1.10.2.3 Log Message Levels

Click for Log Message Levels

IMPORTANT

- Each message has one of these levels, with importance in ascending order:
  - DEBUG > INFO > WARNING > ERROR > CRITICAL

Level	Icon	Description
CRITICAL		Application crashes or becomes unstable. • Data is corrupted. • At that point, the application behavior can be unpredictable.
ERROR		The application does not behave as expected but the processes remain stable.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Level	Icon	Description
WARNING		System is stable but the KAS-IDE warns that an unexpected event can occur. • This is the default logging level. • You can ignore this log.
INFO		Information status of the current process. You can ignore this log.
DEBUG		Any information logged for development purpose. You can ignore this log.

5.1.10.2.4.1 Choose the Appropriate Level for a Log

When a level is set for a source, only messages with the same or higher importance are recorded.

- Drag the level control slider
  - **Up to reduce** the verbosity.
  - **Down to increase** the verbosity.
- When the configuration leads to lower verbosity, the treatment during the filtering is quicker.
  - Example: If a source is set to WARNING, then all messages with levels WARNING, ERROR and CRITICAL are recorded (DEBUG and INFO messages are discarded).
    DEBUG is the most verbose, whereas ERROR is the less verbose.

NOTE

Critical messages are always recorded; as a consequence, the Critical level is not visible on the slider.

5.1.10.2.5 Local Log File Settings

Click for Local Log File Settings

Log files are a group of small files where all the last logs are recorded.

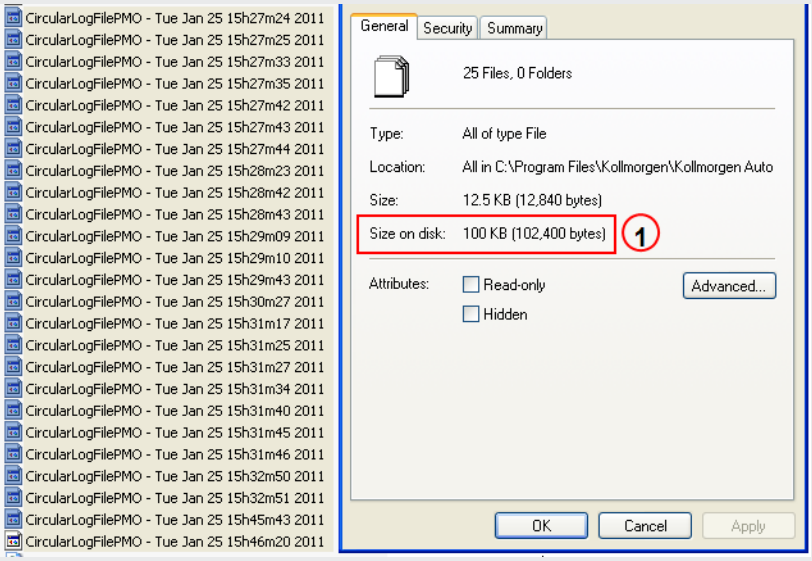
Each log is recorded as a separated line.

TIP

Import the log files into Microsoft Excel using drag-and-drop.

Field	Description
Enabled	The Log File Settings has to be enabled to record all the logs.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Field	Description
Log size	To prevent any overflow, you can define the maximum size on disk dedicated to the group of local log files. • The minimum size is 1MB, the maximum log size is 2MB. • At most, the system stores 20 files. • When a new file is created, an old one is deleted, maintaining the 20 file threshold. • In Figure 5-31, the size on disk is limited to 100 KBytes (see 1).  **Figure 5-31:** Size on Disk The number of rows can vary for each file, depending on what is in the backlog when KAS creates the log files.
Log Directory	The Log Files are stored in this directory. • Each KAS-IDE instance has a unique log directory, defined by the process identification number. • You can access the local log files for this specific instance of the KAS-IDE, by clicking this link to open the directory location in the File Explorer. • See "Log File Storage" (➔ p. 127).
Log name	You can define the file name prefix to be used on the local machine (on the controller, the file name prefix is: controller logs). The suffix to create the complete filename contains a timestamp with this format: - day MMM DD HHhMMmSS YYYY nn. **NOTE** • The Log name change is applied immediately. • The name used by the next IDE instance is determined by the last IDE instance.

5.1.10.2.6 Log File Storage

Click for Log File Storage

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- For the local machine (IDE), the Log files are located in the following location:

  `C:\Users\(user name)\AppData\Local\Kollmorgen\KAS\Astrolabe\logs\IDE\(process ID)`

- For the controller, the Log files are located under:

  `C:\Users\(user name)\AppData\Local\Kollmorgen\KAS\Sinope Simulator\Application\logs`

NOTE

- It is recommended that you use either the KAS-IDE or webpage method, but not both.
- The communication is unidirectional and the configuration is not read at runtime.

5.1.10.2.7 Filtering

Use a filter to narrow the list of recorded messages.

The filter is applied on all the strings displayed on each row of the table widget (i.e., Time, Source, Level and Message).

The drop-down menu gives access to some predefined filters, which can also be edited.

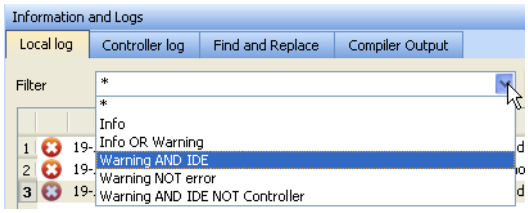


Figure 5-32: Filtering the Messages

Example: Filtering with **Warning NOT error** means that only the lines including the word "warning" but not the word "error" are listed.

5.1.10.2.8.1 Filtering Rules

These are the filter rules:

- You can combine several strings by including one of these Boolean operands:
  - OR
  - AND
  - NOT (or use the exclamation mark "**!**")
- Several keywords separated with spaces are considered as an exact string.
- Filtering is **not** case sensitive.

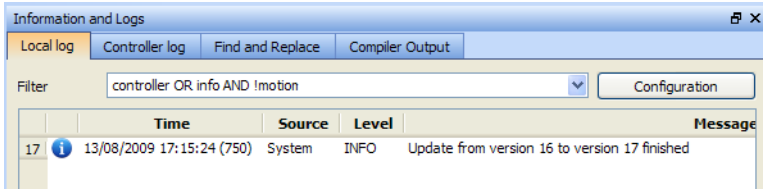


Figure 5-33: Example: Filtering the Messages

⚠WARNING

- When you apply the filter, all the currently recorded messages are parsed and displayed if they match the filter.
- But all the upcoming recorded messages are added as new rows at the end of the table widget with **no filtering**.

5.1.10.2.9.2 Log Scrolling

If you select a message in the table, the scrolling is **stopped**.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

All the upcoming recorded messages are added at the end of the list, but the selected message always remains in the same place (you have to scroll down to make the most recent messages visible).

If you select the last row of the table (shortcut: **Alt+Page Down**), the scrolling is **active**.

The last recorded message is always selected and visible at the bottom of the table.

5.1.10.3 Find and Replace tab

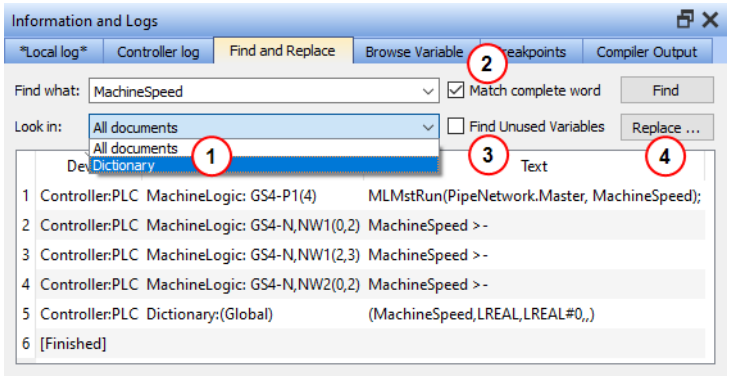


Figure 5-34: Find and Replace tab

This tab is used to search for all the instances of a string of characters (search is **not** case sensitive) in the entire environment and replace it.

You can re-use one of the last ten entries or type a new text string.

Call Out#	Description
1	The operation can be performed across: • All documents: Search all files of the project and the dictionary. • Dictionary: Search the dictionary.
2	The search string can be with a **complete word** only. • When this option is selected, the search is case sensitive. • When selected, only the instances that match the complete words defined in the **Find what** field are selected. • Example: A search for "MyVar" returns "MyVar" but not "MyVariable".

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Call Out#	Description
3	To optimize a project, the **Find Unused Variables** option is used to perform a search in the dictionary to locate variables not used in any program. • A variable is **not used** when there is no effective usage of it in your entire project. • It can still be the case even when a value is assigned to a variable (e.g., MyVar := 100. * Axis1.Velocity;). • The variable MyVar becomes **used** when it is affected as an input argument (e.g., Velocity := MyVar ;). • Select this option and click **Find** to generate a list of the unused variables. • You can select one or more variables (by Shift- or Ctrl-clicking) and right-click to open the context menu. • The menu includes a **Delete** option (Figure 5-35). • Double-click a variable in the list within the table widget to open it in the Dictionary.

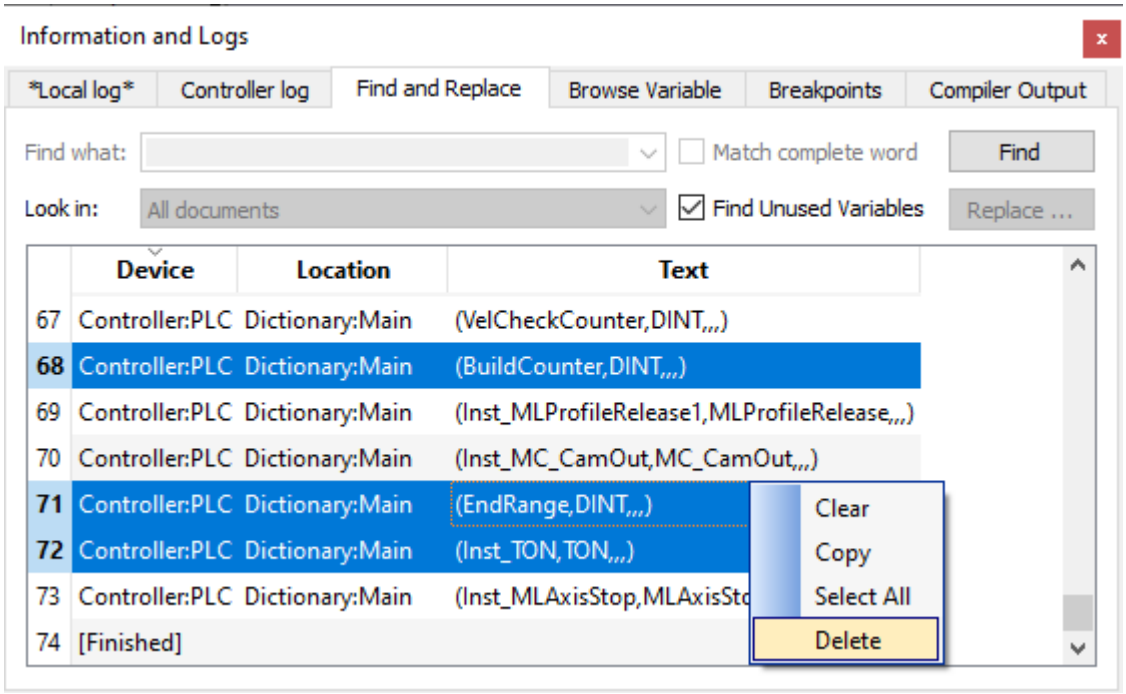


Figure 5-35: Find and Replace tab with Delete menu option Selected

4	Use the **Replace...** button to replace in all documents a string with another one (or re-use one of the last ten entries). • The search can be refined by forcing the search to a **Whole word**, and may be case sensitive or **Case Insensitive** (Figure 5-36). • The replace function is limited. • It is not supported in the Dictionary, Pipe Network, and HMI. • For these, the change must be made manually.

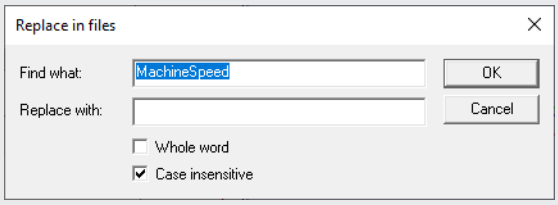


Figure 5-36: Replace in Files dialog box

Once the search is done, the results appear in the table widget at the bottom of this tab.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

If a replace has been performed, the Text column provides more information about the replacement.

Double-click the applicable item in the list to open it in its relevant location (it can be a PLC editor or the Dictionary).

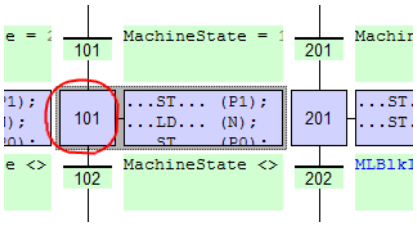
5.1.10.3.1 Location Details

Understand the Location Details – SFC Programs

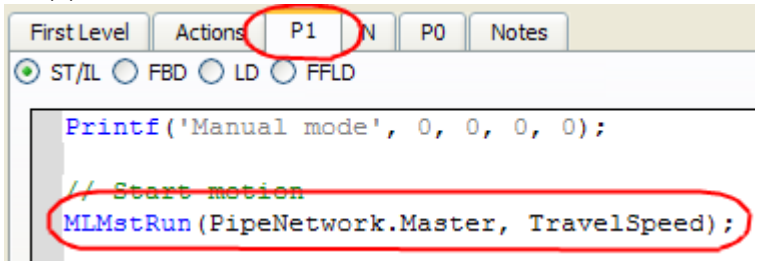
SFC Location details



- Controller : PLC and MachineLogic refer to the program in the Project Explorer
- GS stands for Graphical and Step (T is for Transition)
- 101 is the reference in the editor



- -P1(4) refers to the P1 tab and the 4th line in the source code:

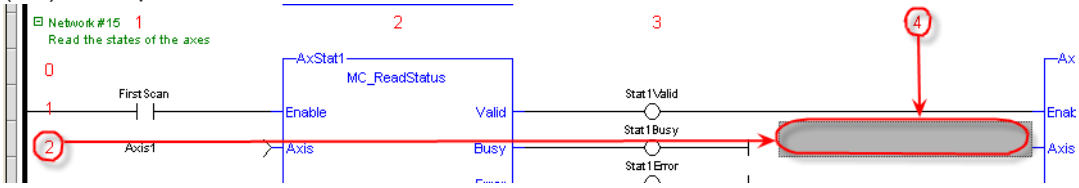


How to Understand the Location Details – FFLD Programs

FFLD Location details

```
Controller:PLC:>> Complex variables stored in a separate segment
Controller:PLC:Loading application symbols...
Controller:PLC:Main
Controller:PLC:Main: G55-N,NW15(4,2): Object not connected on the left
Controller:PLC:Main: G55-N,NW15(4,0): Bad value on box input(s)
Controller:PLC:Error(s) detected
Controller: ------------ PLC failed ----------
```

- **Controller**: PLC and Main refer to the program in the Project Explorer.
  - GS stands for Graphical and Step (T is for Transition)
  - 5 is the reference in the editor.
  - -N refers to the N tab.
  - NW15 stands for Network number **15**.
  - (4,2) correspond to the **X,Y** coordinates of the cell relative to the current network:



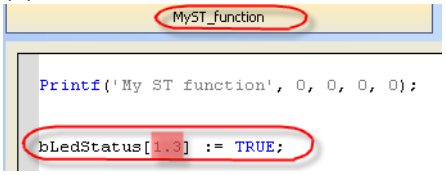
Understand the Location Details – ST Programs

ST Location details

```
Controller:PLC:MyST_function
Controller:PLC:MyST_function: (5): Array index expected - must be a DINT expression
Controller:PLC:Error(s) detected
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- **Controller**: PLC and MyST_function refer to the program in the Project Explorer.
  - (5) refers to the 5th line in the source code.



See "Find and Replace Operations" (→ p. 132).

5.1.10.4 Find and Replace Operations

The Find and Replace command enables you to search for a specified string of characters within your project.

You can use any of the following methods to access this functionality:

- From the **Information and Logs** toolbox
- In the **Dictionary** panel
- From an **editor** (FBD, FFLD, IL, ST)

5.1.10.4.1 Information and Logs

See "Information and Logs window" (→ p. 123) for more information.

5.1.10.4.2 Dictionary

Right-click on the variable name and select the **Find all** command in the menu.

This command starts a search of all documents for the selected variable and displays the results in the table widget within the "Information and Logs window" (→ p. 123) toolbox.

NOTE

This opeartion selects only the instances that match the complete words (e.g., a search for "MyVar" returns "MyVar" but not "MyVar1").

5.1.10.4.3 Editor

It is possible to perform a search and replace from a PLC editor (FBD, FFLD, IL, ST) by selecting the **Find**, **Find next**, and **Find All** commands in the contextual menu.

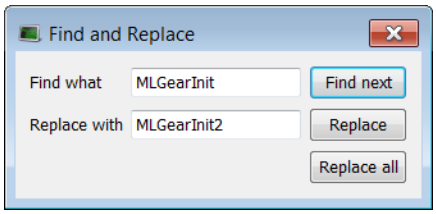


Figure 5-37: Find and Replace from an Editor

Function	Description
Match Whole Word	When selected, only the instances that match the complete words defined in the **Find what** field are selected. Example: A search for `MyVar` returns **MyVar** but not **MyVar1**).
Find next	Used to select in the current editor the next instance of the matched string.
Replace next	Used to replace the next instance of the matched string.
Replace all	Used to replace in the current editor all instances of the matched string.

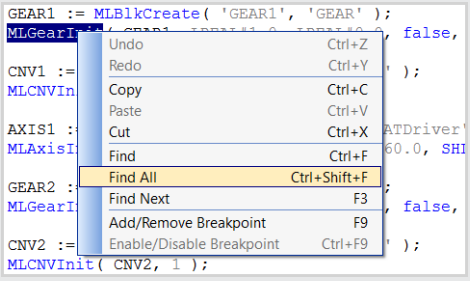
NOTE

The **Find**, **Replace** and **Replace all** operations work only for **variable symbol** property of the **Control**.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TIP

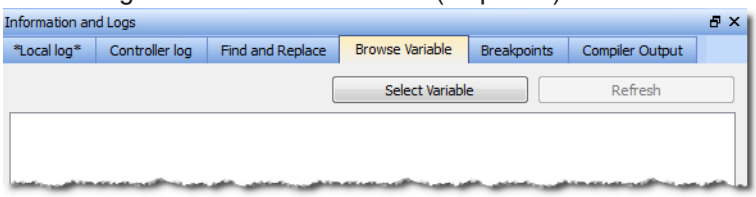
When the Find All command is applied to a variable, the system returns a list of all instances of the variable, not just those in the current program.



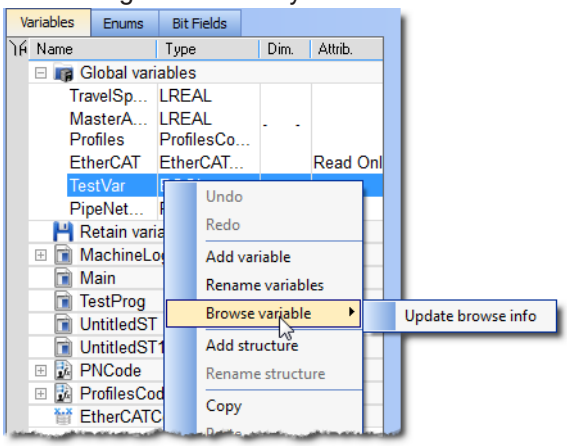
5.1.10.5 Browse Variable tab

This tab is used to browse all instances of a variable. It will show the locations and usage of each instance of a single variable. There are two ways to populate the tab:

- Click the **Select Variable** button.
  See "Using the Browse Variable tab" (➔ p. 133).

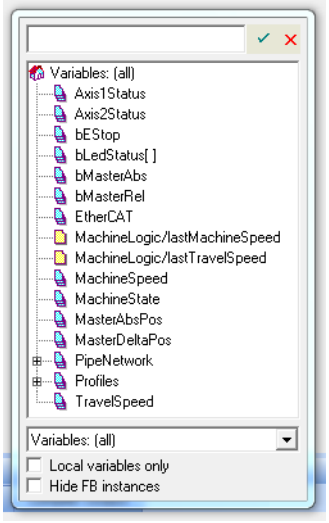


- Right-click on a variable in the Dictionary and select **Browse Variable**.
  See "Using the Dictionary's Browse Variable menu item" (➔ p. 134).



5.1.10.5.1 Using the Browse Variable tab

1. Click the **Select Variable** button to open a variable selection pop-up window.
2. Select a variable from the pop-up list.

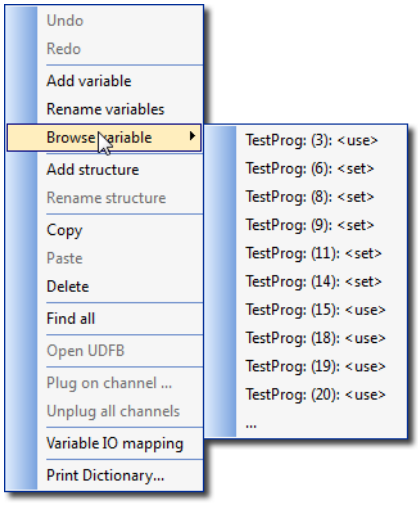


3. The Browse Variable tab is updated with a two-column table detailing the variable's usage
   - **Location** shows the file name and line number where the variable is used.
   - **Usage** shows whether the variable is being read (**use**) or written to (**set**).
4. Double-click on an entry to be taken to the correspondinglocation in the editor.

NOTE

- The process of generating the Browse Variable content after clicking Select Variable can take a significant amount of time, depending upon the number of files in the project. If there are no modifications to the project, further browse operations will not take any time.
- The browse information may not be current if changes are made to the project.
- The browser information needs to be refreshed after saving and compiling a project. A message is shown in the tab to alert you if the information may not be accurate.

5.1.10.5.2 Using the Dictionary's Browse Variable menu item



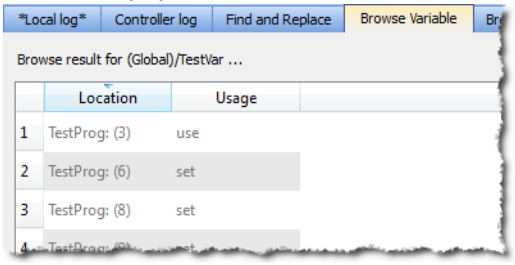
The first time this menu is accessed (per project instance) you are presented with **Update browse info**.

- This updates the browse data.
- Further selections of any variable result in the location and usage being shown directly in the menu.
- Selecting an entry will take you to the corresponding location in the editor.

When the menu has more than ten entries an ellipsis (...) is added to the bottom of the menu.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Selecting the ellipsis (…) opens the Browse Variable tab.
- This tab is populated with the variable locations and usages.



5.1.10.6 Breakpoints tab

The Breakpoints tab lists all of the breakpoints in the PLC program, including their position and status.

- Double-click an entry to move to that location in the editor.
- Breakpoints may be enabled and disabled singly by clicking the **Status** column check boxes.
- Buttons on the left of the tab are used to remove single breakpoints, enable and disable all breakpoints, and remove all breakpoints.

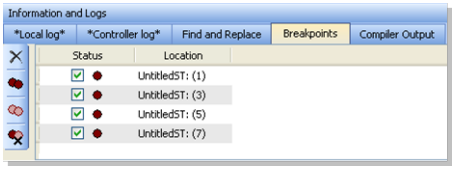


Figure 5-38: Breakpoints tab

Button	Action Performed
	Delete selected breakpoint from the list and editor.
	Enable all breakpoints. This makes all breakpoints active.
	Disable all breakpoints. This makes all breakpoints inactive.
	Remove all breakpoints.

Right-clicking a breakpoint entry in the list provides for enabling, disabling, deleting the entry, and going to that location in the source code.

Breakpoints (both active and inactive) which have been reached in the code are flagged with a yellow triangle.

This provides a quick and easy way to identify the breakpoint.

This can be seen in both the code and the Breakpoints tab.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

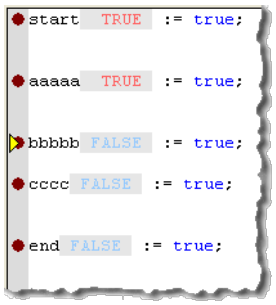


Figure 5-39: Breakpoints code

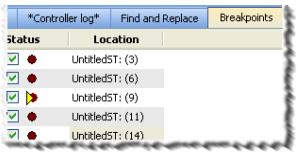


Figure 5-40: Breakpoints tab

TIP

Any program (except for an SFC program) that contains a breakpoint that gets "hit" during debugging will be automatically opened for your convenience.

Breakpoints set in SFC programs cannot be enabled or disabled.

Entries in the Breakpoints tab do not have a check box to perform these actions.

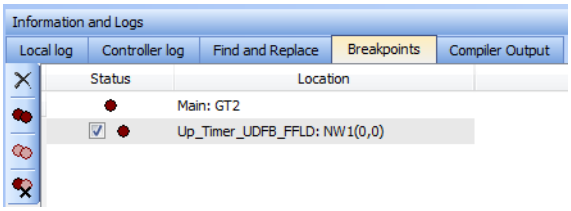


Figure 5-41: Example of a breakpoint (Main: GT2) set in an SFC program.

See Also

- "Breakpoints" (➔ p. 270)
- "Set, Remove, Enable, and Disable Breakpoints" (➔ p. 271)

5.1.10.7 Compiler Output tab

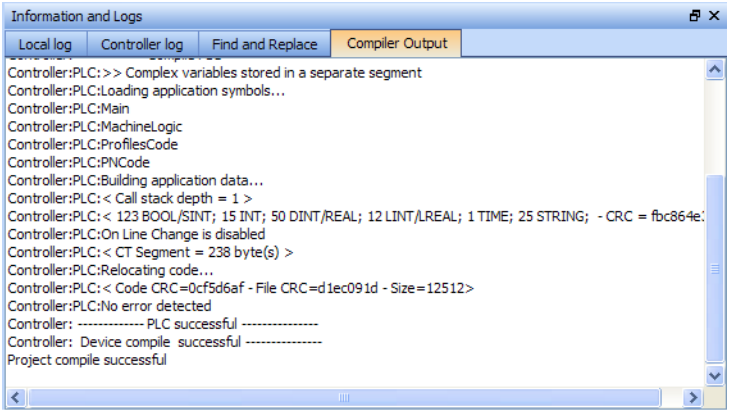


Figure 5-42: Compiler Output tab

This tab shows:

- Information about the last project compilation.
- Information messages as well as Warnings and Errors (highlighted in **red**).
- Successful and unsuccessful output is reported to help identify and troubleshoot problem areas of the program development.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

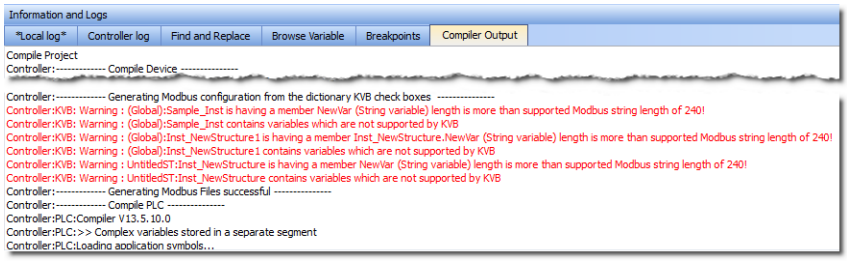


Figure 5-43: Compiler Output tab with Warnings

TIP

Double-click an error to open the program in the workspace and jump directly to the relevant location in the editor.
This lets you rework the program and fix the error.
In Figure 5-43, the first warning is for Sample_Inst.
Double-click the item opens that item's location in the Dictionary. (Figure 5-44)

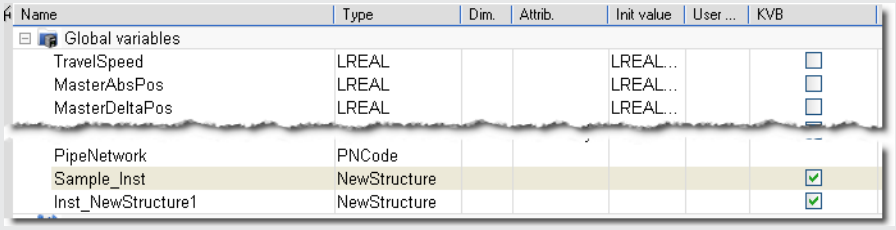


Figure 5-44: Item open in Dictionary

When there is a long list of statements, only the bottom part is displayed. Do not forget to scroll up.

How to Clean-up the Code

1. Scroll up to start from top and locate the first error message.
2. Fix the error.

   TIP

   Because fixing **one** piece of code can eliminate **multiple** compiler output error, it is recommended to recompile each time you correct an error.

When no more errors exist, these messages are shown:

- PLC successful (the IEC 61131-3 code is correct).
- Device compile successful (is related to the Motion part (e.g. CAM profiles), EtherCAT XML file...).
- Generating Modbus files (related to the variables mapped with the HMI).
- Project compile successful (the complete project is ready to be downloaded to the target).

Text Displayed

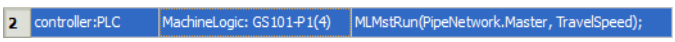
Operands of * or / must be numbers and have the same type.

Meaning

- This error appears in a ST instruction when a constant expression does not have the expected type in a multiplication or division operation.
- Typically, REAL is the default precision for floating points, so you must explicitly declare the long real constants with the LREAL# prefix when required.
- See "Constant Expressions" (➔ p. 301).

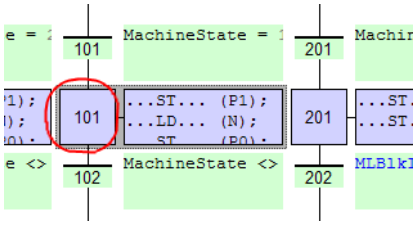
Understand the Location Details – SFC Programs

SFC Location details

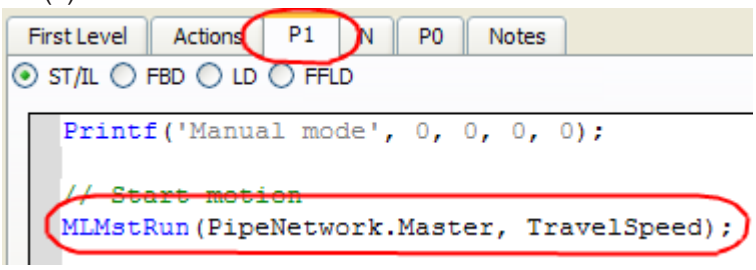


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Controller : PLC and MachineLogic refer to the program in the Project Explorer
- GS stands for Graphical and Step (T is for Transition)
- 101 is the reference in the editor



- -P1(4) refers to the P1 tab and the 4th line in the source code:

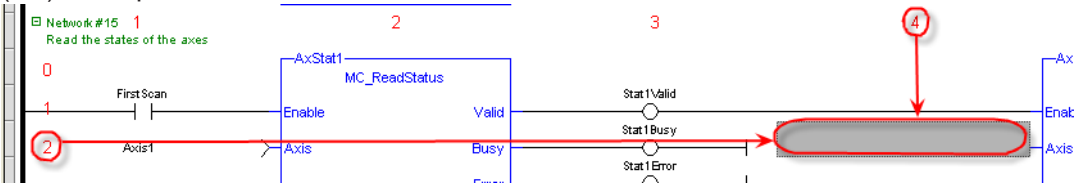


How to Understand the Location Details – FFLD Programs

FFLD Location details

```
Controller:PLC:>> Complex variables stored in a separate segment
Controller:PLC:Loading application symbols...
Controller:PLC:Main
Controller:PLC:Main: G55-N,NW15(4,2): Object not connected on the left
Controller:PLC:Main: G55-N,NW15(4,0): Bad value on box input(s)
Controller:PLC:Error(s) detected
Controller: ---------- PLC failed ----------
```

- **Controller**: PLC and Main refer to the program in the Project Explorer.
  - GS stands for Graphical and Step (T is for Transition)
  - 5 is the reference in the editor.
  - -N refers to the N tab.
  - NW15 stands for Network number **15**.
  - (4,2) correspond to the **X,Y** coordinates of the cell relative to the current network:

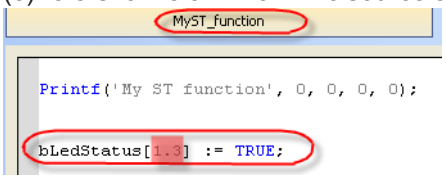


Understand the Location Details – ST Programs

ST Location details

```
Controller:PLC:MyST_function
Controller:PLC:MyST_function: (5): Array index expected - must be a DINT expression
Controller:PLC:Error(s) detected
```

- **Controller**: PLC and MyST_function refer to the program in the Project Explorer.
  - (5) refers to the 5th line in the source code.



5.1.11 Watch Window

This toolbox is used to add variables to a dedicated Watch Window to view its value in real time.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

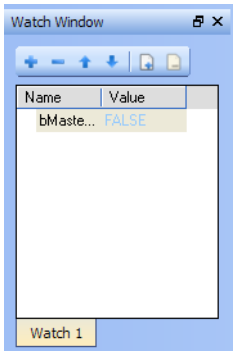


Figure 5-45: Watch Window

NOTE

- PLC variables viewed in the Watch Window are saved as a part of the project.
  - This means that the next time you open the project, those variables are pre-loaded in the Watch Window.
  - This does not apply to AKD variables, which are not saved with the project.
- While variables may be changed, locked, or unlocked, parameters are not.
  - See "Force a Variable" (→ p. 279).
  - AKD parameters in the Watch Window are read only.

5.1.11.1 Multiple Watch Windows

Use the KAS-IDE to group several variables in a single Watch Window.

- A maximum of 10 different Watch Windows are allowed.
- Each window is shown as a tab with its own label.

5.1.11.1.1 Watch Window toolbar

Button	Description
	Add a variable with the PLC Variable Selector.
	Remove a variable.
	Move up the selected variable.
	Move down the selected variable.
	Add a new Watch Window. • Each window is shown as a tab with its own label. • A maximum of 10 tabs is allowed.
	Remove the selected watch window.

TIP

Use **drag-and-drop** from the Dictionary or the PLC editors to directly add an existing variable in the Watch Window.

5.1.11.1.2 Variable Information

Each variable in the table widget has this information:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Field	Description
Name	• Lists the variables as well as structure, arrays, and expressions. • Double-click a variable (or press the **F2** key when the variable is selected) to edit its name (except for structure and array members).
Value	• When the application is running, this is the variable or expression's value. • Double-click a value to force modification of the selected variable.

The contextual menu allows you to:

- Add a variable.
- Remove a variable.
- Remove all variables.

5.1.11.2 Access Structure and Arrays

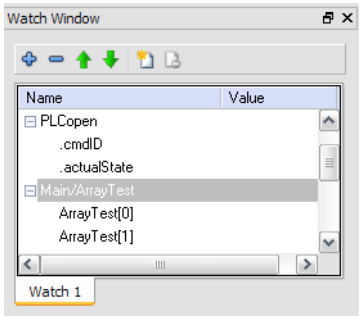


Figure 5-46: Watch Window - Accessing Arrays

When a structure or an array is in the watch window, you can expand its node to display all its members.

IMPORTANT

A structure or array members cannot be deleted, edited, or moved up/down in the list.

5.1.11.3 Add a Variable

1. Double-click the nodes (**(Global)**, **Main**, etc.) to expand their related variables.
   TIP
   Expand AKD node to add AKD parameters to the Watch Window.
2. Select one from the list.
3. Click **OK**.

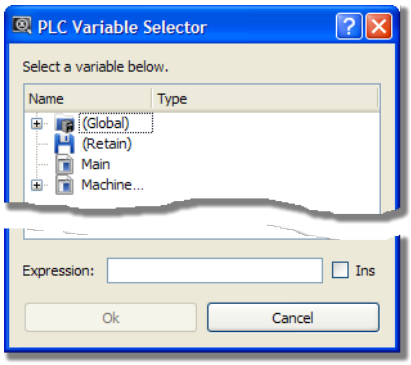


Figure 5-47: Watch Window - Selecting PLC Variable

This variable is added to the current "Watch Window" (→ p. 138).

5.1.11.4 Add an Expression

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

You can enter variable strings as an expression.

Example: Use this procedure to add together two integer variables:

- Click the Add symbol to open the PLC Variable Selector.
- Choose a variable, but do not click OK yet.
  (the variable is added to the expression field where you can do any required editing)
- Select the **Ins** option.
  (this option allows you to insert the next selected variable at the current cursor position in the expression edit field)

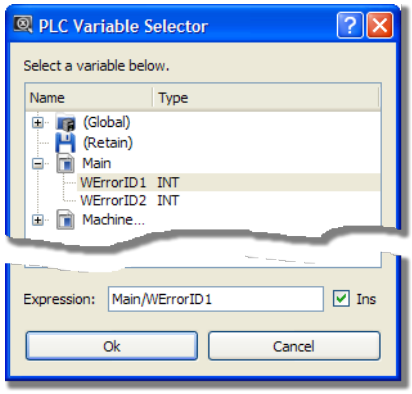
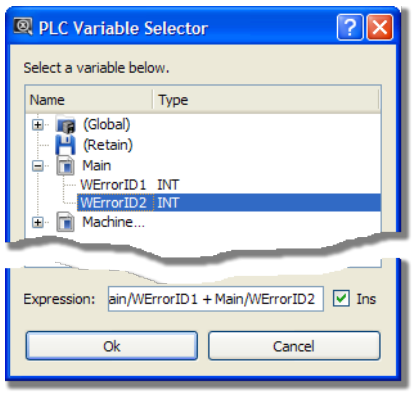


Figure 5-48: Watch Window - Creating Expression

- Press the PLUS SIGN (**+**) in the expression field.
- Select another variable.
- Click the **OK** button.



- Then the expression is displayed into the watch window

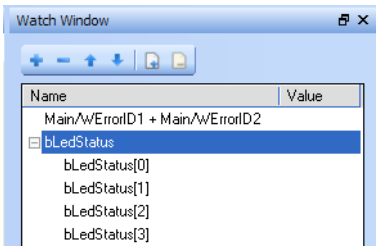


Figure 5-49: Watch Window - Displaying Expression

What you can include in a complex expression:

- Index of array
- Comparison “>”, “<”, “<>”, “=”
- Operator “+”, “*”, “-“, “/”

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

The DIVIDE SIGN (/) is not interpreted as an operator when used with prefixed variables (e.g., MachineLogic/lastTravelSpeed)

5.1.11.5 Force a Variable

At runtime, all variables in the table widget are animated [1] with real-time values.

At runtime, double-click on the value of the variable in the list or press the **ENTER** key when it is selected.

A popup window appears and allows you to "Force" (→ p. 142), "Lock" (→ p. 142), or "Unlock" (→ p. 143) a variable.

5.1.11.5.1 Force

- Change the value of the selected variable.
- Depending on the variable type, you have the possibility to define its value either in the text field or with the check boxes. (Figure 5-50)

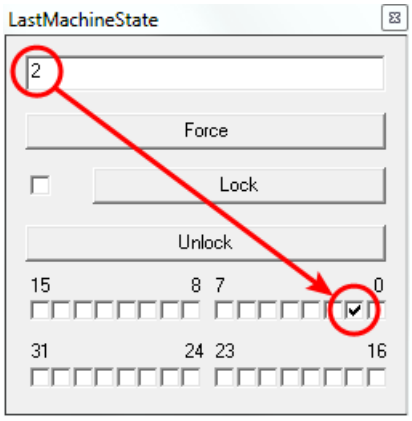


Figure 5-50: Forcing a variable

5.1.11.5.2 Lock

- Variables, member variables of a structure and variables in an array may be locked.
- When a variable is locked, its value is no longer changed by the runtime.
  - You can then force its value from the debugger independently from the runtime operations.
- All variables can be locked and forced at runtime.

NOTE

The value of a locked variable is displayed with square brackets. (Figure 5-51)

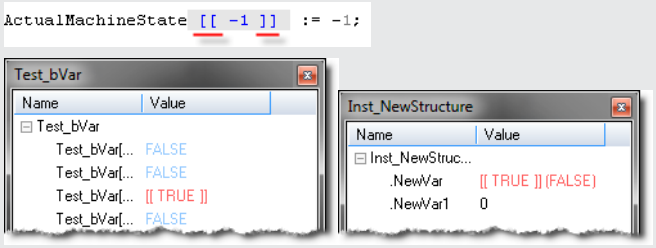


Figure 5-51: Example: Lock variable

The variable locking feature can be enabled or disabled via the PLC Options device toolbar button.

If enabled, the Lock and Unlock buttons are accessible (Figure 5-53):

[1] To better track variables and expressions of the PLC programs in test mode, the KAS-IDE dynamically computes their value along with the execution and displays the result

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

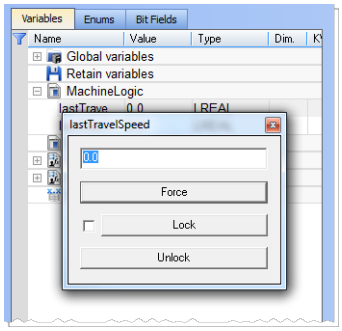


Figure 5-52: Enabled locking

If disabled (default), the Lock and Unlock buttons are not accessible (Figure 5-52):

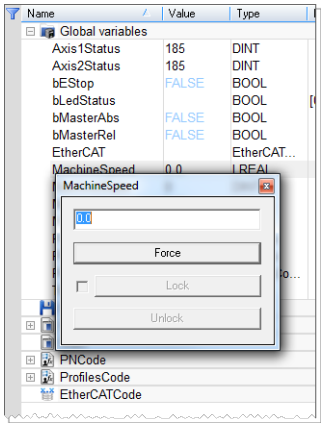


Figure 5-53: Disabled locking

IMPORTANT

If PLC variable locking is enabled, the controller Runtime requires an additional 3% to 10% CPU processing power to manage the PLC variable locking.
For the best controller performance, disable PLC variable locking.

5.1.11.5.3 Unlock

- Remove the lock on a variable so that it can be changed again by the runtime.

5.1.12 Kollmorgen AKD Series Servo Drive

IMPORTANT

First generation AKDs and Kollmorgen Essentials Drive support one axis per drive.
AKD2G drives support two axes per drive.
When configuring drives in the "EtherCAT Devices tab" (➔ p. 536), you must designate which motion axis applies to each drive axis, which is the actual axis the drive controls.

In addition to different views, WorkBench provides a "WorkBench Toolbar" (➔ p. 143) and "WorkBench Status Bar" (➔ p. 144) to display some extra information.

5.1.12.1 WorkBench Toolbar

The toolbar provides access to these actions:

- Enable / Disable the drive (software enable).
- Start / Stop the Service Motion.
- Mode: Position / torque / velocity

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Disable and Clear Faults: Click this button to clear the fault, then click the Enable button to enable the drive again.

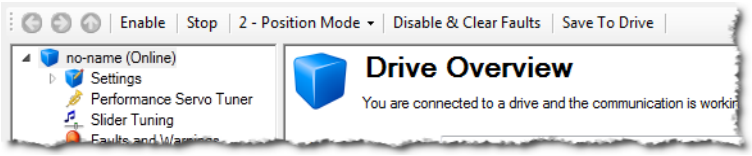


Figure 5-54: AKD Toolbar

TIP

To stop all the Kollmorgen AKD Series Servo Drives views at the same time, click the Stop button in the "Device toolbar" (→ p. 470).

5.1.12.2 WorkBench Status Bar

The status bar provides the following information on the drive:

- A fault indicator (No Faults / x Faults) that becomes red when any AKD gets a fault.
  - Use the "Log Messages Settings" (→ p. 124) to get more details on the drive messages.
- The drive status: active / inactive.
- The software (SW) enable status.
- The hardware (HW) enable status.



Figure 5-55: AKD Status Bar

For the SW and HW enable status indicators, the color code is:

- Green when it is OK (i.e., everything is ready to do motion).
- Red in case of errors / faults.
- Grey for all other cases (e.g., when SW or HW is not enabled, the status is not green because a motion could not happen, and not red because it is not an error).

5.1.13 Status Bar

A status bar located at the bottom of the KAS-IDE main window displays the labels from left to right:

- "Local Version" (→ p. 144)
- "Controller Version" (→ p. 145)
  - Application version located in the controller.
- "Drives State" (→ p. 145)
- "Controller State" (→ p. 145)
  - Stopped/running.
- "Connection State" (→ p. 145)
- "Color Codes" (→ p. 146)



Figure 5-56: Status Bar Labels

- The icon ( ) between the Local and Controller versions is used to show any differences.
  - See "Compare PLC Programs" (→ p. 277).
- The space on the left of the status bar is reserved for messages.

5.1.13.1 Local Version

This label provides information about the local version present in the KAS-IDE.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

These are the different states:

- Nothing displayed.
  - For instance, when no project is loaded.
- Version information.
  - When available.
- Compilation error.
  - Background is **red**.

NOTE

- The **Local Version** section in the Status bar shows **Compiling...** while the project is being compiled.
- If the compilation is successful, the Project version is appears.
- If the compilation fails, a **Compilation Failed** message appears with a **red** background.

TIP

Position the mouse over the text field to display a tooltip with the detailed version information.

5.1.13.2 Controller Version

This label provides information about the version present in the controller.

These are the different states:

- Nothing displayed.
  - When not connected.
- No Application in the controller.
- Version information.
  - When available.
  - Syntax of the version label is: <project_name>:<version>

TIP

When an application is active in the target, you can hold the mouse over the text field to display a tooltip with the detailed version information, including a timestamp of the compilation.

5.1.13.3 Drives State

These are the different states:

- Drives inactive.
  - Drives are disabled or the application is not connected to the target.
- Drives active.
  - At least one drive is active.
- Drives error.
  - At least one drive is in error.
- Drives unknown.
  - Drive status is unknown.
  - Application is not running or OnLineConfigMode is not active.

5.1.13.4 Controller State

This label lets you know if the controller is running or stopped.

These are the different states:

- Nothing displayed.
  - The label is empty when the KAS-IDE is not connected to the target.
- Controller is stopped.
- Controller is running.
- Controller is in Online Config mode.

5.1.13.5 Connection State

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The Connection label displays the Connection state between the KAS-IDE and the controller.

These are the different states:

- Not connected.
- Connecting.
- Connected.
  - Background in **green**.
- Connection Error.
  - Background in **red**.
- Unexpected Disconnection.
  - Background in **red**.

TIP

Hold the mouse over the text field to see a tooltip with detailed information about the Error and the Controller address when connected.

5.1.13.6 Color Codes

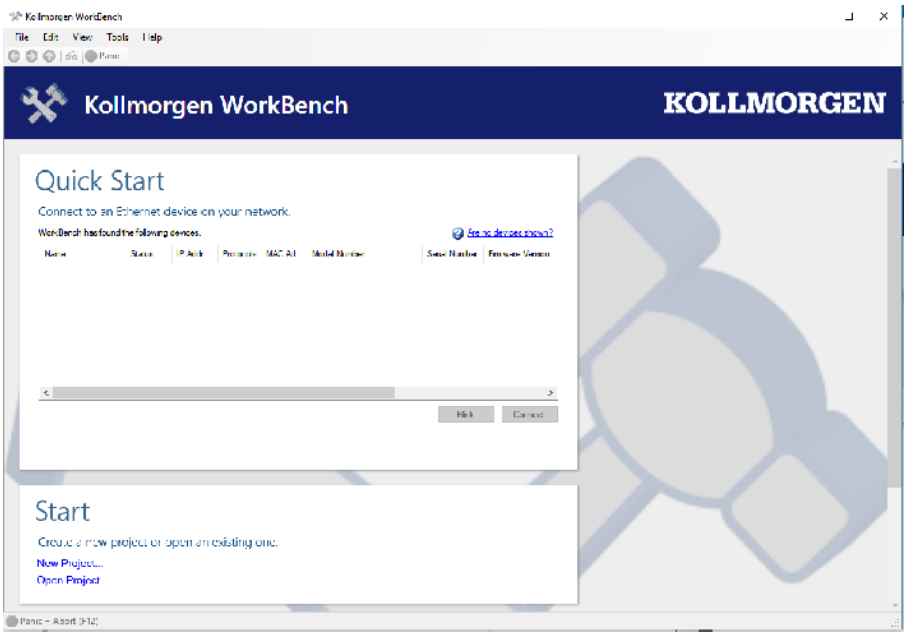
- The Local and Controller version labels has an **orange** background if there is a version mismatch between the KAS-IDE and the controller.
  - This warns you that you have to download the new version of the application.
- The Local version label has a **red** background if the compilation fails.

Connection State	Local Version	Controller Version	Controller Status	Connection Status
Comm. error	Version A			Connection Error
Connected	Version A	No Application	Stopped	Connected
Connected	Version B	Version B	Stopped or running	Connected
Connected	Version B	Version A	Stopped or running	Connected
Connected	Compile error	Version A	Stopped or running	Connected
Connecting	Version A			Connecting...
Disconnected				Not Connected
Disconnected	Version A			Not Connected
Disconnected	Compile error			Not Connected
Disconnected	Version A			Unexpected disconnection

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6 Getting Started



This section provides explanations and procedures to accomplish common tasks with the KAS-IDE.

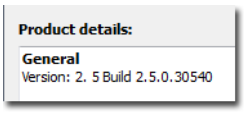
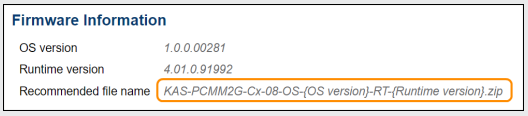
6.1 KAS-IDE Runtime Compatibility

The KAS software includes two main software components, the KAS-IDE and the Runtime.

- The KAS-IDE resides on your PC and the Runtime resides on the programmable automation controller (PCMM2G).
- For optimum operation, the KAS-IDE installed and run on the PC must be compatible with the Runtime installed on the controller or programmable drive.

The KAS installer and controller Runtime contain the version information in the file name.

The version is determined by:

Version Source	Found From	Example
KAS-IDE, Controller, or Simulator	Help menu > About	 **Figure 6-1:** Example: Product details - About Version
PCMM2G	Web server Settings Firmware tab	 **Figure 6-2:** Example: PCMM2G Firmware version

The compatibility between the KAS-IDE and the Runtime is defined by matching the digits in the versions.

The KAS-IDE and Runtime use the same version scheme:

```
major . minor . micro . revision
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- If the KAS-IDE to Runtime major.minor is:
  - NOT equal, they are NOT compatible.
  - Is equal, but the micro is NOT equal, they are not 100% compatible.
- If the KAS-IDE to Runtime major.minor.micro is equal, they are compatible.

Examples

IDE	Runtime	Compatible?	IDE to Runtime Connection
5.00 .xx.xxxx	1.2.xx.xxxx	NO No connection.	The KAS-IDE displays an error message.
5.00.0.xxxx	5.00.1.xxxx	YES*	Connection possible. The KAS-IDE displays warning and requires user to click OK to continue.
5.00.0.xxxx	5.00.0.xxxx	YES	Normal connection.
5.00 .0.30540	5.00 .0.30540	YES	Normal connection.

*Not a recommended configuration.

- The KAS-IDE operates but there can be different features available between the KAS-IDE and the Runtime.
- It is recommended to upgrade either the KAS-IDE or the Runtime to matching versions.

6.1.1 Digital Signatures

The KAS-IDE Installer and the executables are digitally signed to protect the integrity of these files and identify the distributor of the software components.

The digital signature appears as part of the file properties.

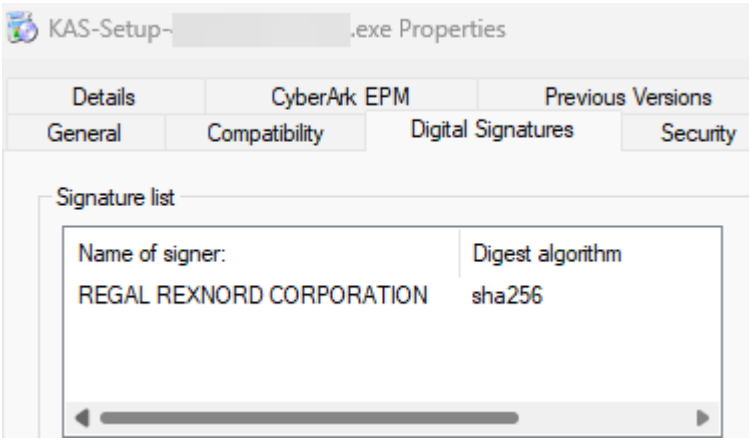


Figure 6-3: Example: KAS Setup Properties Digital Signature

6.2 KAS PLC Code Runtime Compatibility

It is possible after a controller runtime upgrade, that the runtime and the compiled project versions `major.minor.micro` may be different.

The Runtime does not start if there is an incompatibility between the versions.

To correct the incompatible versions either:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Re-compile and download the KAS project file using the same KAS-IDE version as the runtime version.
- Change the controller's runtime version to match the compiled project's version.

6.3 KAS-IDE Project File Compatibility

This section describes forwards and backward compatibility between the KAS-IDE and project files.

6.3.1 Backward Compatibility

The KAS-IDE is backward compatible, meaning the KAS-IDE can open project files made with older versions of the KAS-IDE software.

Notes

- An older project file is converted to take advantage of features in the newer version.
- There may be multiple conversions needed to update the file.
  - If any step is not backward compatible you are warned with a pop-up message and the process continues with the next conversion step.
- You can exit without saving the project conversion.
- Select the **Save as new project** check box to make a copy of the newly converted project file, preserving the original.

6.3.1.1 Update Options

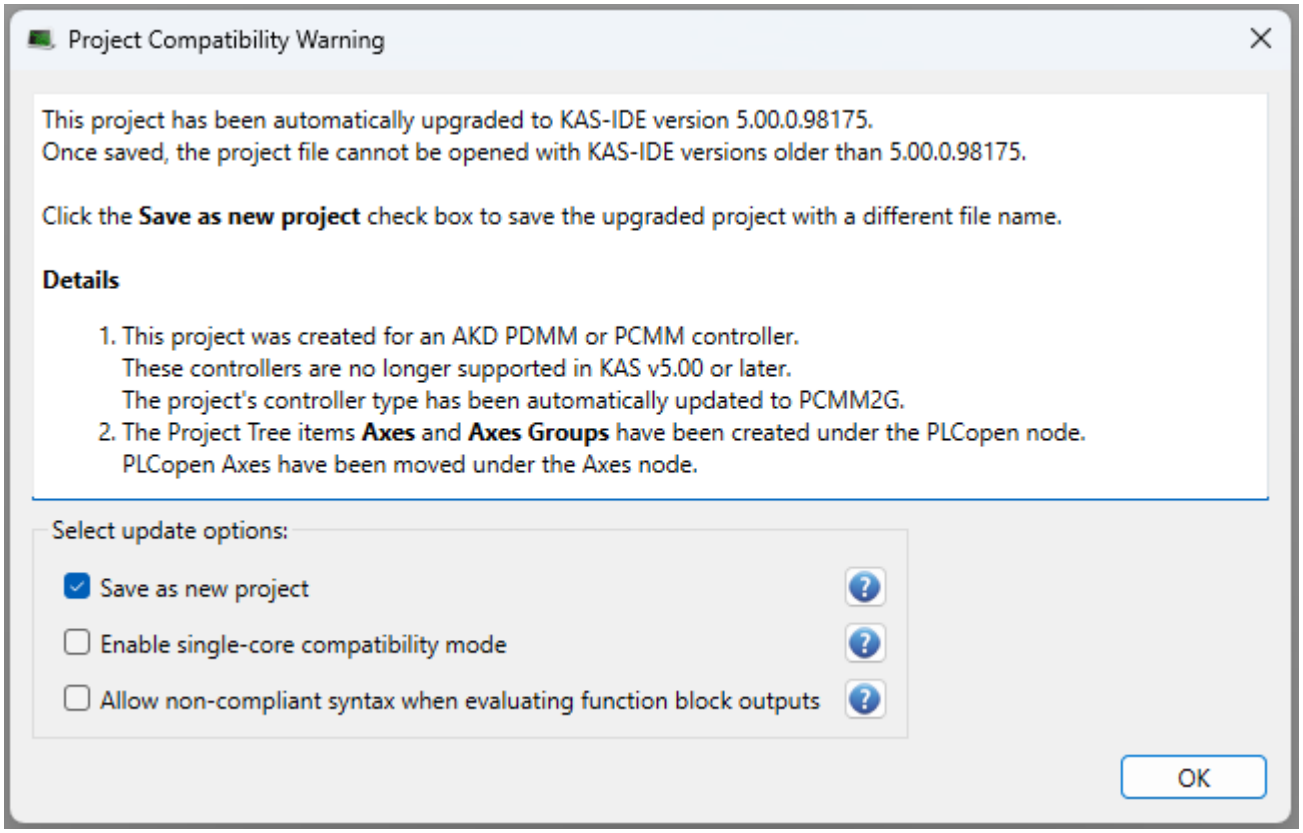


Figure 6-4: Project Compatibility Warning message

1. Select the **Save as new project** check box to make a copy of the newly converted project file. This preserves the original project file.
2. Select the **Enable Single-Core Compatibility Mode** check box if:
   - The application does not follow multi-core controller programming guidelines.
     - See "Program a Multi-Core Controller" (→ p. 386).
   - Reduced performance on the PCMM2G is acceptable for the application.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

IMPORTANT

The limitations listed in "Single-Core Compatibility Mode" (→ p. 161) are acceptable.

3. Select the **Allow non-compliant syntax to evaluate function block outputs** check box if:
   - The application was developed using a version of KAS earlier than v4.00.
   - You are encountering Unexpected complex data in expression compile errors.
   - You prefer not to update the KAS project to use compliant syntax for the corresponding function blocks.
     - See "Unexpected complex data in expression." (→ p. 798).

6.3.2 Forward Compatibility

The KAS-IDE strives to be forward compatible, meaning it can open project files made with newer versions of the software.

If there are no compatibility issues, the KAS-IDE opens the file.

6.4 Start the KAS-IDE

There are multiple ways to start the KAS-IDE.

- Double-click a desktop shortcut.
- Double-click a project file to open it with the most recently installed version of the KAS-IDE.
- **Windows 10 and 11**: From the Start menu, under "**K**".
  - All installed versions are found here, including one legacy version (in the Kollmorgen folder), and all installed versions of v3.01 and later.
  - See "Can I install more than one version of the KAS-IDE?" (→ p. 781).

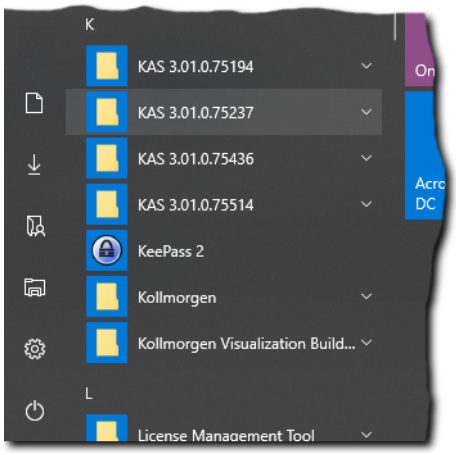


Figure 6-5: Example: Multiple KAS installed versions

TIP

- The KAS-IDE supports simultaneously running multiple instances.
- Double-click a shortcut or project file, or selecting the application from the Start menu, opens another instance.
- Double-click a project file more than once opens a copy of the project in another instance of the KAS-IDE.

See the these FAQs for more options:

- "How do I launch my project with a particular version of the KAS-IDE?" (→ p. 782)
- "How do I choose the default version of the KAS-IDE to use?" (→ p. 783)

6.4.1 Find the KAS-IDE Version

The version of the KAS-IDE or KAS Simulator being used is visible in the title bar of the application. (Figure 6-6)

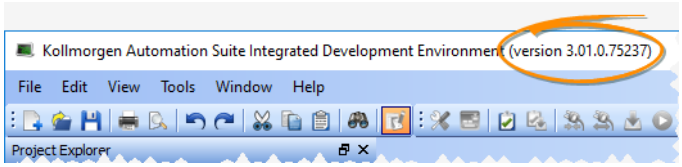


Figure 6-6: Example: Version in KAS Title bar

Click the **About** command in the **Help** menu to view this information. (Figure 6-7)

See "Help menu" (→ p. 463).

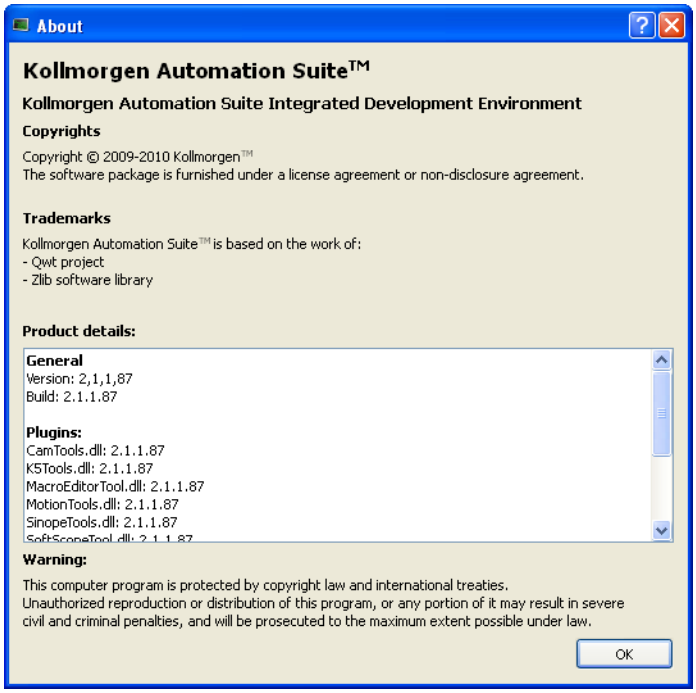


Figure 6-7: About Window

This window displays the application versions as well as all the plug-in versions included in the KAS-IDE and loaded during start up.

6.4.2 Access the KAS-IDE Help System

In the **Help** menu, use the **Documentation** command to open the online help.

See Use the Context-Sensitive Help.

6.4.3 KAS GUI

For a better understanding of KAS menus, toolbar, and workspace items (description and manipulation), see "KAS Graphical User Interface Description" (→ p. 83).

6.4.4 KAS-IDE Local Log tab

The KAS Local log tab (Figure 6-8) provides a running display of activity related to the execution of the application. Items displayed include application startup and initialization information.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

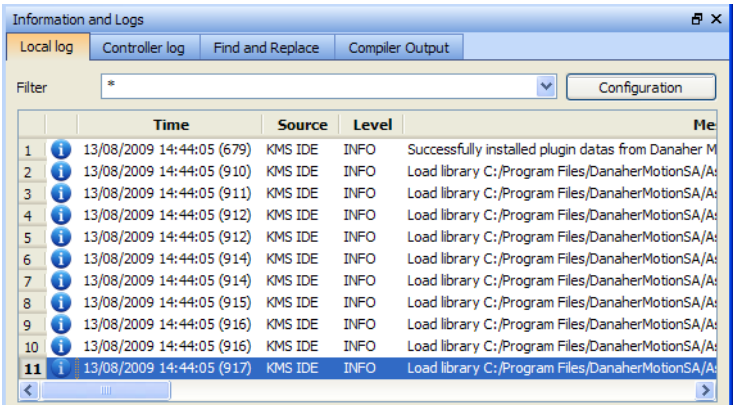


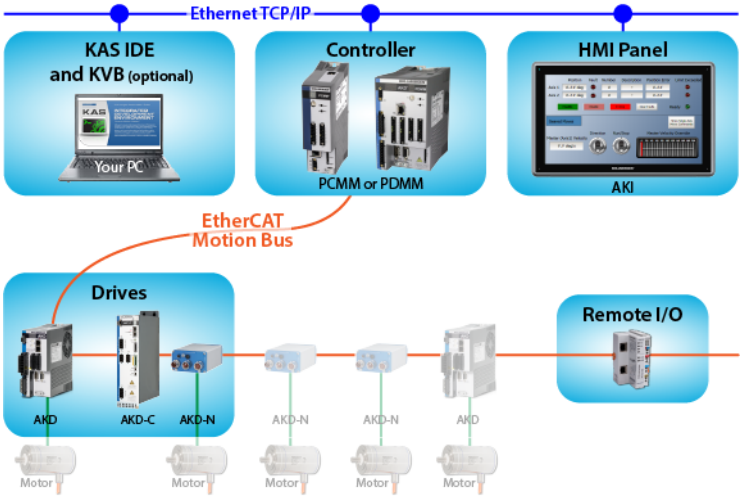
Figure 6-8: Log Messages

See "Information and Logs window" (→ p. 123).

6.4.5 Create a Project

These are the basic aspects of setting up a KAS project:

1. Setting up the controller.
2. Setting up the drives.
3. Defining the remote I/O.
4. Setting up the EtherCAT Motion Bus.
5. Creating the program in the KAS-IDE.
6. Setting up the HMI panel.



6.4.5.1 Add the Controller

1. Click the **File** menu and select **New**.
   The **Create a New Project** dialog box opens. (Figure 6-9)
   By default, PCMM2G is selected as the **Controller**.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

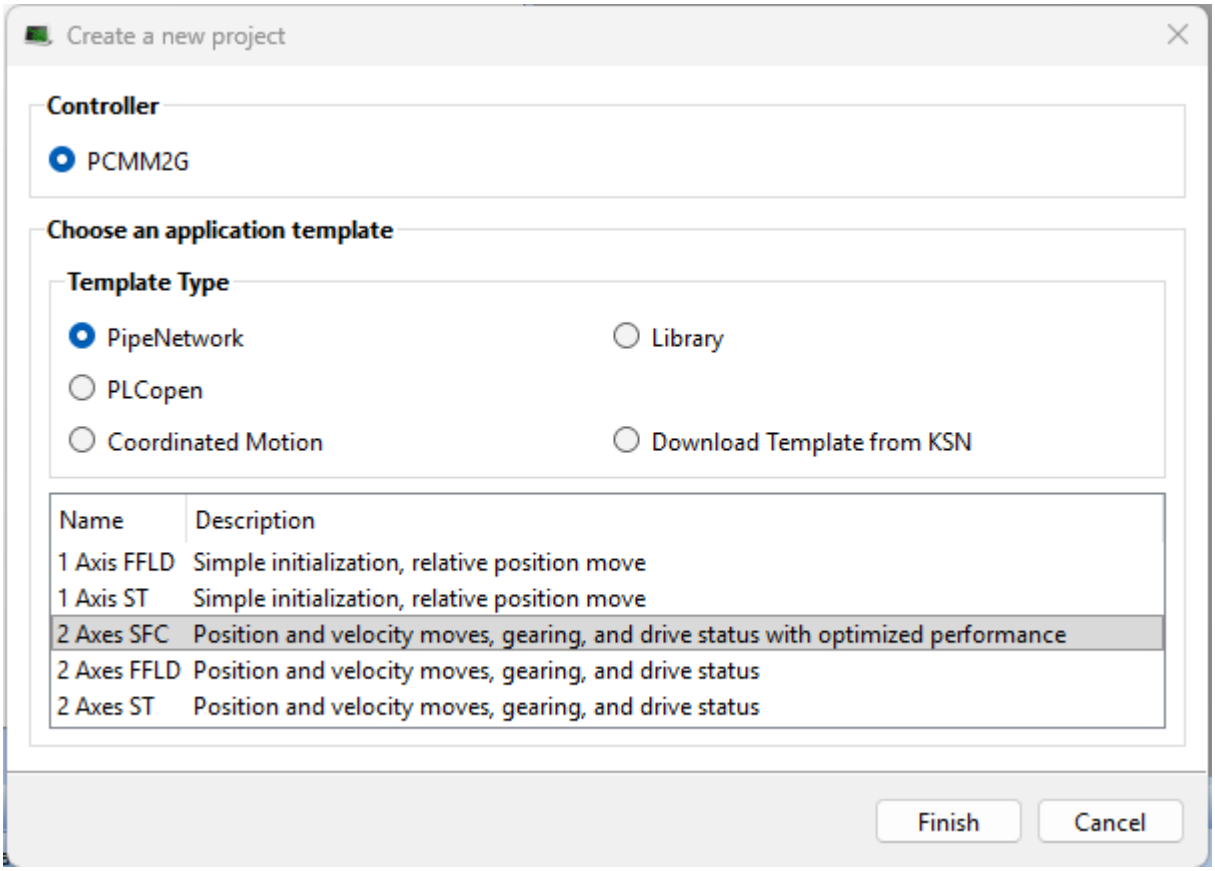


Figure 6-9: Create a new project dialog box

2. In the **Choose an application template** section:
   a. In the **Template Type** section, choose the motion engine option .
   b. In the Template list box, select an application template.
   See the "Available Application Templates" (→ p. 154) list).
3. Click **Finish**.
4. Click the **File** menu and select **Save As**.
   The Save As dialog box opens.
5. Define the Project Name and its location.
6. Click **OK** to save the changes or selections and close the dialog box.

6.4.5.1.1 Available Application Templates

List of available application templates

- "Coordinated Motion" (→ p. 154)
- "KAS Runtime" (→ p. 155)
- "KDN Download" (→ p. 155)
- "Pipe Network" (→ p. 155)
- "PLCopen" (→ p. 155)

6.4.5.1.2 Coordinated Motion

Template Name	Description
"Coordinated Motion - 2-Axis Template" (→ p. 1020)	• Linear / Circular • Raster Scan Motion Path • 2 PLCopen axes

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Template Name	Description
"Coordinated Motion - 3-Axis Template" (→ p. 1021)	• Linear / Circular • Raster Scan Motion Path • 1 Pipe Network axis • 2 PLCopen axes
"Coordinated Motion - 3-Axis (3D) Template" (→ p. 1022)	• Linear 3D • Diamond / Square Motion Path • 3 PLCopen axes

6.4.5.1.3 KAS Runtime

Template Name	Description
Library	Used to create a custom library. See also see "Create and Use Custom Libraries" on page 244.

6.4.5.1.4 KDN Download

Template Name	Description
KDN	1. Click **Finish** to close the dialog. A web browser page opens on KDN with custom-built template files. The template files may have enhanced or specific functionality. 2. Download and save a template file. 3. Open it from KAS-IDE.

6.4.5.1.5 Pipe Network

Template Name	Description
"Pipe Network 1-Axis Template - FFLD" (→ p. 1025)	Simple initialization, relative position move.
"Pipe Network 1-Axis Template - ST" (→ p. 1025)	Simple initialization, relative position move.
"Pipe Network 2-Axes Template - FFLD" (→ p. 1034)	Position and velocity moves, gearing, camming, and drive status.
"Pipe Network 2-Axes Template - ST" (→ p. 1031)	Position and velocity moves, gearing, camming, and drive status.
"Pipe Network 2-Axes Template - FBD, FFLD, SFC, and ST" (→ p. 1026)	Position and velocity moves, gearing, camming, and drive status with optimized performance.

6.4.5.1.6 PLCopen

Template Name	Description
"PLCopen1-Axis Template - FFLD" (→ p. 1037)	Simple initialization, relative position move.
"PLCopen 1-Axis Template - ST" (→ p. 1038)	Simple initialization, relative position move.
"PLCopen 2-Axes Template - FFLD" (→ p. 1047)	Position and velocity moves, gearing, camming, and drive status.
"PLCopen 2-Axes Template - ST" (→ p. 1043)	Position and velocity moves, gearing, camming, and drive status.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Template Name	Description
"PLCopen 2-Axes Template - FFLD and SFC" (→ p. 1039)	Position and velocity moves, gearing, camming, and drive status with optimized performance.

6.4.5.2 Configure the Controller

1. In the Project Explorer, right-click the new controller to open the menu and click **Properties**.
   The Controller Properties dialog box opens. (Figure 6-10)

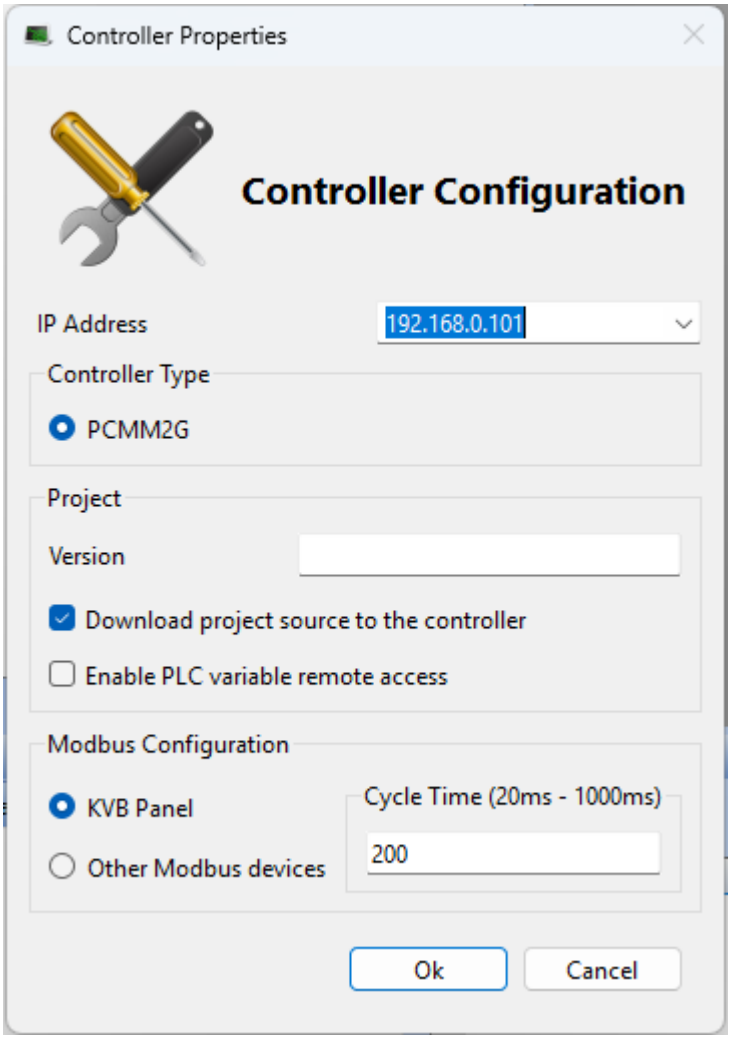


Figure 6-10: Controller Properties dialog box

2. Define the IP address.

NOTE

- For the KAS Runtime Simulator, enter the local host IP address: 127.0.0.1.
- The KAS-IDE supports accessing controllers configured using Fully Qualified Domain Names (FQDN) (e.g., 10.8.45.212.ip.net, kas-pcmm2g-01.lap.company.com), instead of raw IP addresses.
  - This enables seamless connectivity in environments where hostname-based access is required.
  - This capability is particularly beneficial in enterprise environments with remote or secure access configurations and for customers utilizing similar DNS-based setups.

3. Optional: Specify a version number (the string can be composed of any character).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TIP

- Versioning is useful to make improvements to the application and if a version control system is needed.
- The version is saved in your project file.

4. Optional: Choose whether the project's source code should be downloaded to the controller.
   - This is enabled by default and your preference is saved with the project.
   - Disabling this option means a comparison of source on the controller and in the IDE is not available.
5. Optional: Choose whether PLC variables are read/written via an HTTP connection.
   This requires that the source be downloaded to the controller.
   By default this option is disabled so unauthorized changes may not be made.

TIP

Recompile the project before downloading it to the controller.

6. Select whether Modbus goes to a KVB Panel or is handled by another Modbus device.
7. Set the **Cycle Time** (i.e., the communication update frequency).
   See "Modbus Slave" (➔ p. 907).

NOTE

- If KVB Panel is selected, there is no node to add a Modbus configuration in the Fieldbus Editor. (Figure 6-11)

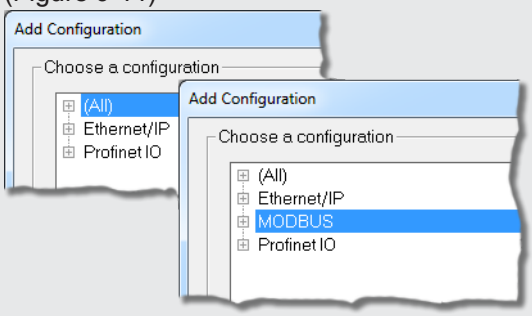


Figure 6-11: Example: KVB Panel

- If Other Modbus devices is selected, the KVB checkbox for Dictionary items is hidden. (Figure 6-12)
  Motion configuration must be added from the Fieldbus Editor.

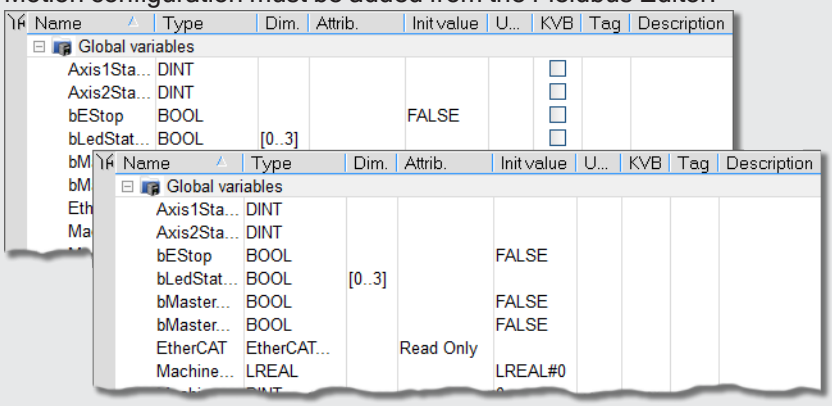


Figure 6-12: Example: Other Modbus Devices

8. Click **OK** to save the changes or selections and close the dialog box.

6.4.5.3 Configure Project Settings

1. On the **Project** menu, click **Settings**.
   The "Project Settings dialog box" (➔ p. 461) opens. (Figure 6-13)

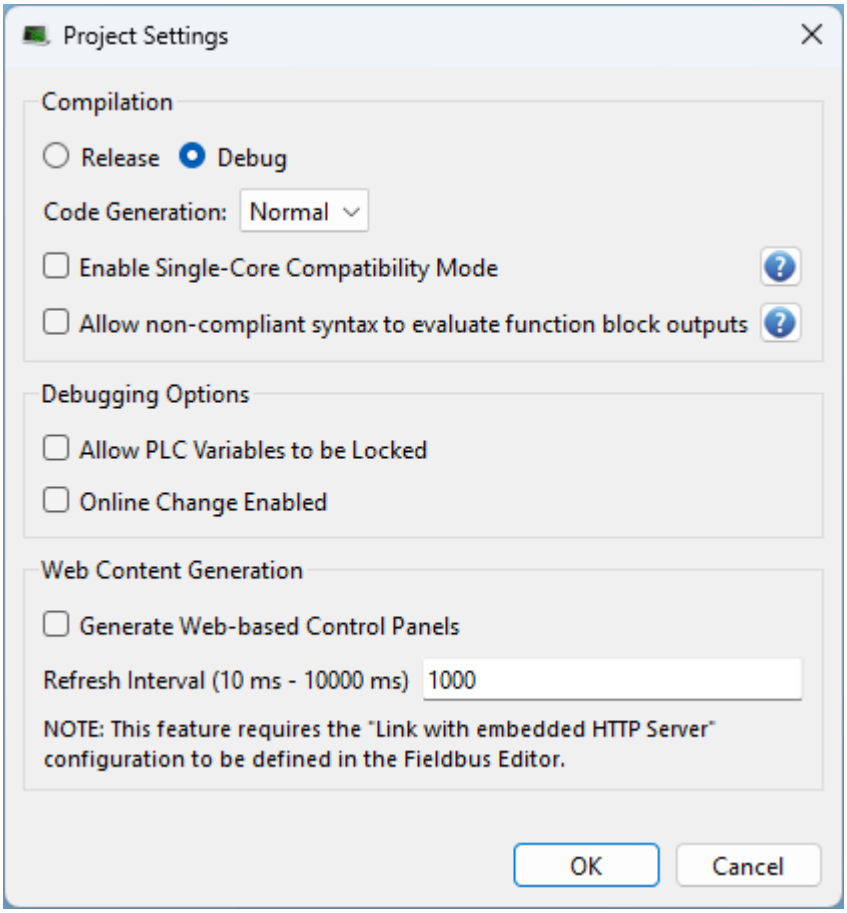


Figure 6-13: Project Settings dialog box

2. In the **Compilation** section:
   a. Select either the **Release** or **Debug** option button.
      - Select the **Release** option button:
        - If step-by-step debugging is not required.
        - To give highest performance to the application.
      - Select the **Debug** option button to use "Step-By-Step Debugging" (→ p. 268) during simulation or online testing.

        NOTE

        - The conditional compilation option _DEBUG is automatically defined when this option is selected.
        - See how the application is compiled in the See Conditional Compiling example.

   b. Click the **Code Generation** drop-down arrow and select either **Normal** or **Optimized**.
      - See "Normal" (→ p. 258).
      - See "Optimized" (→ p. 259).
   c. For PCMM2G only:
      i. Select the **Enable Single-Core Compatibility Mode** check box if:
         - The application does not follow multi-core controller programming guidelines.
           - See "Program a Multi-Core Controller" (→ p. 386).
         - Reduced performance on the PCMM2G is acceptable for the application.

           IMPORTANT

           The limitations listed in "Single-Core Compatibility Mode" (→ p. 161) are acceptable.

      ii. Select the **Allow non-compliant syntax to evaluate function block outputs** check box if:
         - The application was developed using a version of KAS earlier than v4.00.
         - You are encountering Unexpected complex data in expression compile errors.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- You prefer not to update the KAS project to use compliant syntax for the corresponding function blocks.
  - See "Unexpected complex data in expression." (➔ p. 798).

3. In the **Debugging Options** section:
   a. Select the **Allow PLC Variables to be Locked** check box to enable PLC variables to be locked (forced).
   b. Select the **Online Line Change Enabled** check box to update the PLC application while it is running on the controller.
4. In the **Web Content Generation** section for PCMM2G only:
   a. Select the **Generate Web-based Control Panels** check box to allow access to control panel files in the project from the web browser using the KAS Web server.
   b. In the **Refresh Interval** text box, specify the interval at which web pages are refreshed to receive PLC variable updates from the KAS Web server.
5. Click **OK** to save the changes or selections and close the dialog box.

6.4.5.3.1 Configure Custom Project Settings

The KAS Runtime is used to modify specific configurations within a user's application without needing to access the KAS-IDE, recompile and re-downloading the project file to the controller.

- This functionality streamlines configuration management, particularly for customers commissioning multiple machines using a single KAS project file.
- This custom project configuration feature supports modifying the Server IP addresses of the EtherNet/IP IO Scanner (Client) defined in the KAS project.

Procedure

1. Create the controller settings file.
   a. Create a text file named **controller_settings.xml** with this content:

```
<?xml version="1.0" encoding="utf-8"?>
<KasControllerConfig xmlns="KasControllerConfig">
    <IpAddrMapping>
        <IpAddrAlias alias="1.1.1.1" canonical="192.168.1.91" />
        <IpAddrAlias alias="2.2.2.2" canonical="192.168.1.177" />
        <IpAddrAlias alias="3.3.3.3" canonical="192.168.1.213" />
    </IpAddrMapping>
</KasControllerConfig>
```

IMPORTANT

The alias IP addresses **must match exactly** the Server IP Address specified in the KAS project under the Fieldbus Editor > EtherNet/IP IO Scanner (Client) -> Server nodes.

   b. Replace the canonical IP addresses in the configuration file with the actual EtherNet/IP addresses of your adapter controllers.
2. Upload the file to the scanner controller.
   a. Open a command prompt.
   b. Using curl, run these commands to upload the file:
      i. `curl.exe -k -X POST -d "username=administrator" -d "password==<<administrator_account_password>>" -G https://(IP_Address)/authorize/login -c cookie.txt`
      ii. `curl.exe -k -F "sendfile=@<<controller_settings.xml file path>>" -F "filename=controller_settings.xml" https://(IP_Address)/userfiles -b cookie.txt`

NOTE

- Replace the (IP_Address) with the actual IP address of your scanner controller (PCMM2G).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Example:
  ```
  curl.exe -k -F "sendfile=@<<C:\tmp\controller_
  settings.xml" -F "filename=controller_settings.xml"
  https://192.168.0.10/userfiles -b cookie.txt
  ```
- Replace **<<administrator_account_password>>** with the actual password of the controller Web server.
  - See User Authentication.
- See HTTP Commands for User Data File Management.

3. Verify the file upload.
   a. Open a web browser and navigate to https://(IP_Address).
   b. Navigate to: KAS Application > User Data page.
   c. Locate the **controller_settings.xml** file and click the Download ( ) icon next to it. (Figure 6-14)
   d. Once the file is downloaded, open it and verify that its contents are correct and as expected.

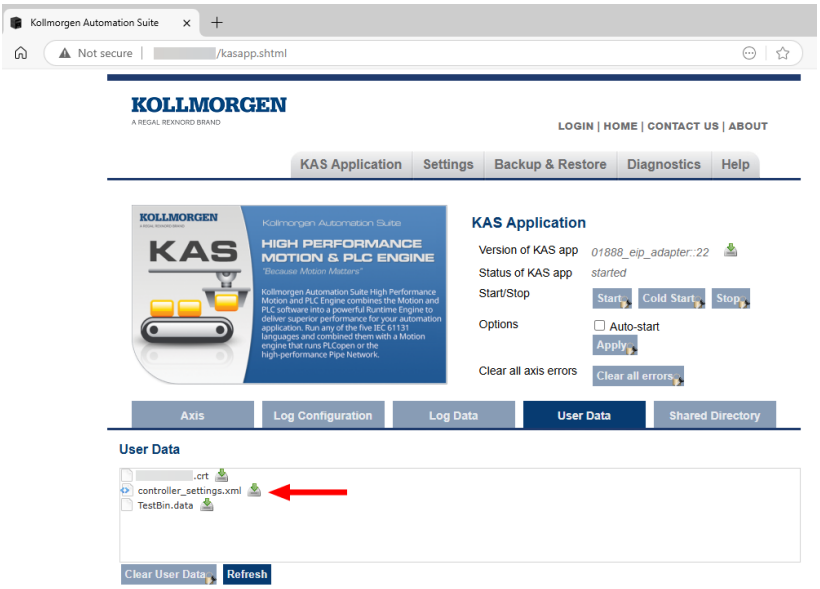


Figure 6-14: controller_settings.xml file

4. Configure the KAS project.
   a. In the KAS-IDE, open the KAS application project.
   b. Stop the project if it is currently running.
   c. In the Project Explorer, double-click Fieldbus to open the FieldBus Editor.
   d. Under the Server nodes, update the IP address to match the alias values from the XML file. (Figure 6-15)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

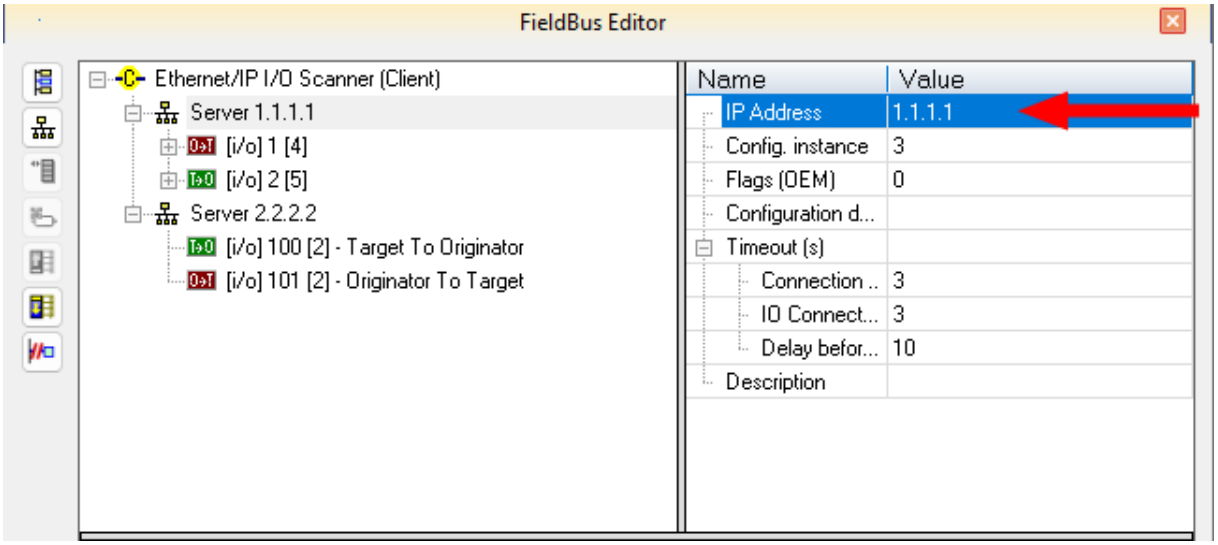


Figure 6-15: Alias IP address

5. Verify the operation.
   a. Compile and download the project to the controller.
   b. Start the project and wait for the application to fully initialize.
   c. Confirm that all EtherNet/IP I/O Scanners are operating correctly and communicating as expected.

6.4.5.4 Single-Core Compatibility Mode

Single-Core Compatibility Mode enables migration of existing PxMM (first-generation) projects to the PCMM2G platform.

- **Configuration**: This setting is enabled for each project.
  - See Configure Project Settings.
- **Execution Behavior**: This mode forces the motion engine and PLC virtual machine to run on the same core, emulating single-core controller behavior.
- **No Code Changes Required**: Applications originally designed for single-core execution run without modification.
- **Diagnostics Support and Persistence**: A warning message is logged at project startup when the compatibility mode is enabled to aid in diagnostics.
  - This setting persists across firmware updates and controller backup/restore operations.
- **Customer Benefit**: Allows continued use of existing applications without the need for immediate code refactoring, easing the transition to multi-core platforms.

6.4.5.4.1 Limitations and Guidelines

This mode preserves compatibility with legacy applications but exhibits operational characteristics that differ from both first-generation controllers and native PCMM2G multi-core operation.

- "Operational Behavior" (→ p. 161)
- "Impact on Migrated Applications" (→ p. 162)
- "Application Guidelines" (→ p. 162)
  - "Recommendations" (→ p. 162)

6.4.5.4.2.1 Operational Behavior

In Single-Core Compatibility Mode:

- The PCMM2G controller does not interrupt PLC code execution when the next motion cycle begins.
- The controller continues executing PLC logic until completion, regardless of cycle time boundaries.

When PLC execution exceeds the motion cycle time:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- EtherCAT communication is delayed until PLC logic completion.
- The controller reports late Tx/Rx messages.
- Distributed clock synchronization is affected.
- E30 errors are likely to occur.

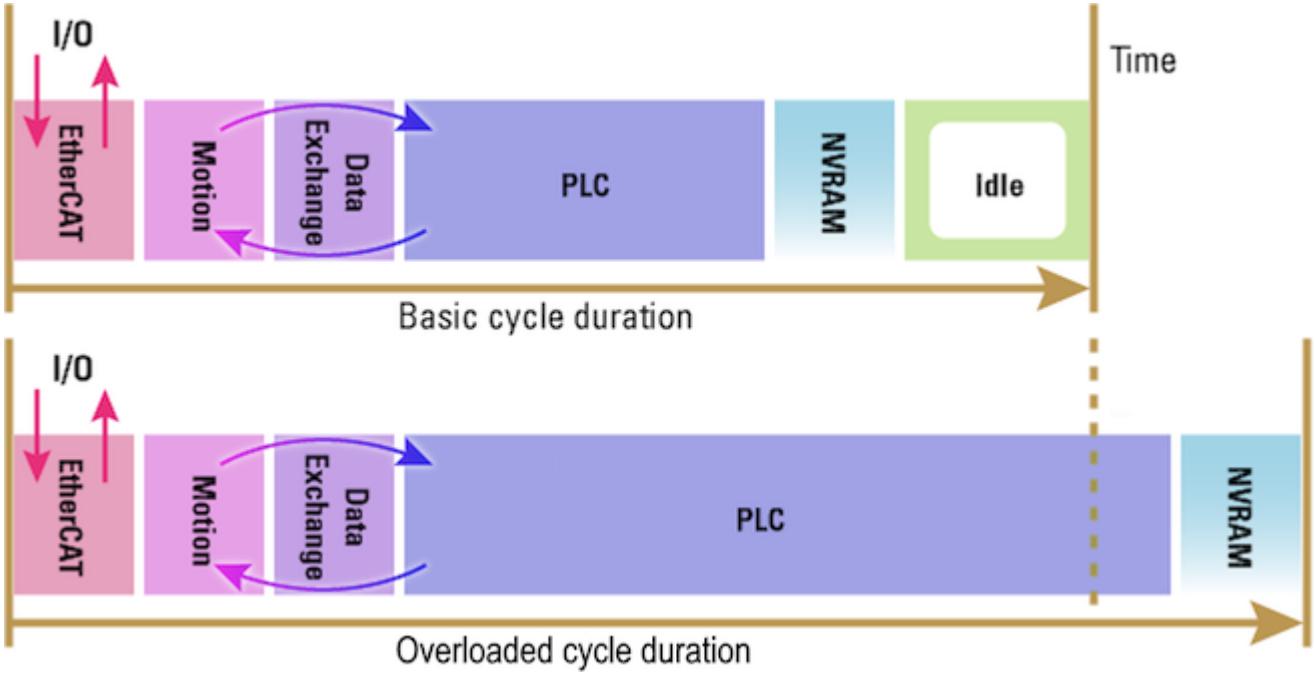


Figure 6-16: Example: SCCM Overloaded Cycle

6.4.5.4.3.2 Impact on Migrated Applications

The PCMM2G provides significantly higher processing performance than first-generation hardware.

Applications that operate within timing constraints on PxMM controllers typically execute with greater timing margin on PCMM2G hardware.

Timing issues may occur when:

- Additional PLC logic increases execution time beyond the original application scope.
- EtherCAT cycle time settings are aggressive relative to total workload.
- Complex computational tasks are added to migrated code.

6.4.5.4.4.3 Application Guidelines

Single-Core Compatibility Mode is appropriate for:

- Migration of existing PxMM controller projects to PCMM2G.
- Applications where minimizing code modification is required.
- Maintaining legacy application behavior during hardware transition.

Multi-core compatible execution is recommended for:

- New projects developed on PCMM2G hardware.
- Applications requiring significant functional expansion.
- Projects where maximum performance and scalability are required.

6.4.5.4.5.4 Recommendations

Kollmorgen strongly recommends updating PLC code for multi-core compatibility when migrating existing code to new projects that increases controller workload.

Mitigation Options

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

If timing constraints are encountered in Single-Core Compatibility Mode, consider these mitigation options:

- **Code Optimization**: Reduce computational complexity in PLC logic.
- **Update the Program Cycle settings**: See Program a Multi-Core Controller.
- **Cycle Time Adjustment**: Increase EtherCAT cycle time if application requirements allow it.
- **Multi-Core Migration**: Update application to use multi-core compatible guidelines.
  - See Define the PLC Cycle.

See Also

"Configure Custom Project Settings" (➔ p. 159)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.6 Configure EtherCAT Motion Bus

TIP

Before configuring the EtherCAT settings, you may want to add third party EtherCAT devices.
See "Add and Configure Third-party EtherCAT Devices" (→ p. 234) for more information.

Double-click the **EtherCAT** node in the Project Explorer to open the EtherCAT properties dialog in the workspace.

This window has these tabs:

Tab	Description
"EtherCAT Devices tab" (→ p. 536)	Displays all the E-Bus devices present in the project tree.
"EtherCAT Master Settings tab" (→ p. 539)	Used to configure the EtherCAT bus master.
"ENI File tab" (→ p. 541)	Used for an external configuration file.
"ESI Files tab" (→ p. 544)	Display, add, and remove available ESI files.

KAS includes an integrated tool to configure the EtherCAT master and start up the fieldbus operation.

Use the configuration tool to:

- Describe your motion "EtherCAT Topology" (→ p. 422) as a configuration tree.
  - See the procedure in "EtherCAT Devices tab" (→ p. 536).
- Associate variables to the I/O channels of devices.
  - See the procedure in "Map Input and Output to Variables" (→ p. 526).

6.4.6.1 Slave Devices

Slave devices can support several PDOs.

NOTE

PDOs contain real-time cyclic data which is deterministic.
Non-cyclic data is not deterministic and is defined by Service Data Objects (ScDO).

In "EtherCAT Motion Bus Concepts" (→ p. 416), input and output parameters are grouped in predefined blocks called PDOs.

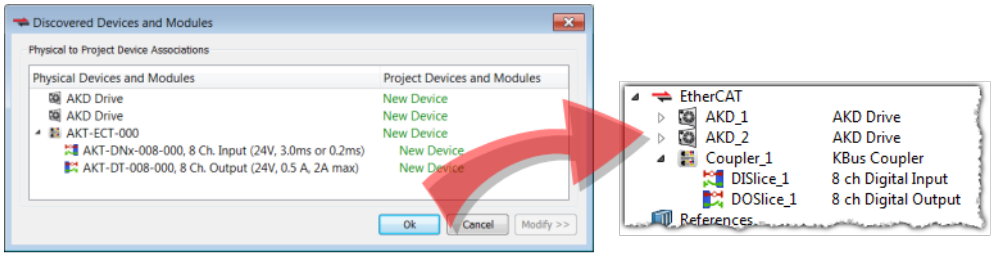
- Some PDOs are mandatory; others are optional.
  - See click here... for the list.
- One of the main tasks of the EtherCAT configuration is to select the PDOs used by each slave and group them all in the EtherCAT image.
  - See "EtherCAT Devices Summary Form" (→ p. 536) for more information.

6.4.6.2 EtherCATScan and Association

Physical EtherCAT devices, modules, couplers, and slices are discovered during the **Scan Devices** routine and are automatically set as project devices in the Project Tree.

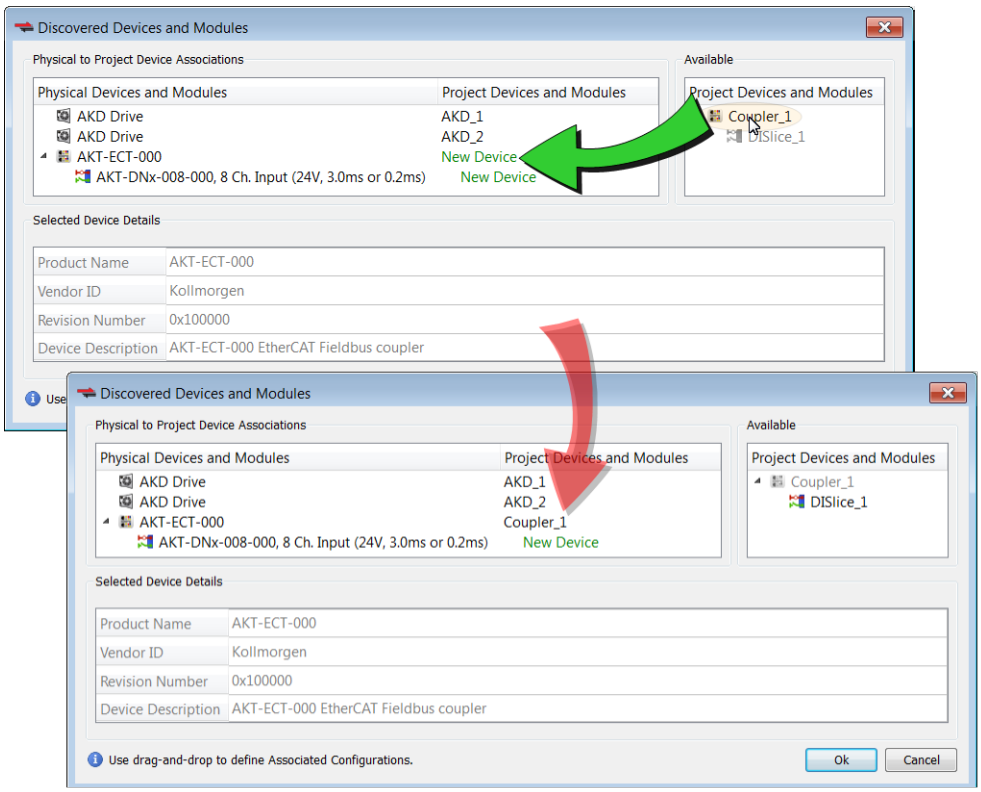
- These device associations persist through subsequent scans unless you change them.
- The **Scan Devices** is accessed by double-clicking the EtherCAT node in the Project Explorer, then click the **Scan Devices** button in the **Devices** tab.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



Changing the network's physical topology (e.g., adding, moving, or removing devices) or changing the associations between physical devices and project devices requires the **Scan Devices** be run again.

- The **Discovered Devices and Modules** window shows any changes.
- The **Modify** button will be enabled.
  - When Modify is clicked the dialog box expands to show Available Project Devices and Modules.
  - Click a device and drag it to the corresponding **New Device** to create an association.
  - Repeat as necessary for slices and modules.



Once the associations are complete, click **Ok**.

See these procedures for more details:

- New projects with no project devices: "Scan and Associate Network Devices" (→ p. 167).
  - This creates default configurations for the discovered physical devices/modules and couplers/slices.
- Adding, removing, or moving devices: "Re-Scan and Change Device Associations" (→ p. 171).
- Adding, removing, or moving couplers with slices: "Re-Scan and Change Coupler/Slice Associations" (→ p. 173).
- Adding, removing, or moving devices with modules: "Re-Scan and Change Device/Module Associations" (→ p. 177).

The physical network topology is compared to the project's expected network topology after the EtherCAT devices are associated and the PLC application is started.

An error is reported if the topologies do not match and the PLC program does not run.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

The physical topology must match the project's expected topology for the PLC application to operate the devices properly.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.6.3 Scan and Associate Network Devices

Use this procedure to discover the physical EtherCAT devices.

1. In the Project Explorer, double-click the **EtherCAT** node to open the **Devices** tab. If there are no devices, the EtherCAT network topology has not yet been scanned and discovered.
2. In the **Devices** tab, click the **Scan Devices** button.

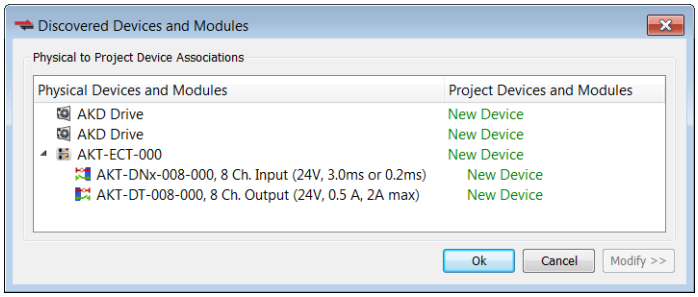
NOTE

The topology discovery is only enabled when the controller is not running an application.

- If the scan process fails, refer to the "EtherCAT Error Messages" (➔ p. 808).



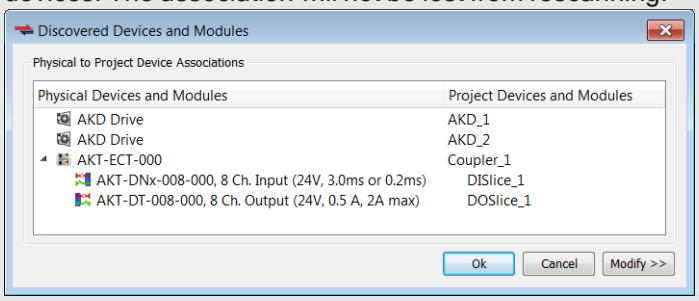
- If physical devices are discovered during the scan, they will appear under the Physical Devices column.



3. Press **OK**. The devices will appear in the **Devices** tab and in the **Project Explorer**. Their associated project devices are set to the defaults.
4. If you have previously compiled your project, you will be asked if you wish to recompile the project.

TIP

If you click **Scan Devices** again, the existing project devices will be automatically matched to the same physical devices. The association will not be lost from rescanning.



To change the device associations see "Re-Scan and Change Device Associations" (➔ p. 171).

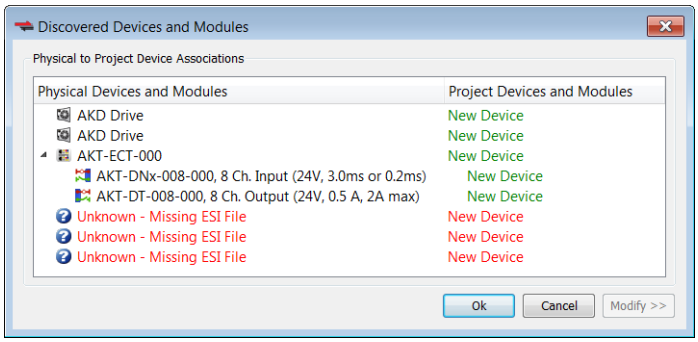
To troubleshoot an error due to a missing ESI file, please see "Unknown - Missing ESI File" (➔ p. 167).

6.4.6.3.1 Unknown - Missing ESI File

The **Discovered Devices and Modules** view identifies a device as "Unknown" if a corresponding ESI file is not found during the Scan for physical EtherCAT devices.

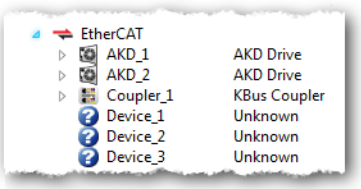
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Example: The last three devices are missing ESI files:



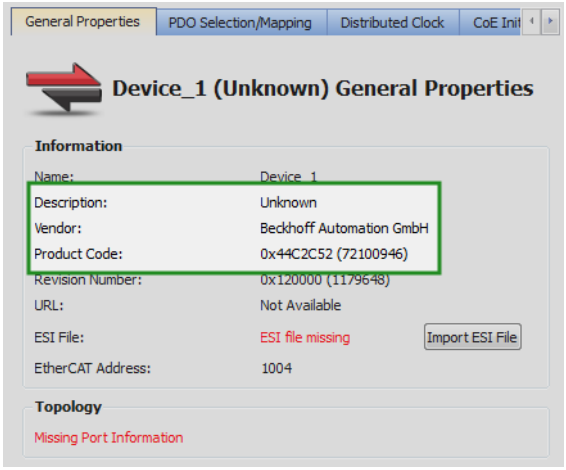
Click **Ok**.

The Unknown devices are added to the Project View:



To correct the problem:

1. Double-click on the “Unknown” device in the Project View or right-click and select **Properties**.
2. From the General Properties view, identify the **Vendor**, **Product Code**, and **Revision Number**.



- Contact the vendor for the EtherCAT ESI file for this device.
- The vendor may provide an ESI file that supports multiple devices or they may provide a separate ESI file for each device.
- Import the ESI file using the **Import ESI File** button and the device specific information is automatically updated in the Project View and General Properties.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.6.3.2 Missing ESI File - MDP Device

During the Scan for physical EtherCAT devices, if a MDP device is discovered that does not have a corresponding ESI file, the **Discovered Devices and Modules** view identify the device as **Unknown** and are not able to discover any modules under the device.

Example: Figure 6-17 shows the last device is missing an ESI file.

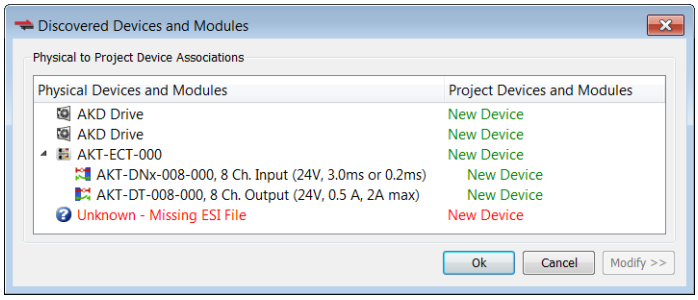


Figure 6-17: Discovered Devices and Modules dialog box

Click the **Ok** button and the **Unknown** device is added to the Project View. (Figure 6-18)

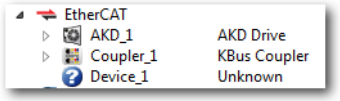


Figure 6-18: Device added to Project View

To correct the problem:

1. Double-click the **Unknown** device in the Project View or right-click and select **Properties**.
2. On the **General Properties** tab, identify the **Vendor**, **Product Code**, and **Revision Number**. (Figure 6-19)

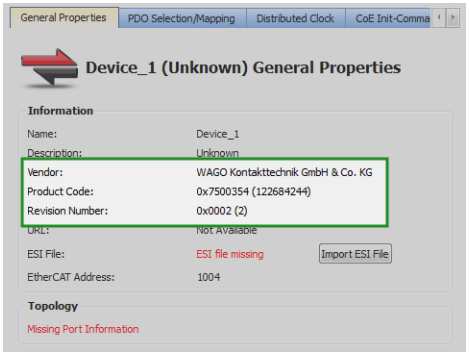
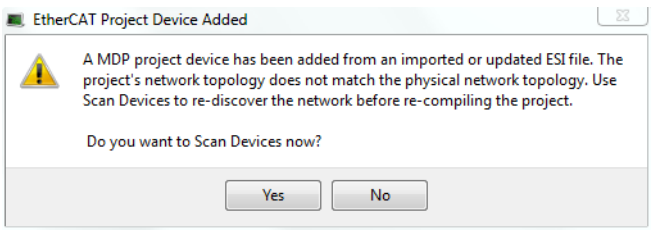


Figure 6-19: General Properties tab

3. Contact the vendor for the EtherCAT ESI filefor this device.
   The vendor may provide an ESI file that supports multiple devices or they may provide a separate ESI file for each device.
4. Import the ESI file using the **Import ESI File** button.
   A message appears indicate that this a MDP device and the EtherCAT network must be scanned to re-discover the modules. (Figure 6-20)



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Figure 6-20: Unknown device message

5. Click **Yes** to continue.

After the scan and association of devices/modules, the MDP device and its modules are added to the Project View. (Figure 6-21)

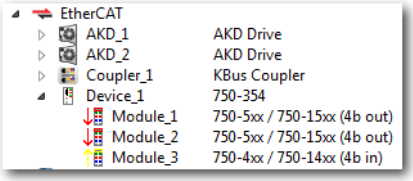


Figure 6-21: Added devices

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.6.4 Re-Scan and Change Device Associations

Device Association must be updated after the following actions.

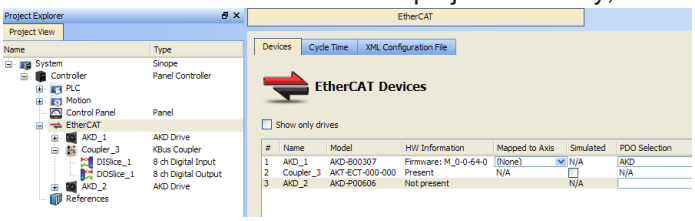
- Changing the physical network topology by adding, moving, or removing EtherCAT devices.
  ⚠CAUTION
  Changing the EtherCAT network topology (plugging and unplugging EtherCAT cables) is supported while power is supplied. Personal injury or damage to devices may occur if EtherCAT devices are installed, wiring is changed, or the terminal configuration is changed while power is supplied ("hot swapping").
- If you want to change the associations between devices and configurations.
- Changing the FSoE address via physical switch settings (DIP switches, rotary switches, etc.) for Safety devices.

To update the associations:

1. Re-scan and re-discover the physical network topology.
2. Click the **Modify** button to expand the Discovered Devices and Modules view.
3. Drag-and-drop the project coupler and/or slice(s) to associate them with the physical couplers/slices.

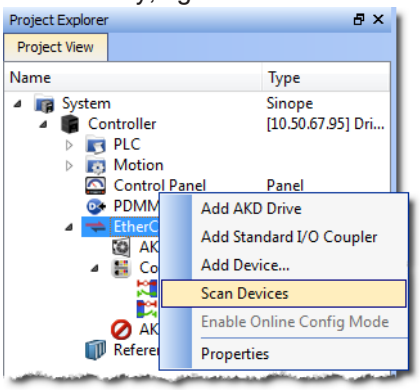
For example, a project has the following configuration:

- The first AKD Drive and a I/O Coupler were created during a previous scan
- The second AKD was added to the project manually, and is not yet associated with any physical drive

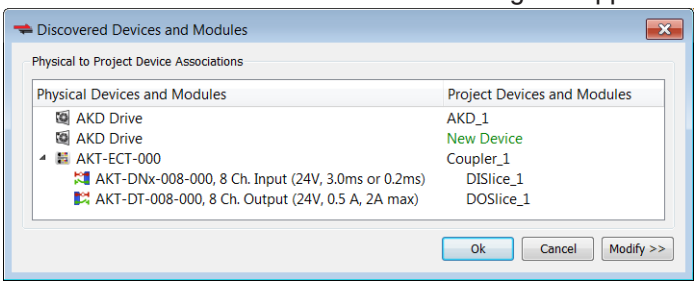


To update the network topology:

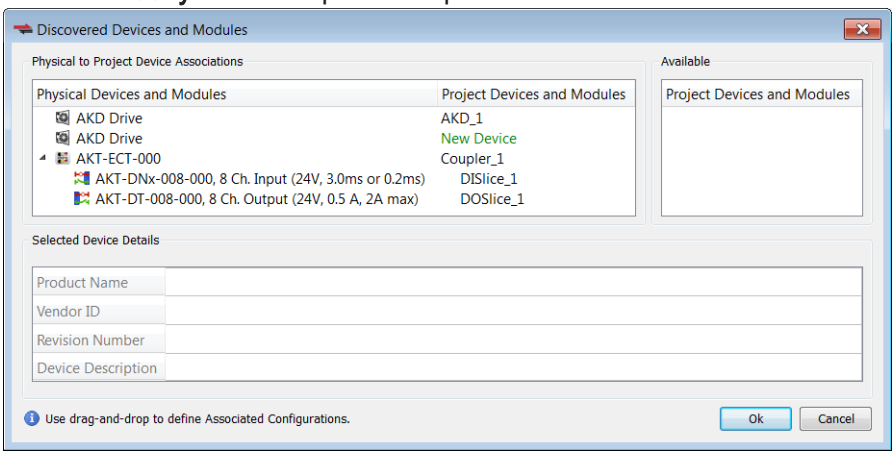
1. In the Project Explorer, double-click the EtherCAT node to open its **Properties** and click the **Scan Devices** button in the Devices tab.
   Alternatively, right-click on the EtherCAT node in the Project Explorer and select **Scan Devices**.



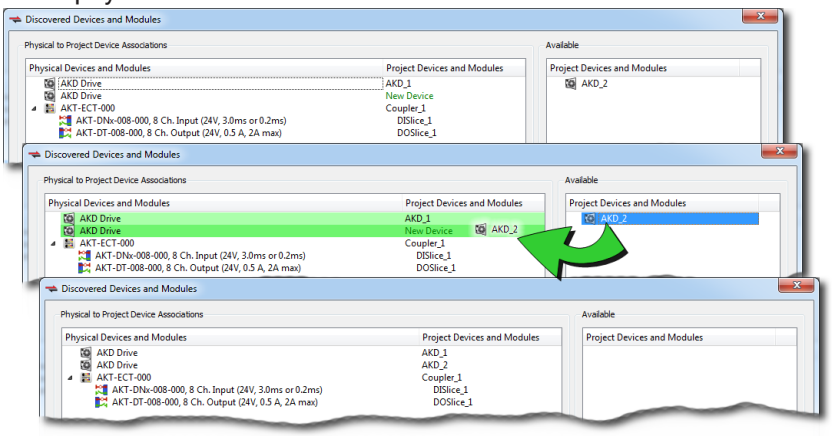
2. The Discovered Devices and Modules dialog box appears after a successful scan.



3. Click the **Modify** button to open the expanded view.

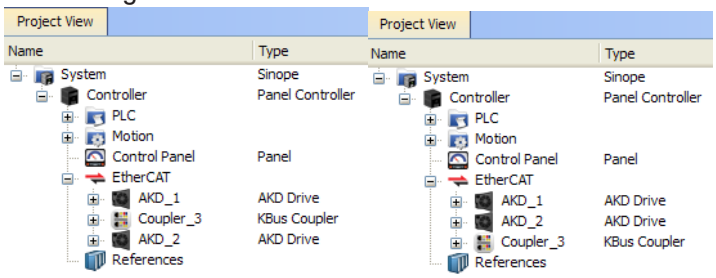


4. Drag-and-drop *AKD_2* from **Available** to the second physical AKD drive. This associates the project device to the physical device.



5. Click **OK** to confirm the modified association.
6. If you have previously compiled your project, you will be asked if you wish to recompile the project.

The KAS-IDE automatically reorders the EtherCAT nodes and the I/O terminals in the Project Explorer to match the physical device order on the network. The following figures show the Project View before and after associating devices.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.6.5 Re-Scan and Change Coupler/Slice Associations

Coupler or slice associations must be updated after the following actions.

- Changing the physical network topology by adding, moving, or removing I/O slices attached to an EtherCAT coupler
- Changing the associations between physical and project I/O slices.

To update the associations:

1. Re-scan and re-discover the physical network topology.
2. Click the **Modify** button to expand the Discovered Devices and Modules view.
3. Drag-and-drop the project coupler and/or slice(s) to associate them with the physical couplers/slices.

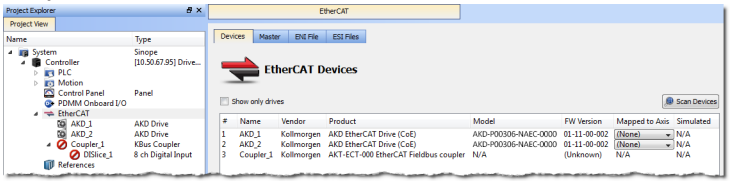
IMPORTANT

After changing the association of an EtherCAT device, you have to recompile the project and download this new version to save your modifications on the target.

6.4.6.5.1 EtherCAT Scan & Association Example 1 - Associating Couplers and I/O Slices

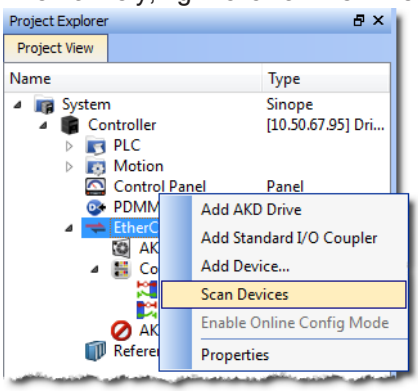
In this example the project has the following configuration:

- Two AKD drives which were created during a previous scan
- A coupler and I/O slice were manually added to the project but are not yet associated with the physical coupler and I/O slice.



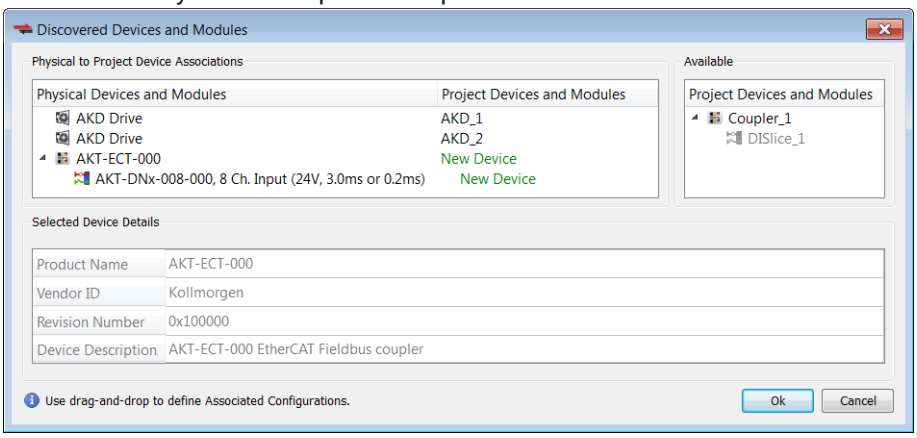
To update the network topology:

1. In the Project Explorer, double-click the EtherCAT node to open its Properties and click the Scan Devices button in the Devices tab.
   Alternatively, right-click on the EtherCAT node in the Project Explorer, and select Scan Devices.

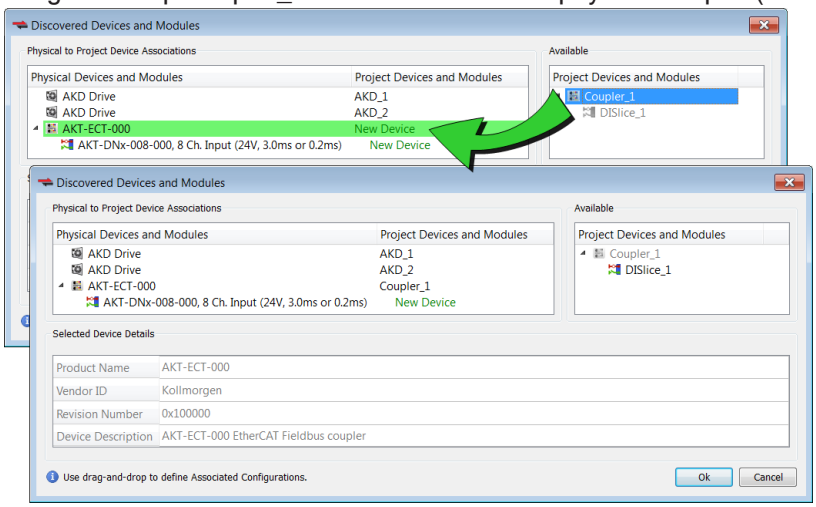


2. The Discovered Devices and Modules dialog box appears after a successful scan.

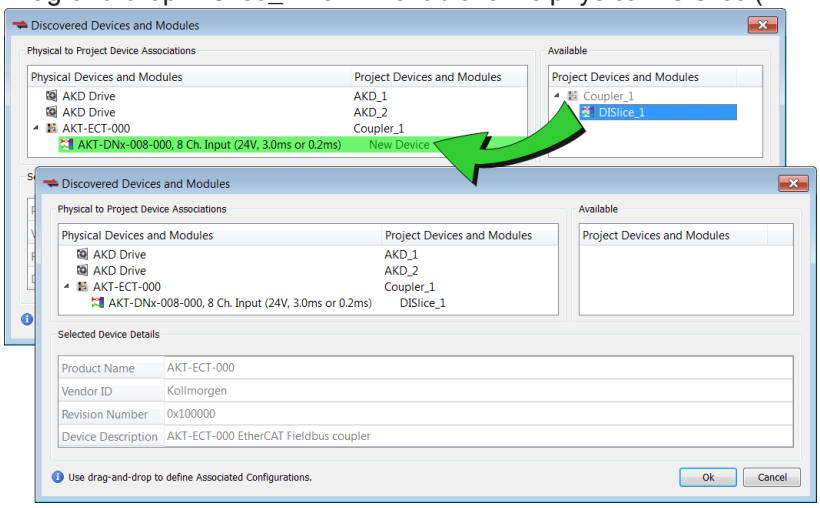
3. Click the Modify button to open the expanded view.



4. To associate the project coupler and I/O slice to the physical coupler and I/O slice:
   1. Drag-and-drop Coupler_1 from Available to the physical coupler (AKT-ECT-000).



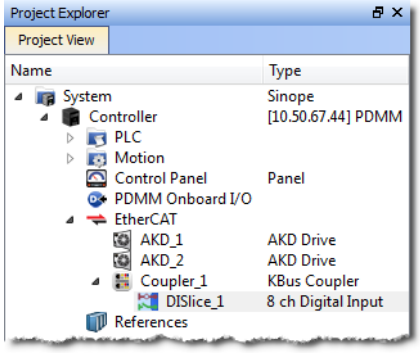
   2. Drag-and-drop DISlice_1 from Available to the physical I/O slice (AKT-DNx-008-000).



5. Click OK to confirm the modified associations.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The KAS-IDE updates the EtherCAT nodes for the coupler and I/O slice in the Project Explorer to match the physical coupler and I/O slice.

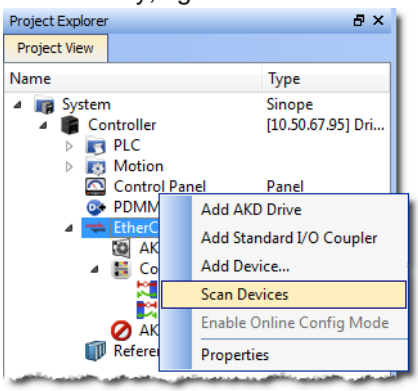


6.4.6.5.2 EtherCAT Scan & Association Example 2 - Adding Physical I/O Slices

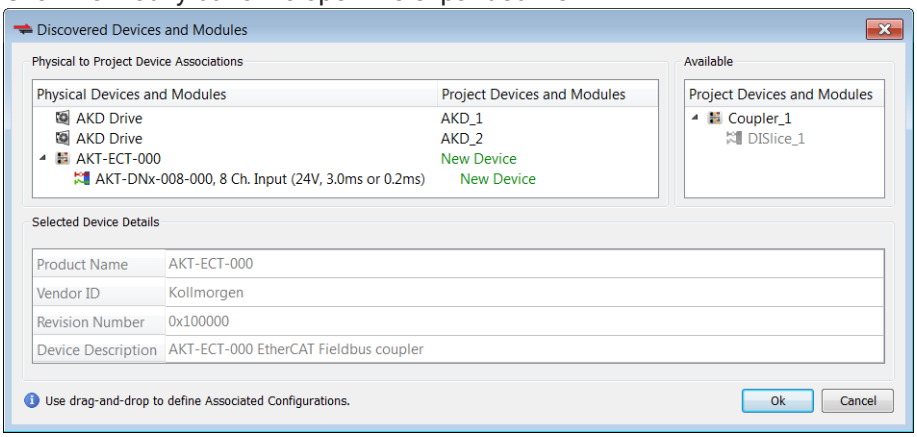
In this example another physical I/O slice is added to the coupler.

To update the network topology:

1. In the Project Explorer, double-click the EtherCAT node to open its Properties and click the Scan Devices button in the Devices tab.
   Alternatively, right-click on the EtherCAT node in the Project Explorer, and select Scan Devices.

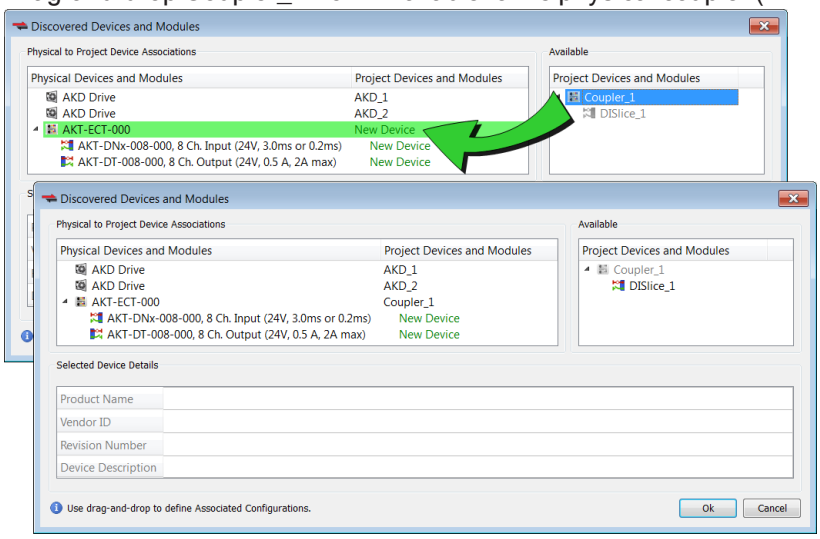


2. The Discovered Devices and Modules dialog box appears after a successful scan.
3. Click the Modify button to open the expanded view.

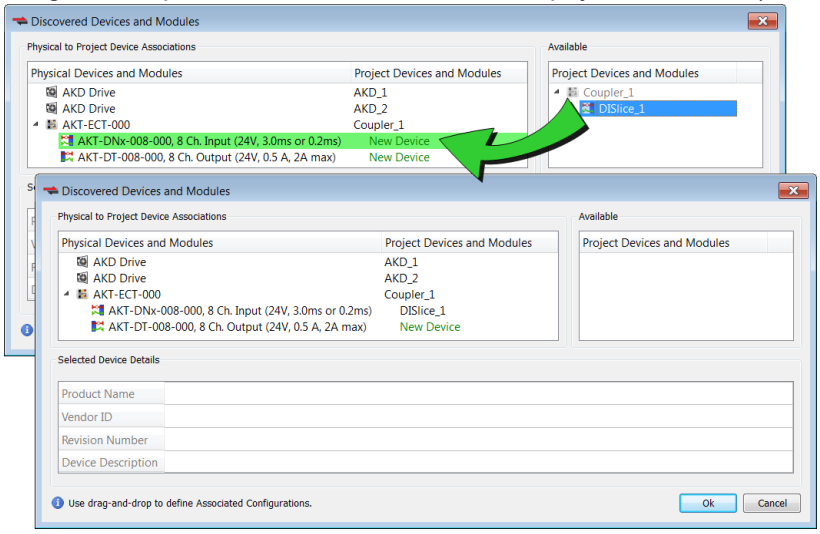


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4. To associate the coupler and I/O slice configurations with the physical coupler and I/O slice:
   1. Drag-and-drop Coupler_1 from Available to the physical coupler (AKT-ECT-000).

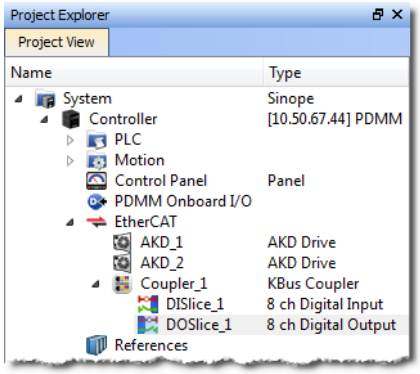


   2. Drag-and-drop DISlice_1 from Available to the physical I/O slice (AKT-DNx-008-000).



5. Click OK to confirm the modified association.

The KAS-IDEcreates the EtherCAT node for the added I/O slice in the Project Explorer to match the physical I/O slice:



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.6.6 Re-Scan and Change Device/Module Associations

Device or module associations must be updated after the following actions.

- Changing the physical network topology by adding, moving, or removing modules attached to an EtherCAT modular device.
- Changing the associations between a physical module and a project module.

To update the associations:

1. Re-scan and re-discover the physical network topology.
2. Click the **Modify** button to expand the Discovered Devices and Modules view.
3. Drag-and-drop the project coupler and/or slice(s) to associate them with the physical couplers/slices.

IMPORTANT

After changing the association of an EtherCAT device, you have to recompile the project and download this new version to save your modifications on the target.

NOTE

A MDP device's configurable properties (PDO selections/content, PLC variable maps, and CoE Init-Commands) are determined by the modules under the device. The default values for the device's configuration are determined from the ESI file.

The device's configuration can be modified from the KAS-IDE EtherCAT device configuration tabs. Although the project modules cannot be configured directly, they can be associated with Physical modules by dragging-and-dropping them in the Discovered Devices and Modules view.

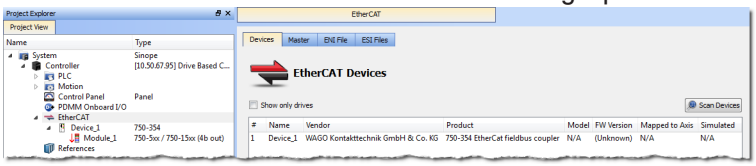
When a project module is associated with a Physical module, its configuration (PDO selections/content, PLC variable maps, and CoE Init-Commands) is automatically updated at the device level. The order of the modules under the device, is determined by order the Physical modules are discovered. The order of CoE Init-Commands also follows the order of the Physical modules. This includes any user specified CoE Init-Commands.

It is recommended to check the CoE Init-Command ordering after changing module associations, to verify it still meets any user-specific ordering requirements.

6.4.6.6.1 EtherCAT Scan & Association Example 3 - Add a Module to a Device

In this example a module is added to the device on the physical network:

- A device with a module which was created during a previous scan.

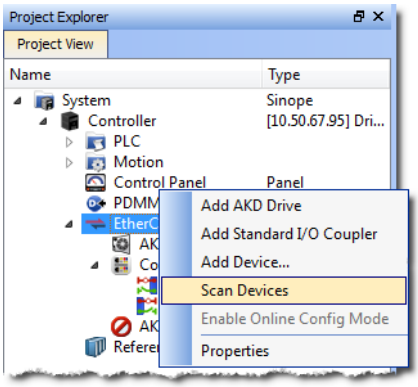


- Another module needs to be added to match the device on the physical network.

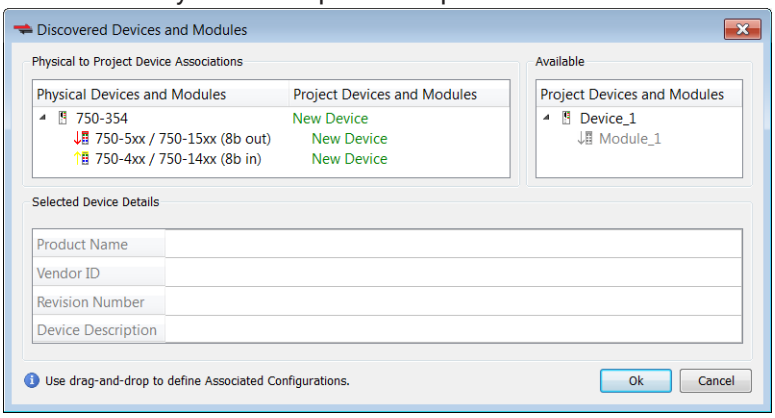
To update the network topology:

1. In the Project Explorer, double-click the EtherCAT node to open its Properties and click the Scan Devices button in the Devices tab.
   Alternatively, right-click on the EtherCAT node in the Project Explorer, and select Scan Devices.

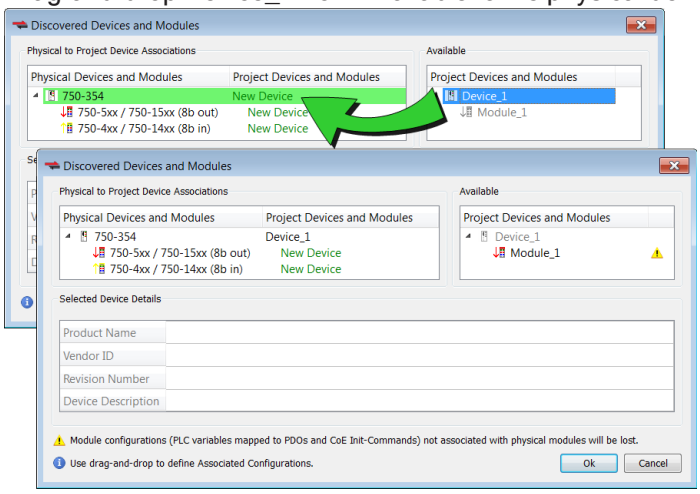
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



2. The Discovered Devices and Modules dialog box appears after a successful scan.
3. Click the Modify button to open the expanded view.

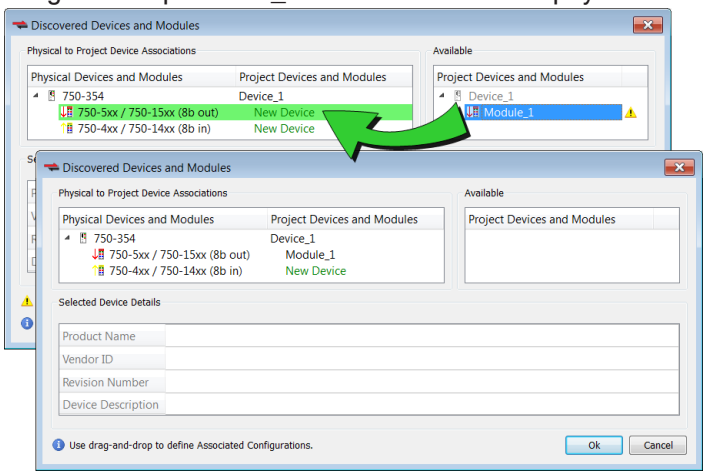


4. To associate the project device and module to the physical device and module:
   1. Drag-and-drop Device_1 from Available to the physical device (750-354).



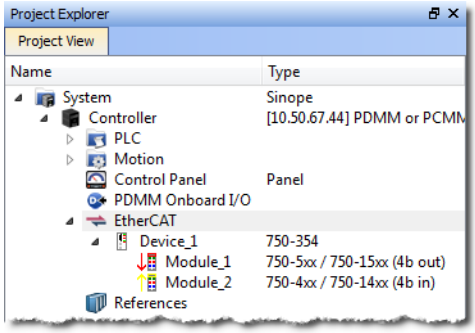
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

2. Drag-and-drop Module_1 from Available to the physical module (750-5xx).



5. Click OK to confirm the modified associations.

The KAS-IDE updates the EtherCAT nodes for the device and modules in the Project Explorer to match the physical device and modules.

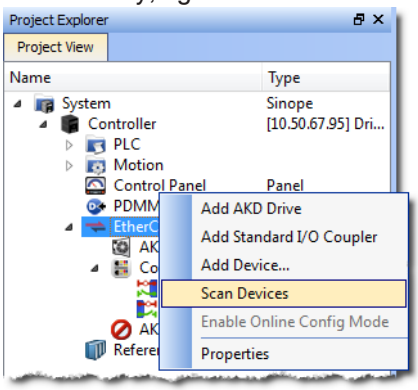


6.4.6.6.2 EtherCAT Scan & Association Example 4 - Remove a Module From a Device

In this example a module is removed from the device on the physical network.

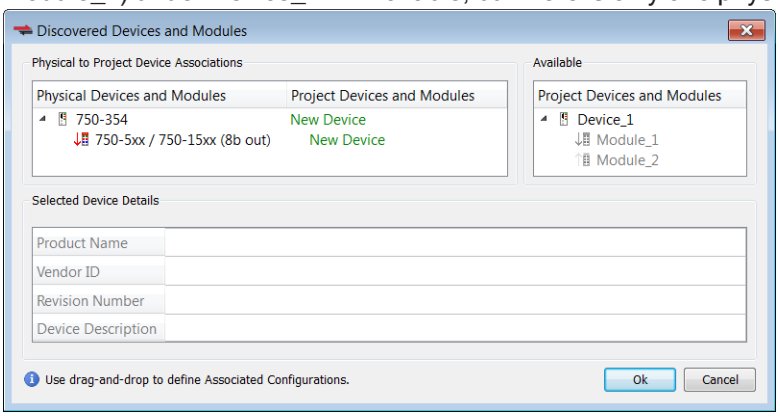
To update the network topology:

1. In the Project Explorer, double-click the EtherCAT node to open its Properties and click the Scan Devices button in the Devices tab.
   Alternatively, right-click on the EtherCAT node in the Project Explorer, and select Scan Devices.

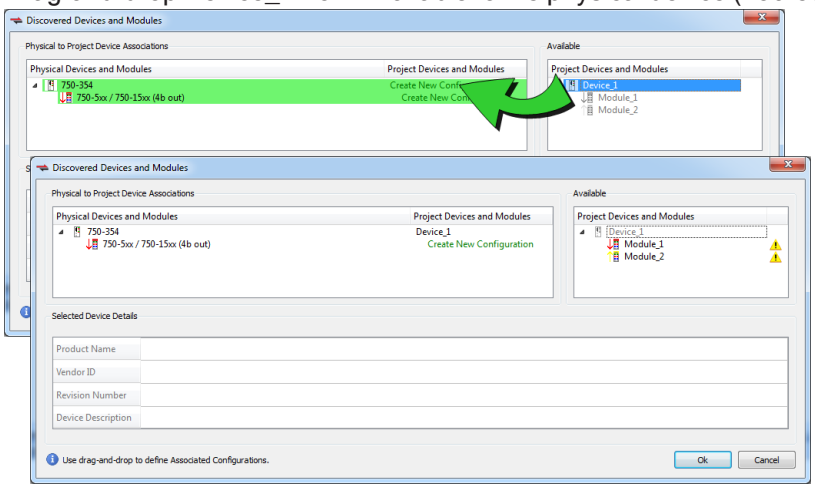


2. The Discovered Devices and Modules dialog box appears after a successful scan.

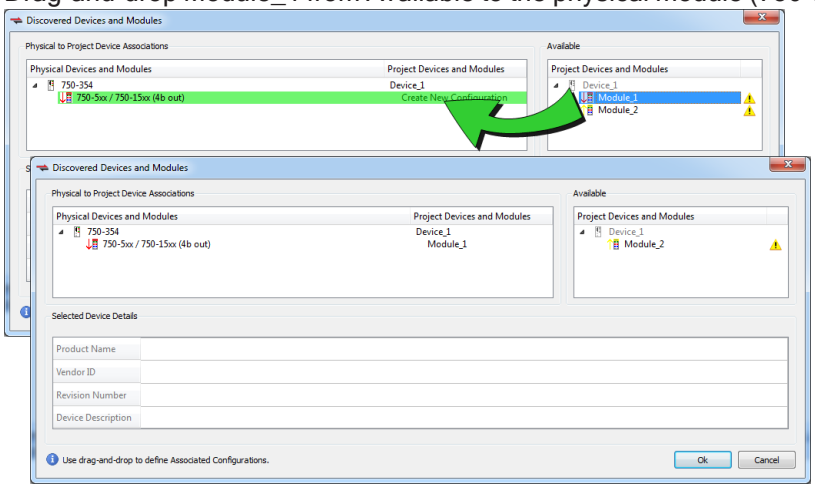
3. Click the Modify button to open the expanded view. Note that there are two modules (Module_1 and Module_2) under Device_1 in Available, but there is only one physical module.



4. To associate the project device and module to the physical device and module:
   1. Drag-and-drop Device_1 from Available to the physical device (750-354).



   2. Drag-and-drop Module_1 from Available to the physical module (750-5xx).



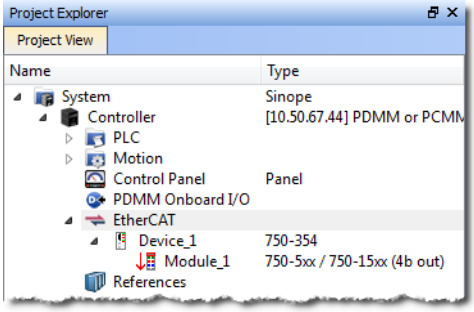
NOTE

Module_2 cannot be associated with a physical module. This is because there are no compatible Physical modules available. Module_2 and any of its user configuration settings will be lost when the OK is clicked.

5. Click OK to confirm the modified configuration. Module_2 is automatically deleted.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The KAS-IDEcreates the EtherCAT node for the device and modules in the Project Explorer to match the physical device and modules. Module_2 is no longer available.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.7 Create a Program

Once the project has been created and the EtherCAT devices have been defined, a program may be created.

TIP

The KAS-IDE programming environment provides language dedicated editors for:

- "Free Form Ladder Diagram (FFLD) Editor" (→ p. 496)
- "Function Block Diagram (FBD) Editor" (→ p. 487)
- "Sequential Function Chart (SFC) Editor" (→ p. 505)
- "Structured Text (ST) / Instruction List (IL) Editor" (→ p. 511)

1. Right-click the **Programs** item in the project tree.
2. Click **New Programs** and select the language to use. (Figure 6-22)

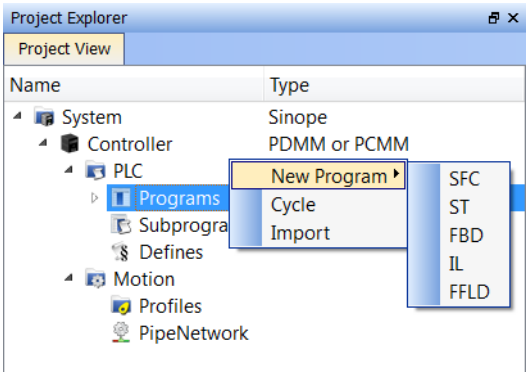


Figure 6-22: Programs > New Program

The **New Program** dialog box opens. (Figure 6-23)

3. Name the program.

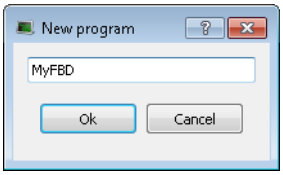


Figure 6-23: New Program dialog box

4. Click **Ok** to create the program and open the language-specific editor. (Figure 6-24)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

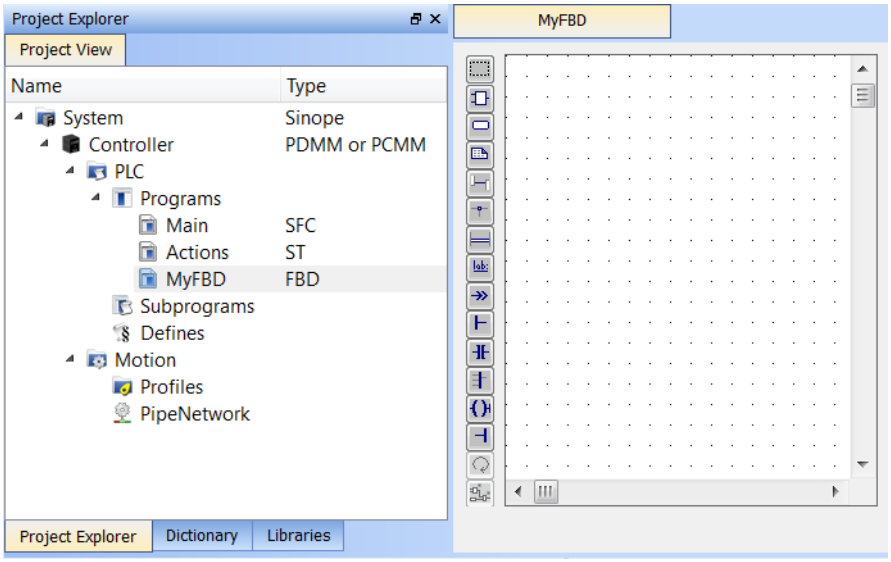


Figure 6-24: Language-specific Editor

The empty program is ready for you to add functions and function blocks.

Other information you may need to complete your program includes:

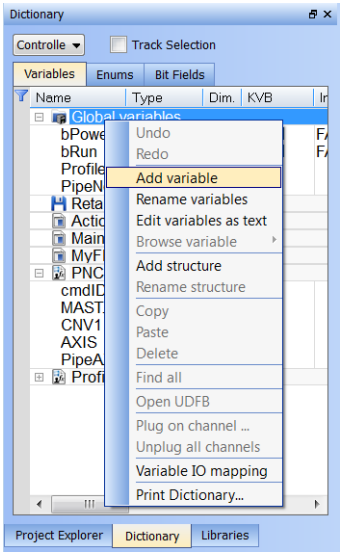
- "Create Variables" (→ p. 183)
- "Create and/or Call Functions or Function Blocks" (→ p. 185)
- "Using Constants" (→ p. 186)
- "Map Input and Output to Variables" (→ p. 186)

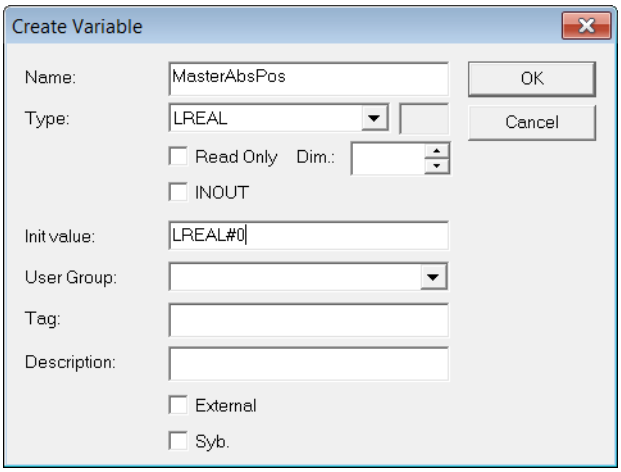
See "IEC 61131-3 Editors" (→ p. 481) in the "Programs: Structure and Syntax" (→ p. 481) section about programming languages.

6.4.7.1 Create Variables

Variables may be created in either of two ways:

- From the "Variables tab" (→ p. 102) of the Dictionary. Clicking **Add Variable** opens the **Create Variable** dialog box.





- From the program Editor. Below we see examples of the FBD (click on the Variable button) and FFLD (double-click on the in or out pins of the function) Editors. Enter a name and click OK to open a dialog box that allows you to define the variable.

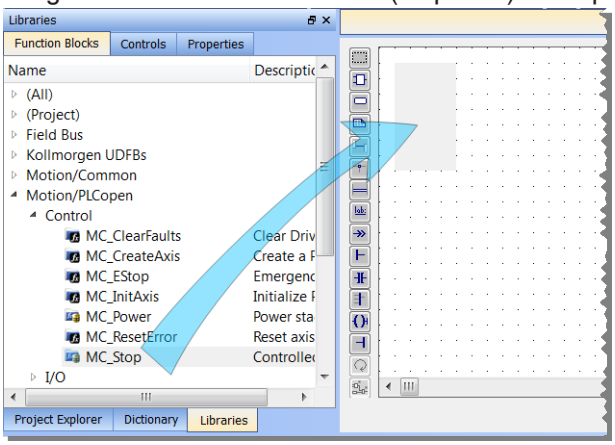
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.7.2 Create and/or Call Functions or Function Blocks

1. Drag an item from the "Libraries" (➔ p. 100) into a program.

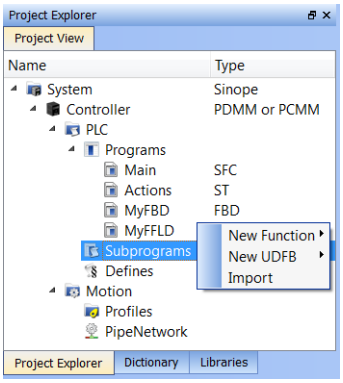


2. Name the new instance of the function or function block.

See "Call Functions or Function Blocks" (➔ p. 523).

6.4.7.2.1 Call a Function or Function Block

1. Declare the function or function block by right-clicking on Subprograms in the Project Tree.



2. Name the function or function block.
3. Define the inputs and outputs, and any private variables.
4. Call the new function or function block.

See Also

- "Create Functions and Function Blocks" (➔ p. 520)
- Differences between Functions and Function Blocks

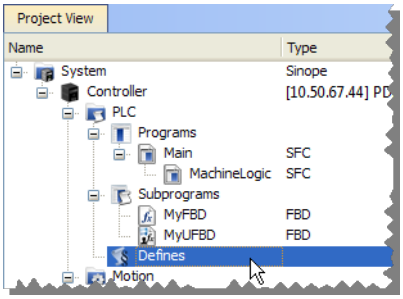
6.4.7.3 Using Constants

Constants, or Defines, exist at Internal, Global, and a Local level.

Level	Scope
"Internal Defines" (→ p. 524)	All the projects present on your machine.
"Global Defines" (→ p. 525)	All the programs in your project. These are user-defined.
"Local Definitions" (→ p. 526)	Only the current program currently open.

How to define constants

1. Double click on the Defines item in the Project Tree.



2. Use the text editor to create the defines using this syntax.

```
#define   Identifier   Equivalence   (* comments *)
```

See "Use the Defines List" (→ p. 523).

6.4.7.4 Map Input and Output to Variables

I/O from the EtherCAT motion bus or the controller is easily mapped to PLC variables through the Project Explorer.

The exact process varies slightly depending on the device.

1. Expand the Controller, and EtherCAT nodes in the Project tree.
2. Double-click on the device or I/O slice entry.
3. Map the variables in any of three ways:
   - Drag and drop variables from the Dictionary.
   - Use the "PLC Variable Creation Wizard" (→ p. 530) for Kollmorgen devices.
   - Directly map / unmap the inputs and outputs using the "PLC Variable Selector dialog box" (→ p. 531).

See "Map Input and Output to Variables" (→ p. 526).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.8 Add Motion

These are the ways to add motion, depending on the motion engine:

- "Design Motion with Pipe Network" (➔ p. 187).
- "Design Motion with PLCopen Axis" (➔ p. 195).
- If you are not sure which engine to use, see "Pipe Network or PLCopen" (➔ p. 324).
- For high-level discussions about motion, see the sections in "Motion Concepts" (➔ p. 320).

6.4.8.1 Design Motion with Pipe Network

This section describes how to create and modify a Pipe Network.

6.4.8.1.1 Create the Pipe Network

1. In the Project Explorer, double-click the **PipeNetwork** button to open the graphical "Pipe Network Editor" (➔ p. 733). (Figure 6-25)

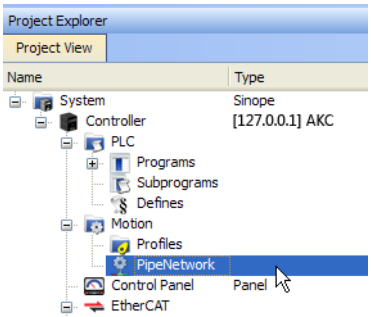


Figure 6-25: Pipe Network - Open Editor

NOTE

If you have created a project from a template (e.g., the standard two-axis template) there is already a Pipe Network in the editor.

2. Right-click on the editor's background and select the **Add Pipeblock** command in the menu. (Figure 6-26)
3. In the drop-down menu, select the type of Pipe Block you want to add.

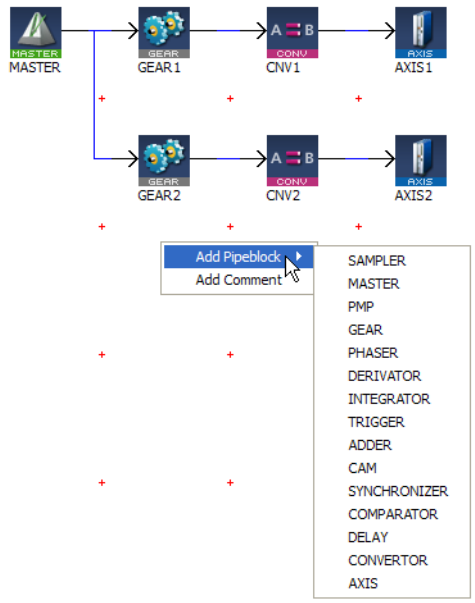


Figure 6-26: Pipe Network - Add Pipeblock

4. To link the newly created Pipe Block, move the arrow to the corresponding Pipe Block with a drag-and-drop operation. (Figure 6-27)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

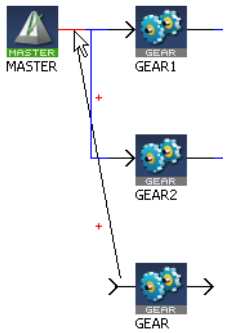


Figure 6-27: Pipe Network- Create a Link

6.4.8.1.2.1 Delete a Pipe Block

There are two methods.

- Right-click on the Pipe Block and select the **Delete** command in the contextual menu.
- Select the Pipe Block by clicking on it and then press the **Delete** key.

6.4.8.1.3.2 Change a Pipe Network Link

1. Select the link so that it becomes Red. (Figure 6-28)

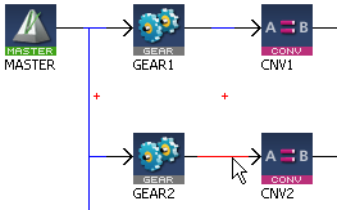


Figure 6-28: Pipe Network- Edit a Link

2. You can either:
   - Right-click and select the **Delete** command if you want to remove the link. (Figure 6-29)

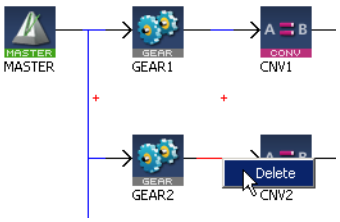


Figure 6-29: Pipe Network- Delete a Link

   - Move the arrow to another Pipe Block with a drag-and-drop operation. (Figure 6-30)

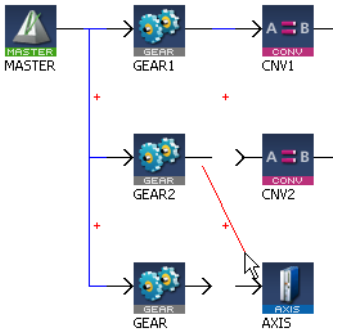


Figure 6-30: Pipe Network- Move a Link

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.8.1.4 Edit Properties of Pipe Blocks

Initial values for Pipe Network blocks are entered in the parameter screen for each block.

To get to the parameter screen, right-click on a Pipe Block and select the **Properties** command in the contextual menu.

- Right-click on the Pipe Block and select the **Properties** command in the menu. (6.4.8.1.4)

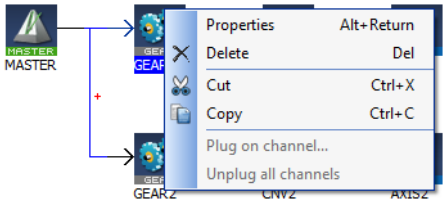


Figure 6-31: Pipe Network - Pipe Block Properties

- The **Properties** window can also be opened by selecting the block and pressing **Alt+Enter**.

You can change the name (or even the type of Pipe Block) in the **General** tab.

The **Parameters** tab gives access to properties related to the type of Pipe Block.

Example

In this example, the selected name AngleAdvance would be used in the PLC application program for this Pipe Network block. (Figure 6-32)

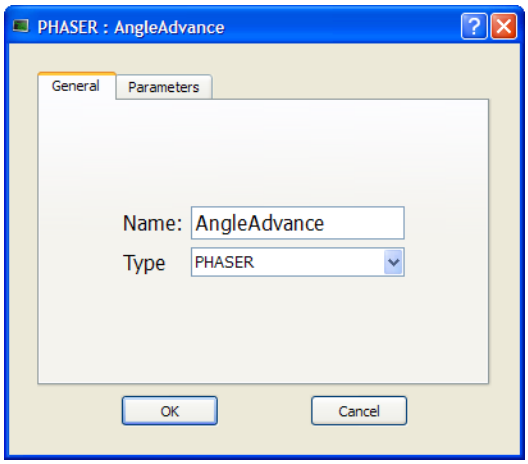


Figure 6-32: Example: Pipe Network name

6.4.8.1.5 Map the Axis to the Drive

To link the axis to an EtherCAT drive, use the "Mapped to Axis" (→ p. 537) procedure.

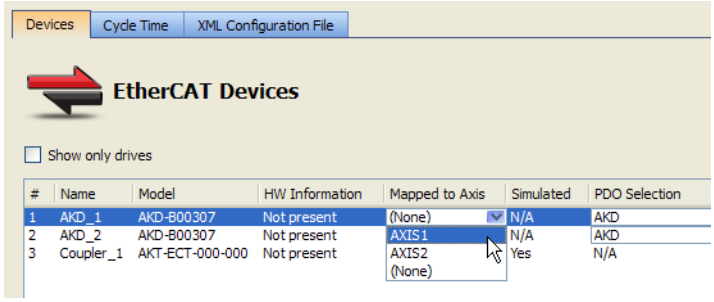


Figure 6-33: Pipe Network - Mapping Axis to Drive

6.4.8.1.6 Add Comments to the Pipe Network

To add a comment:

1. Right-click on the Pipe Network editor's background and select the **Add Comment** command in the menu
2. Right-click on the comment opens the contextual menu to let you edit (**Properties** command) or delete the comment

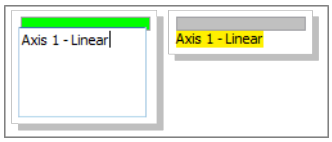


Figure 6-34: Pipe Network comments: editing, and deselected.

Click and drag the gray bar to move the comment.

6.4.8.1.7 Set the Axis Block Position Units

You can set the position units in the parameter screen of the Axis block.

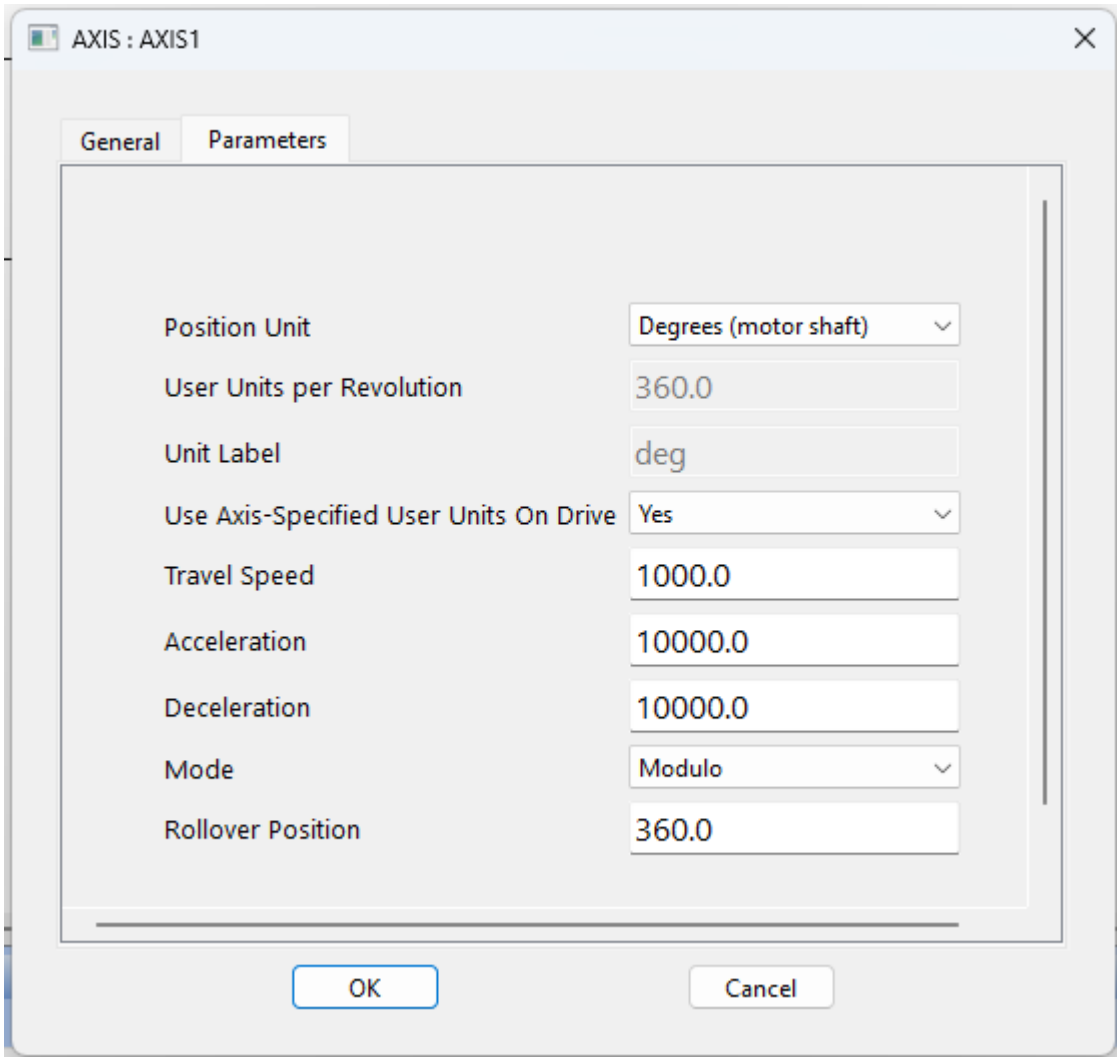


Figure 6-35: Setting Axis Units

Guidelines for Suitable Settings

- The user unit is adapted for the machine.
- The user unit should:
  - Be meaningful for the user.
  - Be used for all related axes.
  - Refer to the position of the final moving object, instead of any intermediate part (e.g., the belt position rather than the position of the axis shaft)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Speed is defined in User Units for position / second (e.g., cm/s).

Acceleration is defined in User Units for position / second^2 (e.g., cm/s2).

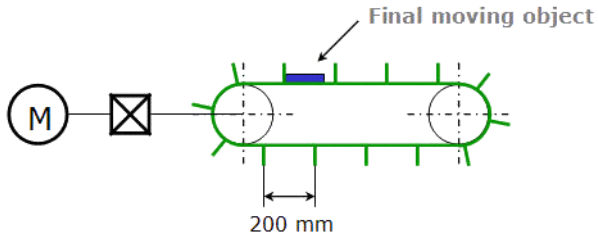


Figure 6-36: Setting the Units - Example

In this example:

- The belt flaps are about 60 degrees away from each other on the wheel, so 6 flaps would pass on a rotation of the axis.
- This corresponds to 1200mm/revolution.
- Because the diagram specifies millimeters, select Custom from the Position Unit drop down box.
  - Enter **1200.0** into the User Units Per Revolution text box.
  - Enter **mm** box into the Unit label text box.

6.4.8.1.8 Show Pipe Network and Profiles-generated Code

You can access the code equivalent to the graphical representation with the contextual menu of the Pipe Network item in the Project Explorer as follows:

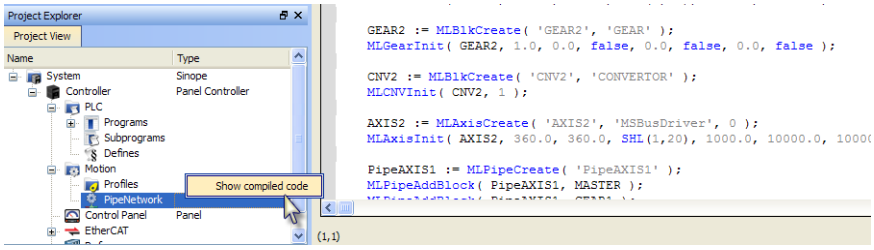


Figure 6-37: Display Source Code of the Pipe Network

The KAS-IDE provides a set of Functions and function blocks for each of the Pipe Blocks. These function blocks allow the logic part of the application to control and interact with the motion engine.

6.4.8.1.9 Pipe Network Functions for the PLC

After creating the Pipe Network, the complete project must be compiled before using the Pipe Network in PLC Programs.

- Compiling creates a list of Functions used in the PLC Program.
- These Functions simplify programming by combining the same function block for all axes in the Pipe Network:

Pipe Network Function	Included Function Blocks (for 2 axis system)
MLPN_ACTIVATE	MLPipeAct(PipeAXIS1); MLPipeAct(PipeAXIS2);
MLPN_CONNECT	MLCNVConnect(CNV1, AXIS1); MLCNVConnect(CNV2, AXIS2);
MLPN_POWER_ON	MLAxisPower(AXIS1); MLAxisPower(AXIS2);

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Pipe Network Function	Included Function Blocks (for 2 axis system)
MLPN_POWER_OFF	MLAxisPowerOFF(AXIS1); MLAxisPowerOFF(AXIS2);
MLPN_DEACTIVATE	MLPipeDeact(PipeAXIS1); MLPipeDeact(PipeAXIS2);

See click here... about constant definitions related to Pipe Network.

TIP

To see how these functions are used, open a project, go to "Project Explorer" (➔ p. 85), right-click on **PipeNetwork** and select the **Show compiled Code** command.

6.4.8.1.10 Initialize and Start a Pipe Network

See Motion State Machine

The Motion State Machine is driven by the IEC 61131-3 application with the help of dedicated function blocks.

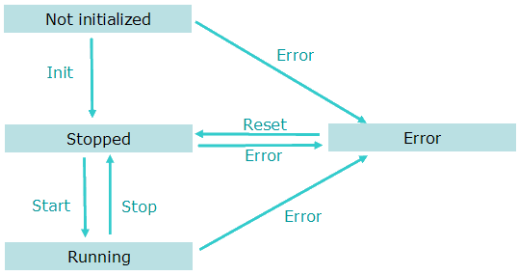


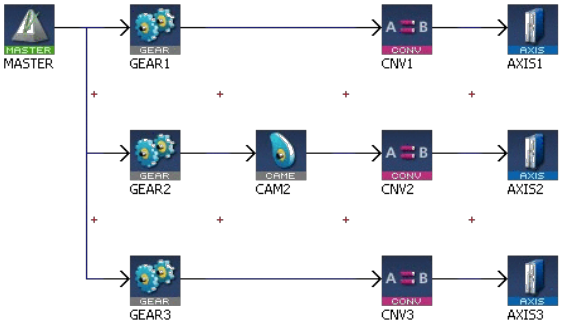
Figure 6-38: Motion State Machine

Each arrow represents a transition from one State to another one.

To start a Pipe Network in an IEC 61131-3 application program, these steps must be performed with their respective functions:

Step	ML Function Block	Description
1. Motion Init	MLMotionInit	1. Initialization of the Motion is done with this dedicated function. 2. Set the Motion engine update rate. 3. Wait for acknowledgment: MLMotionStatus() = MLSTATUS_INITIALISED to continue program operation.
2. Create Cam Profiles	Profiles MLPR_CREATE_PROFILES	Create Cam Profiles from cam files.
3. Create Pipe Network	Pipe Network MLPN_CREATE_OBJECTS	
4. Motion Start	MLMotionStart	1. Starts the motion engine, motion bus driver, and initializes the EtherCAT network to operational mode. 2. Waits for acknowledgment: MLMotionStatus() = MLSTATUS_RUNNING to continue program operation.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Step	ML Function Block	Description
5. Power on all axes	Pipe Network MLPN_POWER_ON	
6. Activate the pipes	Pipe Network MLPN_ACTIVATE	
7. Connect the axes to the pipes	Pipe Network MLPN_CONNECT	Example: In this Pipe Network, the function connects the Converter blocks (CNV1, CNV2 and CNV3) to the Axis blocks: 

6.4.8.1.11 Pipe Network Interaction with PLC Programs

Each Pipe Block is supported by several ML function blocks in the function block Library.

As soon as you add a Pipe Block, it is included in the Variable Editor.

1. Add the FB into your program (see the "Programs: Structure and Syntax" (➔ p. 481) procedures).
2. Select the variable to update. (Figure 6-39)

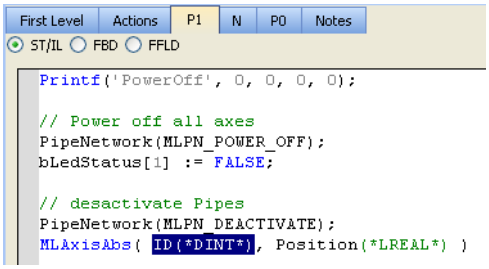


Figure 6-39: Selected variable

3. Press **CTRL+SPACE** to open the Variable Editor.
4. Expand the Pipe Network node and select the name of the Pipe Block in the list. (Figure 6-40)
   All the Pipe Blocks created in the Pipe Network are listed.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

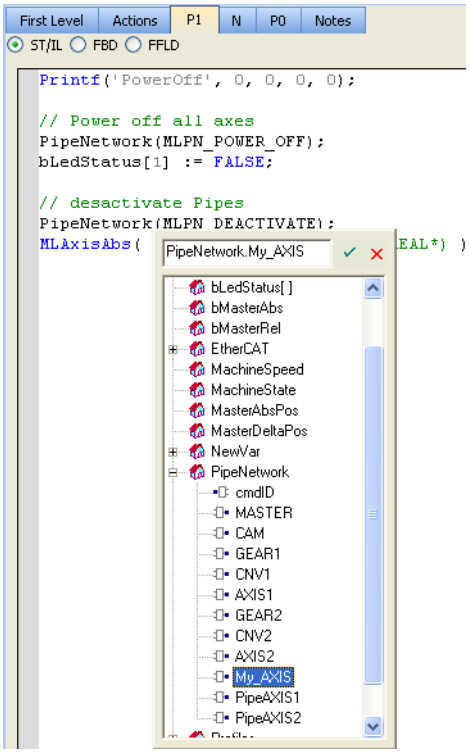


Figure 6-40: Selected Pipe Block

The ST instruction is updated. (Figure 6-41)

```
// desactivate Pipes
PipeNetwork(MLPN_DEACTIVATE);
MLAxisAbs( PipeNetwork.My_AXIS, Position(*LREAL*) )
```

Figure 6-41: Updated instruction

NOTE

When you add a new Block in the Pipe Network, you first need to compile your project to make the block visible in the list of items.

5. Click the  icon to update the code.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.8.2 Design Motion with PLCopen Axis

This section describes how to create and modify a PLCopen Axis.

6.4.8.2.1 Create a PLCopen Axis

1. In the Project Explorer, under "PLCopen" (→ p. 92), right-click the "Axes" (→ p. 92) folder.
2. Select the **New Axis** command in the menu.
   The **Create PLCopen Axis** dialog box opens. (Figure 6-42)

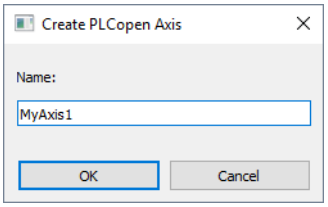


Figure 6-42: Create PLCopen Axis dialog box

3. In the Name text box, enter the new axis name.
4. Click **OK** to save the changes or selections and close the dialog box.
5. Right-click on the new axis.
6. Complete the axis properties.
   See "Change a PLCopen Axis" (→ p. 196).

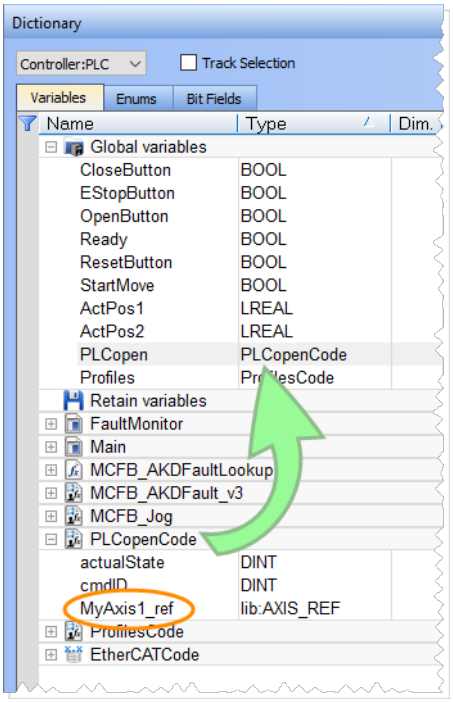


Figure 6-43: Dictionary - Variables tab

7. Compile the project to generate the code for the PLCopen axis reference(s).
   Each axis appears in the variable type PLCopenCode.
   This code is used by the PLCopen Global variable.
   The PLCopen.XXXXXX_ref (e.g., MyAxis1) is an instance of an that can be used in PLC programs. (Figure 6-44 and Figure 6-45)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

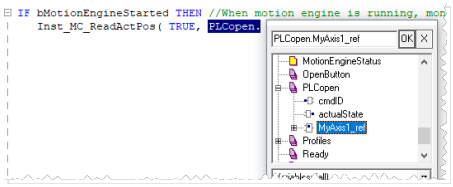


Figure 6-44: Example: PLCopen.XXXXXX_ref (e.g., MyAxis1)

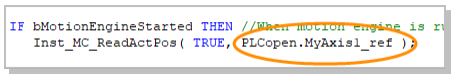


Figure 6-45: Example: PLCopen.XXXXXX_ref (e.g., MyAxis1)

TIP

The PLCopen axis Name and Number (created in the PLCopen Axis Data dialog) are automatically applied to the PLCopen instance under Global variables in the Dictionary when the project is compiled.

6.4.8.2.2 Change a PLCopen Axis

1. In the "Project Explorer" (➔ p. 85), double-click a PLCopen axis.
2. Right-click the PLCopen axis in the project manager.
3. Select **Properties** in the menu. (Figure 6-46)

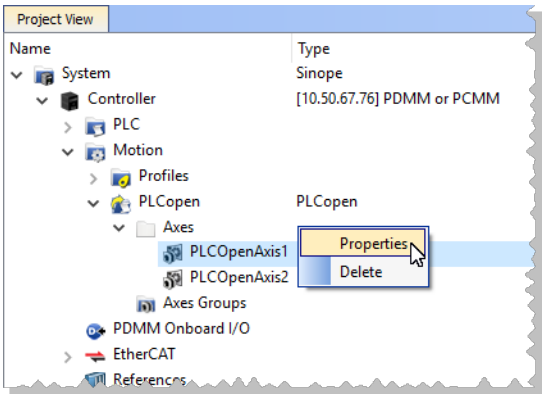


Figure 6-46: PLCopen Axis right-click menu - Properties

The PLCopen Axis Data dialog opens. (Figure 6-47)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

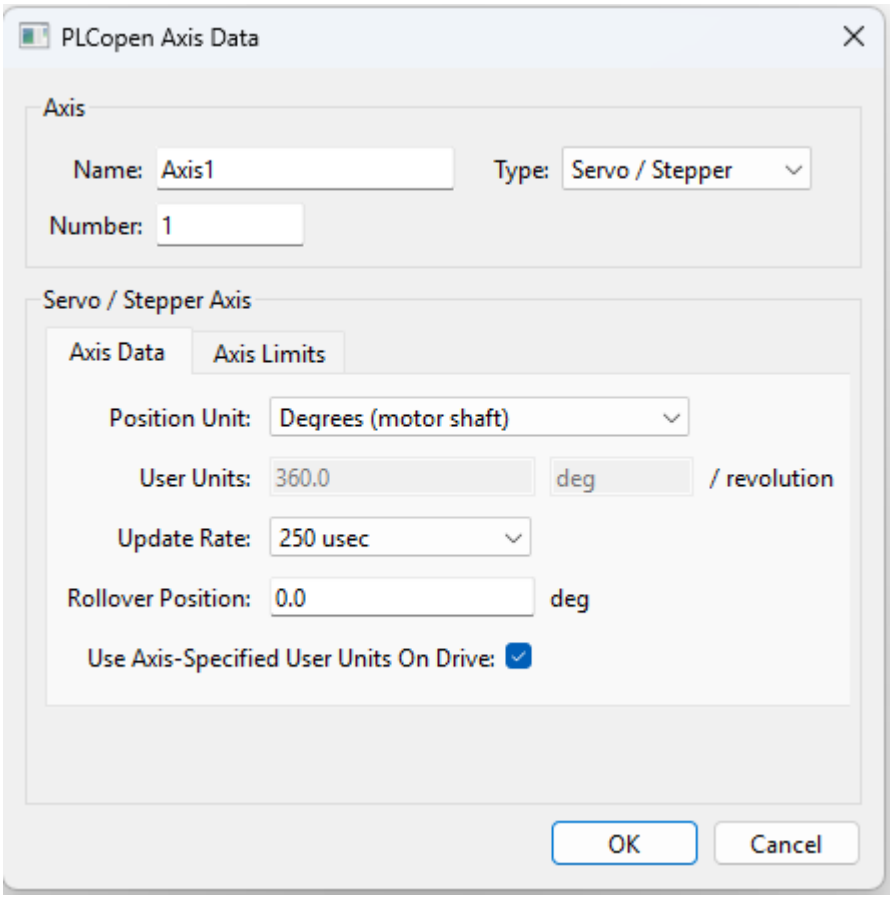
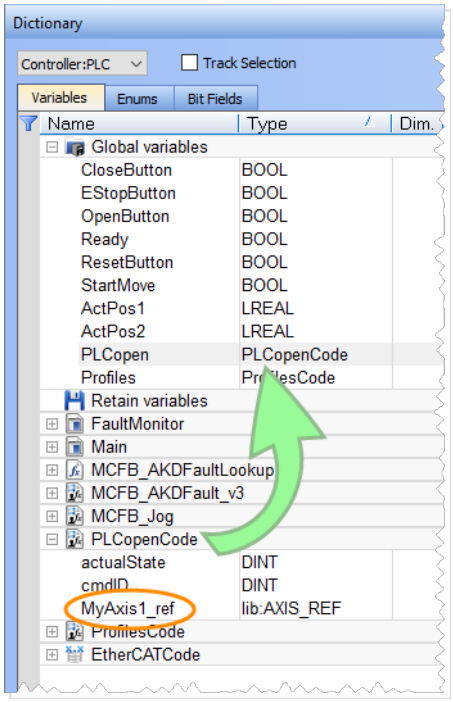


Figure 6-47: PLCopen Axis Data dialog

6.4.8.2.3.1 Axis Name and Number

When the project is compiled, the PLCopen axis Name and Number (from the PLCopen Axis Data dialog - Figure 6-47) are automatically applied to the PLCopen instance of an AXIS_REF structure under Global variables in the Dictionary. (Figure 6-48)



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Figure 6-48: Dictionary - Variables tab

The PLCopen.XXXXXX_ref (e.g., MyAxis1) instance can be used in the PLC programs. (Figure 6-49 and Figure 6-50)

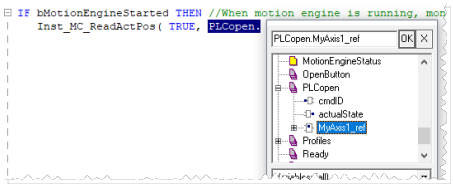


Figure 6-49: Example: PLCopen.XXXXXX_ref (e.g., MyAxis1)

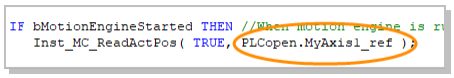


Figure 6-50: Example: PLCopen.XXXXXX_ref (e.g., MyAxis1)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.8.2.4.2 Common Axis Parameters

These are the types of axes available:

- Servo/Stepper
- Digitizing
- Virtual

All types have common parameters related to an axis.

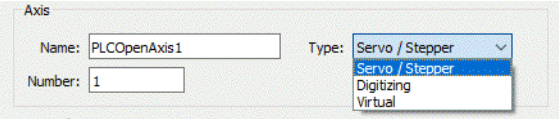


Figure 6-51: PLCopen Axis Parameters

Parameter	Description
Name	This is the user-defined name of the axis. • The name can consist of 1-16 alphanumeric characters. • Spaces are not allowed in the name. • The Axis Name identifies the axis displayed in the Project Tree and in the Simulator.
Type	With a **Servo/Stepper** axis: • Commands are sent to the axis and feedback is read from the axis. • After selecting this option, the "Axis Data Parameters" (→ p. 202) and "Axis Limits Parameters" (→ p. 203) must be defined. A **Digitizing** axis contains position feedback data only, no motion can be commanded. • Digitizing axes can use extra or secondary feedback from the Drive or from a third-party EtherCAT feedback device. • Selecting a third-party EtherCAT feedback device require entered EtherCAT Bus data. • See "Digitizing Axis Parameters" (→ p. 200). • This varies depending upon the EtherCAT devices. • See "Create a Digitizing Axis" (→ p. 851) for specifics on different methods. A **Virtual** axis is an axis with no feedback or drive hardware. • The feedback for a virtual axis is automatically generated from the command position. • There is no limit to the number of virtual axes that may be used in an application.
Number	The axis number (1-256) specifies the axis for PLCopen motion function blocks.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.8.2.5.3 Digitizing Axis Parameters

When creating a digitizing axis, these items must be set:

- "Common Axis Parameters" (➔ p. 199) (name and number)
- EtherCAT Bus parameters
- "Axis Data Parameters" (➔ p. 202)

The Digitizing axis type has additional parameters to define for third-party EtherCAT feedback devices.

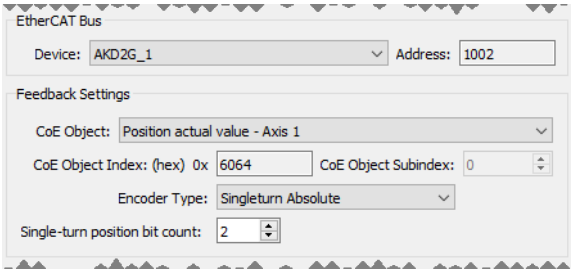
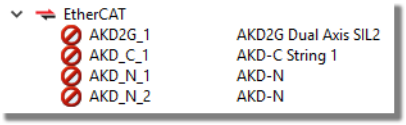


Figure 6-52: PLCopen Axis - Bus Parameters

The EtherCATdevice parameters are:

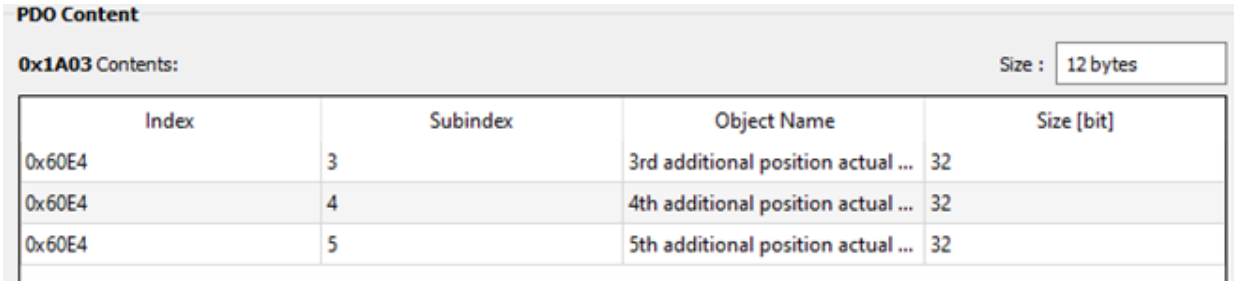
Parameter	Description
Device	Select the device that provides the digitizing axis feedback. The list is populated by the devices defined in the EtherCAT section of the Project View.  **Figure 6-53:** Example 1: Project View  **Figure 6-54:** Example 2: Project View
Address	The digitizing **Address** is the EtherCAT address of the device in the network. • It is the 4-digit node address of the servo drive on the bus. • This address is required to assign a digitizing axis to an EtherCAT node that already has a servo axis assignment. • The address for the EtherCAT slaves are defined as: • First device: 1001 • Second: 1002 • Third: 1003, etc.
CoE Object	Select the item that will be used to read the feedback.
CoE Object Index	The CoE object index of the feedback device. This is a hexadecimal value.
CoE Object Sub-Index	The sub-index of the feedback device. This is a decimal value,
Encoder Type	Select **Singleturn Absolute** or **Multiturn Absolute**.
Single-turn position bit count	This is the single turn position bit count of the feedback device. The valid bit count range is from 1 to 31.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Example

AKD2G: Digitizing axes mapped to an AKD2G device are particularly easy to work with.

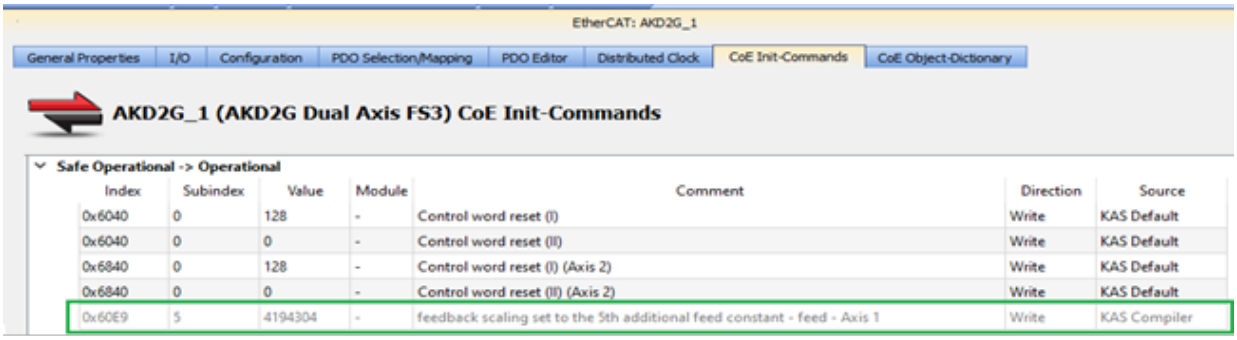
- The PDO 0x1A03 is populated with available feedbacks.



Index	Subindex	Object Name	Size [bit]
0x60E4	3	3rd additional position actual ...	32
0x60E4	4	4th additional position actual ...	32
0x60E4	5	5th additional position actual ...	32

Figure 6-55: Example: PDO 0x1A03

- When a feedback object 0x60E4:1 to 0x60E4:5 is selected, the:
  - **Encoder Type** selection is hidden.
  - **Encoder Resolution** is set to 31-bits.
  - **Feedback Units** is initialized to 2^{31}.
- The **Feedback Units** is automatically adjusted to $2^{(\text{Encoder Resolution})}$ when the **Encoder Resolution** is changed.
- Feedback units are configured on the AKD2G device based on the resolution specified for the digitizing axis.
  This is done using CoE Init-Commands during project compilation.



See "Digitizing Axis - PLCopen Setup" (➔ p. 855).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.8.2.6.4 Axis Data Parameters

In the **PLCopen Axis Data** dialog box (Figure 6-56):

- If a Servo/Stepper axis is selected, two tabs are available:
  - Axis Data
  - Axis Limits
- If a Digitizing axis is selected, only the Axis Data tab is available.
- By default, new axes are defined using degrees as the measurement unit.

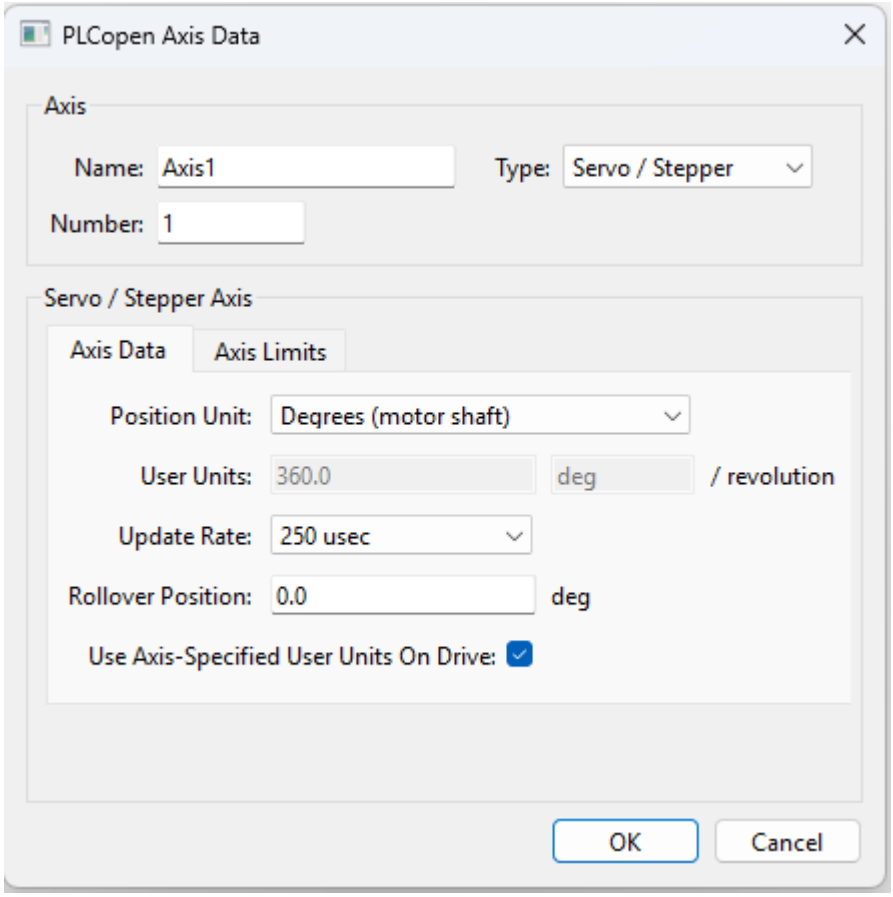


Figure 6-56: Servo/Stepper Axis - Axis Data tab

The Servo/Stepper Axis - Axis Data parameters are:

Parameter	Description
Position Unit	• This field sets the units used by the drive axis. • When the axis is defined as an AKD2G the controller applies the user units value and label to the drive. • These user units are applied for both Op-mode (running application) and Pre-Op mode (online config) when the EtherCAT network is started up. • Counts 16-bit • Degree • Radian • Custom - Selecting Custom allows you to specify the User Units value.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Parameter	Description
User Units / Revolution	• The value used to define an axis position or distance for an axis. • The default values are: • Counts: 65,536 / revolution • Radian: 6.28318530717 (or 2π) / revolution • Degree: 360 / revolution • When the Unit is set to Custom you may enter your own value and units name, such as `gradians`, `centimeters`, or `inches`.
Update Rate	The rate at which the axis's feedback is read and a new command position is generated. The choices are: • 250 µsec • 500 µsec • 1 msec • 2 msec • 4 msec • 8 msec This rate can be slower or equal to the EtherCAT Cycle Time. • The EtherCAT Cycle Time specifies the rate at which data is transferred between the control and the drives. • The axis Update Rate is the rate at which the PLCopenCode reads the feedback, runs its interpolation, and generates a new command position. • By allowing some axes to run at a slower rate and staggering the updates on which these axes are interpolated, more axes and/or quicker execution times can be achieved since every axis does not have to be interpolated every update. • If you select an axis Update Rate which is faster than the EtherCAT Cycle Time, the axis is set to run at the EtherCAT Cycle Time. • See "EtherCAT Master Settings tab" (→ p. 539).
Rollover Position	• The value at which the axis position rollovers to zero. Rollover Position is specified in User Units. Example • If the rollover position is 100: • The axis position counts up from 0 to 999 and then rollover back to 0. • In the reverse direction, the axis position counts down to 0 and then rollover to 999. • If Rollover Position is 0, no rollover occurs. • Axis positions become negative values when counting down below 0.
Use Axis-specified User Units on Drive	• When this check box is selected, and if the Axis is mapped to a AKD2G or a Kollmorgen Essentials Drive, the KAS compiler automatically generates the required CoE Init commands to set the drive User Units same as the Pipe Network Axis. • By default, the check box is selected.

6.4.8.2.7.5 Axis Limits Parameters

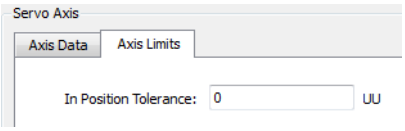


Figure 6-57: Servo Axis - Axis Limits

The Servo Axis - Axis Limits parameters are:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Parameter	Description
In Position Tolerance	• The maximum distance between the axis's actual position and its commanded endpoint for the axis to be considered in position. • The In-Position Tolerance is specified in User Units.

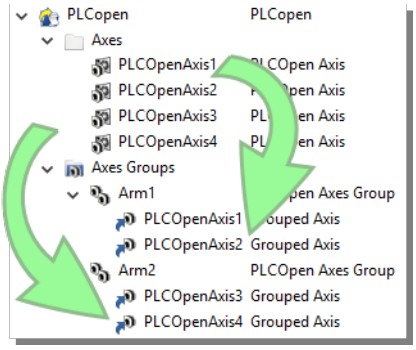
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.8.2.8 Create PLCopen Axes Groups

If the project utilizes coordinated motion and/or robotic transformations, you need to create one or more Axes Groups in the Project Explorer.

Axes Groups are collections of related axes defined by the data type AXIS_GROUP_REF.

- They are typically used by coordinated motion and function blocks such as MC_SetKinTra.
- Right-click in this section to create a new Axes Group.
- Groups are populated by dragging an existing axis from the above section into the group.



Right-click a group to **Rename** or **Delete** the group.

See Also

- MC_KIN_REF Structure
- MC_SetKinTra
- "What are Axes Groups?" (→ p. 827)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.8.2.9 Initialize and Start PLCopen

Click to see a Motion State Machine example

The Motion State Machine is driven by the IEC 61131-3 application with the help of dedicated function blocks.

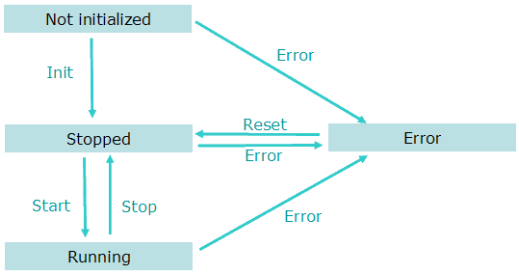


Figure 6-58: Motion State Machine

Each arrow represents a transition from one State to another one.

Complete these steps and functions to start the PLCopen engine in an IEC 61131-3 application program:

Step	Function Calls	Description
Motion Init	MLMotionInit	• Initialization of the Motion is done with this dedicated function. • Set the Motion engine update rate. • Wait for acknowledgment: MLMotionStatus() = MLSTATUS_ INITIALISED to continue program operation.
Create Cam Profiles	Profiles	• Create Cam Profiles from cam files. • See "Use the Defines List" (→ p. 523).
Create/Initialize PLCopen Axes	PLCopen (0);	• Create and initialize PLCopen axes from the PLCopenCode attached to the PLCopennode in the Project tree. • Right-click the **PLCopen** node and select **Show Compiled Code** to see the PLCopenCode.
Motion Start	MLMotionStart	• Starts the motion engine, motion bus driver, and initializes EtherCAT network to operational mode. • Waits for acknowledgment: MLMotionStatus() = MLSTATUS_ RUNNING to continue program operation

6.4.8.3 Design Coordinated Motion

These topics show how to create a coordinated motion application, including adding coordinated motion to existing applications.

- "Blending Between Moves" (→ p. 822)
- "Blending with Transitions" (→ p. 823)
- "Create a Linear or Circular Coordinated Motion Application" (→ p. 824)
- "Differences between MC_GrpHalt and MC_GrpStop" (→ p. 833)
- "Handling Axis Errors" (→ p. 838)
- "Transition Between Moves" (→ p. 840)

See Also

- "Coordinated Motion" (→ p. 820)
- Coordinated Motion Function Blocks

6.4.8.3.1 Create a Linear or Circular Coordinated Motion Application

A Coordinated Motion application can be created in one of these ways:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Use a Coordinated Motion template to create a new application.
  - Two Coordinated Motion templates are currently available.
    - The first template controls two PLCopen axes in coordinated motion.
    - The second template controls two PLCopen axes in coordinated motion plus a third independent Pipe Network axis.
- Modify an existing application to included coordinated motion functions.
  - When modifying an existing application, axes need to be grouped to define the axes that will be active when performing coordinated motion on that group.
  - See "What are Axes Groups?" (➔ p. 827).

NOTE

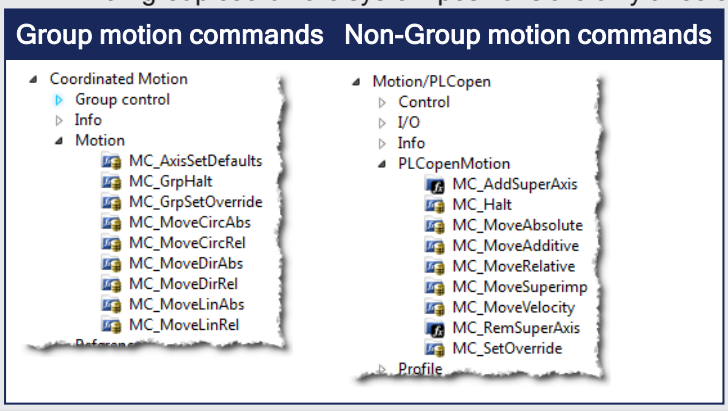
Coordinated motion can only be performed with PLCopen axes.
Pipe Network axes do not support this feature.
Pipe Network axes can be moved independently from coordinated motion groups.
Any synchronization between coordinated motion and Pipe Network axes must be performed by the PLC application.

6.4.9 Groups

- Related axes are grouped in an axes group.
- Coordinated motion is then performed on an axes group.
- See "What are Axes Groups?" (➔ p. 827).

TIP

Remember these vital concepts when making interpolated motion.
- Interpolated motion requires creating a motion group that results in a second group coordinate system.
- Group coordinate system positions are only affected by group motion.
- Non-group coordinate system positions are only affected by non-group motion.



NOTE

Axes that become part of a motion group are first homed using non -group function blocks to establish a home or starting position for the group motion.

6.4.9.0.1.1 Function Block Calls

These function blocks should be called before executing coordinated motion.

1. Call MLMotionInit (BasePeriod) to initialize the motion engine.
   Base period is specified in microseconds.

```
MLMotionInit(1000.0);   // 1000 µSec -> 1 mSec
```

2. Call MC_CreateAxesGrp (Enable, GroupName, UpdateRate, MaxNumberOfAxes, AxesGroupRef) to create a Coordinated Motion Axes Group.

   NOTE

   MC_CreateAxesGrp must be called between MLMotionInit() and MLMotionStart().

   ```
   Inst_MC_CreateAxesGrp(TRUE, 'GROUP1', 6, 2, Group1_ref);
   ```

   In this example, the axes group name is **GROUP1**, the update rate is 1 mSec (specified by '6') and the maximum number of axes that can be added to the group is 2.
   The group reference variable **Group1_ref** is used in future coordinated motion function block calls to reference this newly created group.

3. Call MC_InitAxesGrp (Enable, AxesGroup, VelLimit, AccLimit, DecLimit, JerkLimit) to initialize the path limits for velocity, acceleration, deceleration, and jerk.

   ```
   Inst_MC_InitAxesGrp(TRUE, Group1_ref, 100.0, 300.0, 300.0, 1000.0);
   ```

   In this example, the kinematic limits for axes group **Group1_ref** is set.
   The velocity limit is set to 100.0 user units/second.
   Acceleration and deceleration limits are set to 300.0 User units / sec^2.
   Jerk is set to 1000.0 User units / sec^3.

4. Call MC_CreatePLCAxis (AxisName, BusInterface, BusAddress, AxisNumber, AxisType, UserUnits, FeedbackUnits, Rollover, UpdateRate) to create a Coordinated Motion Axis.
   This function must be called for each Coordinated Motion Axis wanted in the application.

   NOTE

   MC_CreatePLCAxis needs to be called between MLMotionInit() and MLMotionStart().

   ```
   Inst_MC_CreateAxis(TRUE, 'CoordAxis1', 'EtherCATDriver', 1001, CoordAxis1_
   AxisNum, 0, 360, 1048576, 0, 6);
   Inst_MC_CreateAxis(TRUE, 'CoordAxis2', 'EtherCATDriver', 1002, CoordAxis2_
   AxisNum, 0, 360, 1048576, 0, 6);
   ```

   In this example:

   - Two axes are created and are named **CoordAxis1** and **CoordAxis2**.
   - The bus interface for both is **EtherCATDriver**.
   - The address of the drive on the bus is 1001 and 1002.
   - The axis numbers are set with variables CoordAxis1_AxisNum and CoordAxis2_AxisNum which is set to an integer value between 1 and 256.
     - Each axis number is unique.
   - The axis type for both, 0, indicates a servo axis.
   - The user units are 360, which is the user unit portion of the user unit/feedback ratio.
   - The feedback units are 1048576.
     - This is the feedback portion of the user unit/feedback ratio.
   - The rollover position for both, 0, indicates no rollover.
   - The update rate for both, 6, indicates a 1mSec update rate.

5. Call MLMotionStart () to start the Motion and the motion bus driver.
   This also initializes the EtherCAT network to operational mode.

   ```
   MLMotionStart();
   ```

6. Call MC_AddAxisToGrp (Execute, AxesGroup, Axis, IdentInGroup) for each axis to add to the group.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
Inst_MC_AddAxisToGrp(TRUE, Group1_ref, CoordAxis1_ref, 0);
Inst_MC_AddAxisToGrp(TRUE, Group1_ref, CoordAxis2_ref, 1);
```

In this example, we are adding two axes, CoordAxis1 and CoordAxis2, to the group referenced by **Group1_ref**.
The axes are stored in the IdentInGroup positions 0 and 1.
When the group was created, it was specified that no more than 2 axes will be part of this group.
Therefore, valid IdentInGroup locations are 0 and 1.

7. Call MC_Power (Enable, Axis, EnablePositive, EnableNegative, BufferMode) for each Coordinated Motion Axis to enable the drive and close the servo loop.

```
Inst_MC_Power1(TRUE, CoordAxis1_ref, TRUE, TRUE, 0);
Inst_MC_Power2(TRUE, CoordAxis2_ref, TRUE, TRUE, 0);
```

In this example, drives CoordAxis1_ref and CoordAxis2_ref are enabled and the position loop is closed.

8. Call MC_GrpEnable (Execute, AxesGroup) to change the state of the Coordinated Motion Axis Group from GroupDisabled to GroupStandby and allow motion to be performed on the group.

```
Inst_MC_GrpEnable(TRUE, Group1_ref);
```

In this example, **Group1_ref** state is changed from GroupDisabled to GroupStandby.
The group must be in GroupStandby to perform the motion.

9. In these examples, we want to set the current location of the axes in the group to 0, 0.
This is done by calling MC_GrpSetPos (Execute, AxesGroup, Position[], Relative, CoordSystem, BufferMode).

```
PosAbs[1]:= 0;
PosAbs[2]:= 0;
Inst_MC_GrpSetPos(TRUE, Group1_ref, PosAbs, 0, MC_COORDINATE_SYSTEM_ACS,
0);
```

In this example, the axis positions of **Group1_ref** is set to 0, 0.
PosAbs specifies the position for each axis in the group.
Relative input, 0, uses PosAbs to set the absolute position.
The coordinate system is set to ACS.

NOTE

No motion is performed when this function block is executed.

10. **Optional:** To Add more axes to the group, change the code this way:
    - In Step 2: Update the MaxNumberOfAxes input argument so that the group can handle the desired number of axes.
    - In Step 4: Create the additional axes to add to the group.
    - In Step 6: Add the additional axes to the group.
    - In Step 7: Power on the additional axes.
    - In Step 9: Increase the size of the PosAbs array so it matches the number you used in step 2. Set the position of the additional axes to zero.

After the above function calls have been made, we can start coordinated motion moves.

See Also

- "Perform a Linear Move" (→ p. 831)
- "Perform a Circular Move" (→ p. 827)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.4.9.0.2.2 What are Axes Groups?

Related axes are grouped in an Axes Group to support interpolation. Axes Groups are accessed using the type AXES_GROUP_REF.

This image shows the relationships between the different CSs and groups.

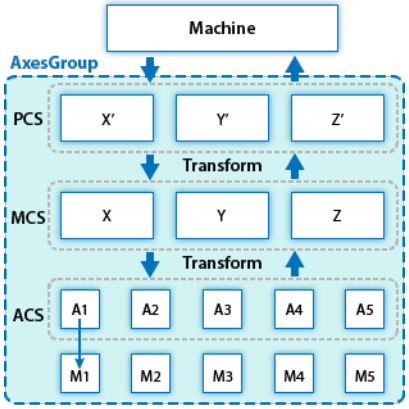


Figure 6-59: Overview of AxesGroup

The Axes Group, shown in blue, provides the interface to the user of the group of axes.

To access the relevant coordinate system, the relevant function blocks have an input Coordinate System which supports the three levels ACS, MCS, and PCS.

Parameters in the AxesGroupRef can include remaining time and remaining distance before target position (or velocity or equal) is reached.

6.4.9.0.3.3 Perform a Circular Move

Circular moves can be programmed using absolute or relative positions using the following function blocks:

- MC_MoveCircAbs which commands interpolated circular movement on an axes group to the specified absolute positions.
- MC_MoveCircRel which commands interpolated circular movement on an axes group to the specified relative positions.

Prior to performing any coordinated moves, some setup is needed.

- See "Create a Linear or Circular Coordinated Motion Application" (→ p. 824).
- Once these steps have been performed, a circular move can be performed.

In these examples, two circular moves are performed:

- "Absolute Circular Move" (→ p. 210)
- "Relative Circular Move" (→ p. 211)

6.5 Absolute Circular Move

In this move, an absolute circular move goes from (0, 0) to (90, 90).

- CircMode specifies that the aux point (0, 180) is crossed during the paths start to end.
- Call MC_MoveCircAbs.

```
PosCircAuxAbs[0] :=   0;   // A point on the circle that is crossed on
the
PosCircAuxAbs[1] := 180;   // path from start to end point.
PosCircEndAbs[0] :=  90;   // Absolute end point.
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
PosCircEndAbs[1] :=  90;

Inst_MC_MoveCircAbs(TRUE, Group1_ref, CIRC_MODE_BORDER, PosCircAuxAbs,
PosCircEndAbs, CIRC_PATH_CLKWISE, MaxVel, MaxAcc, MaxDec, 0, CS_ACS, BM_
BUFFERED, TM_NONE, TransParam);
```

In the example, a circular move is performed on axis group **Group1_ref**.

- CircMode is defined as CIRC_MODE_BORDER.
  - This mode indicates that the AuxPoint array input indicates a point on the circle which is crossed on the path from the starting point to the end point.
  - See "Circular Moves Diagrams" (➔ p. 829) about CircMode movement options.
- The AuxPoint array, **PosCircAuxAbs**, defines an absolute point on the circle which is crossed on the path from the starting point to the end point.
  - The contents of this array are determined by the CircMode variable CIRC_MODE_BORDER.
- The EndPoint array, **PosCircEndAbs**, contains the absolute end point for each axis in the group.
  - The absolute end point of the axis stored in position 0 (IdentInGroup) of the group is 90.0.
  - The absolute end point of the axis stored in position 1 of the group is 90.0.
- PathChoice is only relevant when CircMode is set to CIRC_MODE_CENTER.
  - In this case, this parameter is not used.
- The maximum velocity is specified by variable **MaxVel** and is specified in User units / sec.
- The maximum acceleration and deceleration are specified by variables **MaxAcc** and **MaxDec** and are specified in User units / sec^2.
- The maximum jerk is not supported and can be set to a value of 0.
- The coordinate system is ACS.
- The BufferMode is set to BM_BUFFERED, indicating the move is buffered.
  - See "Buffer Modes" (➔ p. 371).
- The TransitionMode is set to TM_NONE, indicating no transition mode is used.
  - See "Transition Between Moves" (➔ p. 840).
- The TransParam array is required.
  - The contents can be set to 0 because the transition mode is not being used.
  - The TransParam array is a 2-element array containing the corner distance and velocity for the transition.

6.6 Relative Circular Move

This is a relative circular move whose end point is (90, 90) from the end of the first move.

- CircMode specifies that the aux point (0, 90) is the relative center of the circle.
- The BufferMode input is set to **Buffered**; this move waits for the first move to complete before it begins executing.
- Call MC_MoveCircRel.

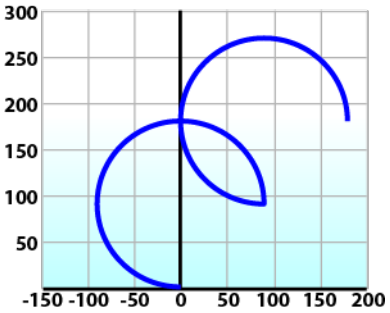
```
PosCircAuxRel[0] :=  0;   // Relative center of the circle.
PosCircAuxRel[1] := 90;
PosCircEndRel[0] := 90;   // Relative end point.
PosCircEndRel[1] := 90;   // Start pt 90,90 + rel 90,90 -> 180,180
absolute end pt

Inst_MC_MoveCircRel(TRUE, Group1_ref, CIRC_MODE_CENTER, PosCircAuxRel,
PosCircEndRel, CIRC_PATH_CLKWISE, MaxVel, MaxAcc, MaxDec, 0, CS_ACS, BM_
BUFFERED, TM_NONE, TransParam);
```

In the example, all the variables have the same meaning as the circular absolute example except:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- CircMode is defined as CIRC_MODE_CENTER.
  - This mode indicates that the AuxPoint array input indicates the center point of the circle.
  - See "Circular Moves Diagrams" (➔ p. 829) about CircMode movement options.
- The AuxPoint array, **PosCircAuxRel**, defines the relative center point of the circle.
  - The contents of this array are determined by the CircMode variable, CIRC_MODE_CENTER.
- The EndPoint array, **PosCircEndRel**, contains the relative end point for each axis in the group.
  - The relative end point of the axis stored in position 0 (IdentInGroup) of the group is 90.0.
  - The relative end point of the axis stored in position 1 of the group will be 90.0.
- PathChoice is relevant when CircMode is set to CIRC_MODE_CENTER.
  - In this case, PathChoice is CIRC_PATH_CLKWISE which specifies the direction of the path.



6.6.0.0.1.1 Perform a Linear Move

Linear moves can be programmed using absolute or relative positions using these function blocks:

- MC_MoveLinAbs commands interpolated linear movement on an axes group to the specified absolute positions.
- MC_MoveLinRel commands interpolated linear movement on an axes group to the specified relative positions.

Prior to performing any coordinated moves, some setup is needed.

- See "Create a Linear or Circular Coordinated Motion Application" (➔ p. 824).
- Once these steps have been performed, a linear move can be performed.

In these examples, two linear moves are performed.

- "Absolute Linear Move" (➔ p. 212)
- "Relative Linear Move" (➔ p. 213)
- "Linear Move With More Than Two Axes" (➔ p. 213)

6.7 Absolute Linear Move

In this move, an absolute linear move goes from (0, 0) to (100, 200).

- Call MC_MoveLinAbs.
  - PositionArray is an array of absolute end positions containing one position for each axis in the group.
  - The input velocity, acceleration, deceleration, and jerk establish the maximum values for the move.

```
PosArrayAbs[0] := 100;
PosArrayAbs[1] := 200;
TransParam[0]  :=   0;

Inst_MC_MoveLinRel(TRUE, Group1_ref, PosArrayAbs, MaxVel, MaxAcc,
MaxDec, 0, CS_ACS, BM_BUFFERED, TM_NONE, TransParam);
```

In the example, a linear move is performed on axis group **Group1_ref**.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- PosArrayAbs contains the absolute end points of the axes in the group.
  - The axis stored in position 0 (IdentInGroup) of the group is moved to 100.0.
  - The axis stored in postiion 1 of the group is moved to 200.0.
- The maximum velocity is specified by variable **MaxVel** and is specified in User units / sec.
- The maximum acceleration and deceleration are specified by variables **MaxAcc** and **MaxDec** and are specified in User units / sec^2.
- The coordinate system is ACS.
- The BufferMode is set to BM_BUFFERED, indicating the move is buffered.
  - See "Buffer Modes" (➔ p. 371).
- The TransitionMode is set to TM_NONE, indicating no transition mode is used.
  - See "Transition Between Moves" (➔ p. 840).
- The TransParam array is required.
  - The contents can be set to 0 because the transition mode is not being used.
  - There has to be one array entry for each axis in the group.

6.8 Relative Linear Move

In this move, a relative linear move goes a distance of (-75, 50) from the end of the first move.

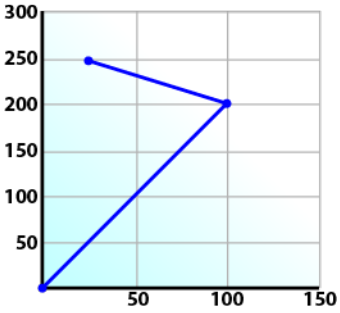
- Call MC_MoveLinRel.
- The Distance input is an array of distances, one distance for each axis in the group.
- The input velocity, acceleration, deceleration, and jerk establish the maximum values for the move.

```
DistArrayRel[0] := -75.0;   // Start pt 100 - rel 75 ->  25 absolute end
pt
DistArrayRel[1] :=  50.0;   // Start pt 200 + rel 50 -> 250 absolute end
pt
TransParam[0]   :=   0;

Inst_MC_MoveLinRel(TRUE, Group1_ref, DistArrayRel, MaxVel, MaxAcc,
MaxDec, 0, CS_ACS, BM_BUFFERED, TM_NONE, TransParam);
```

In the example, all the variables have the same meaning as the absolute linear example except DistArrayRel.

- DistArrayRel contains the relative distance to move for each axis in the group.
- The axis stored in position 0 (IdentInGroup) of the group is moved a distance of -75.0.
- The axis stored in position 1 of the group is moved a distance of 50.0.



6.9 Linear Move With More Than Two Axes

NOTE

The dimensionality of the move is determined by the number of axes mapped to the group.
This implies that a group which could hold a maximum of three or more axes does two dimensional moves if it only has two valid axes mapped to it.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

To perform higher dimensional moves, additional axes must be added to the group.

The procedure for this are in "Create a Linear or Circular Coordinated Motion Application" (→ p. 824).

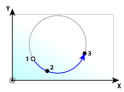
After the additional axes are added, perform these steps.

1. From the Dictionary, update the array size of the variable being passed (PosArrayAbs and DistArrayRel in the "Absolute Linear Move" (→ p. 212) and "Relative Linear Move" (→ p. 213) examples) to the Position input so its length matches the maximum number of axes allowed in the group.
2. Set the desired values for the additional axes in the now larger position arrays.

6.9.0.0.1.1 Circular Moves Diagrams

6.10 CircMode = BORDER

- The user defines the end point and a border point (= input AuxPoint) on the sector of the circle which the machine will traverse.
- For Relative mode, both points are defined relative to the starting point.
- Advantage
  - The border point can usually be reached by the machine, i.e., it can be taught.
- Disadvantage
  - Restricted to angles < 360° in one single command.



1. Starting point.
2. Border point.
3. End point.

6.11 CircMode = CENTER

- The user defines the end point and center point (= input AuxPoint) of the circle.
- The input PathChoice defines clockwise or counter-clockwise motion.
- For Relative mode, both points are defined relative to the starting point.
- Advantage
  - Full 360° moves are possible.
- Disadvantages
  - Cannot perform zero-distance moves.
  - Over-determination of the circle equation.



1. Starting point.
2. Center point.
3. End point.
4. Clockwise move.
5. Counter-clockwise move.

A 360° move is performed if the end point is the same as the start point.



1. Starting and Ending point.
2. Center point.
3. Clockwise move.
4. Counter-clockwise move.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.1 Add an HMI Device

To control an application, HMI panels can be downloaded to a dedicated HMI device.

It can also be embedded into targeted "Controllers" (➔ p. 767).

- Modbus TCP is used for communication from an HMI panel's Ethernet port to the X32 connector on the top of a AKD PDMM or PCMM.
- Modbus communication may be active while programming the controller through the KAS-IDE while the AKD PDMM or PCMM is connected to a network, switch, or hub.
- The KAS-IDE is automatically linked to the KVB HMI development software, and the Modbus addresses are automatically set.

When running the KAS Simulator, an internal HMI editor is also available to debug the application.

See "Design the HMI with the Internal Editor" (➔ p. 715).

6.11.1.1 Create a KVB Project

KVB projects are managed in the Project Explorer.

Use this procedure to create a project:

1. In the Project Explorer, right-click on the **System** item to open the contextual menu.
2. Select the **Add HMI device** command.
3. Select the device name within the list and click **OK**.

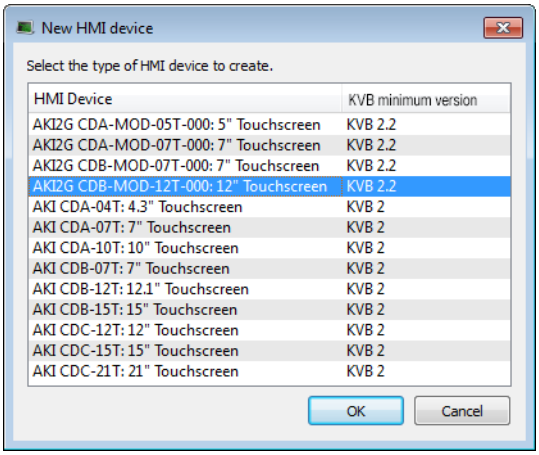


Figure 6-60: Select an AKI to add.

TIP

The content varies depending the version of KVB installed on your system.
If you have KVB 1.1 installed you may open projects that have a KVB 2.1 panel but you cannot open the panel in KVB.

4. Right-click on the newly created item and select the **Rename** command to change its name.
5. Right-click and select the **Add KVB Project** command.

NOTE

This command is disabled when a KVB panel is already created for the current HMI device.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.1.2 Map Variables to HMI

For HMI, the variable mapping is done in two phases.

- Phase 1 - Tag all of the variables to be exported from the HMI project (see procedure below).
- Phase 2 - Use this mapping file when designing the HMI

The tag operation directly takes place in the Dictionary:

1. Open the Variable list editor available in the **Dictionary** toolbox

TIP

Double-click the Dictionary header to view the widget as a popup window.

2. In the KVB column, select the variables to map.

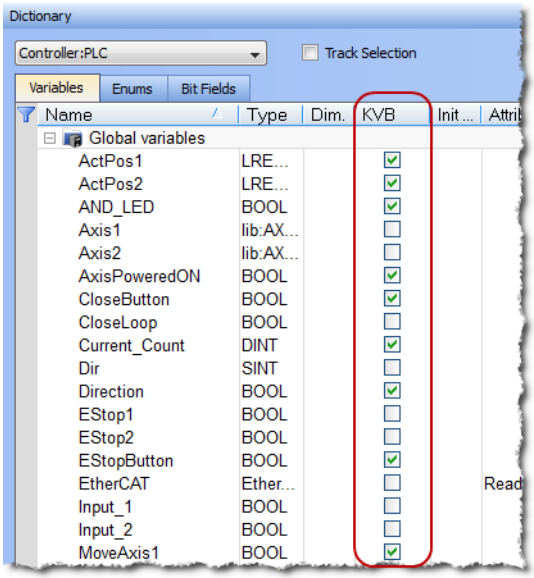


Figure 6-61: Variable Mapping to KVB.

TIP

Selecting the variable alone does not guarantee it is exported.
The POU must also be set to compile.
See "Ensure Variables are Exported" (→ p. 254).

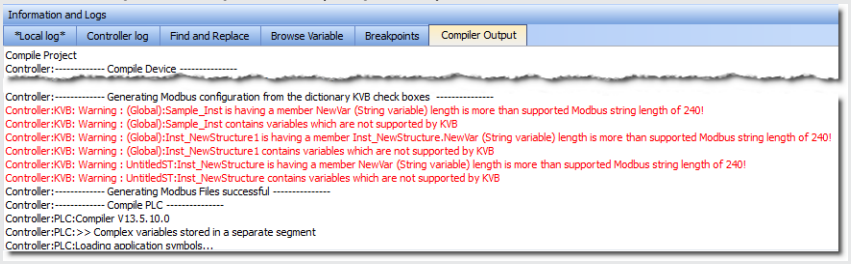
IMPORTANT

Based on Modbus, the communication is limited to 32-bits.
As a consequence:

- Data type conversion can lead to a loss in accuracy:
  - LREAL variables are saved as REAL.
  - LINT variables are converted to DINT.
  - ULINT variables are saved as UDINT.
- LWORD data types cannot be used in the HMI when using KVB.
  - This is because variables of this type are not exported, even when selected.
- STRING variable data types are allowed in the HMI when using KVB 2.0 or later
  - They are not supported in KVB 1.2.
  - STRING variables allow a maximum of 240 characters.
    - STRING variables whose size exceed the maximum string length of 240 characters generate a warning message in the Compiler Output window when being exported over Modbus.
    - Only the first 240 characters are exported.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- See "Compiler Output tab" (→ p. 136).



- Variables of types PNCode, ProfilesCode, or instances of UDFB cannot be used in the HMI.

TIP

Non-primitive data types which are not exported over Modbus generate a warning message in the Compiler Output window.
See "Compiler Output tab" (→ p. 136).

3. Compile the application to create the Modbus mapping file.

NOTE

- If the set of tagged variables is changed in the dictionary, the text file must be updated by recompiling the project.
- The Modbus variables defined in KAS-IDE are imported into Kollmorgen Visualization Builder only when KVB is started.
  There is no update in real-time between the two applications.

4. Open the **KVBProject** from the Project Tree.
   This automatically transfers the tagged variables.

6.11.1.2.1.1 Variables to Export

These types of variables can be exported to the HMI:

- The fundamental data types: BOOL, BYTE, DINT, DWORD, INT, LREAL, REAL, SINT, STRING, TIME, UDINT, UINT, USINT, WORD
- Arrays of supported data types
- Structures that include members of supported data types

6.11.1.2.2.2 Structures to Export

- Arrays of BOOLs such as a structure that includes a BOOL array member and instances of UDFB members.
- An array of structures that include INT and LREAL members
- A structure that includes both instances of UDFB member as well as an embedded structure which includes an INT member and instances of a UDFB member.
  - This is because the embedded structure has an INT member and the outer structure can be exported too.

6.11.1.2.3.3 Structure that Cannot be Exported

A structure that includes instances of UDFB or LWORD members only.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.1.3 Design KVB Panel with Kollmorgen Visualization Builder

- Double-click the new KVB panel to open the builder (for more details, refer to "Use the Kollmorgen Visualization Builder (KVB)" (→ p. 709).)

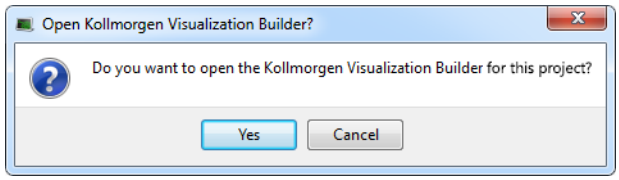


Figure 6-62: Open the Kollmorgen Visualization Builder Builder

IMPORTANT

You must have the specific application already installed on your machine.

TIP

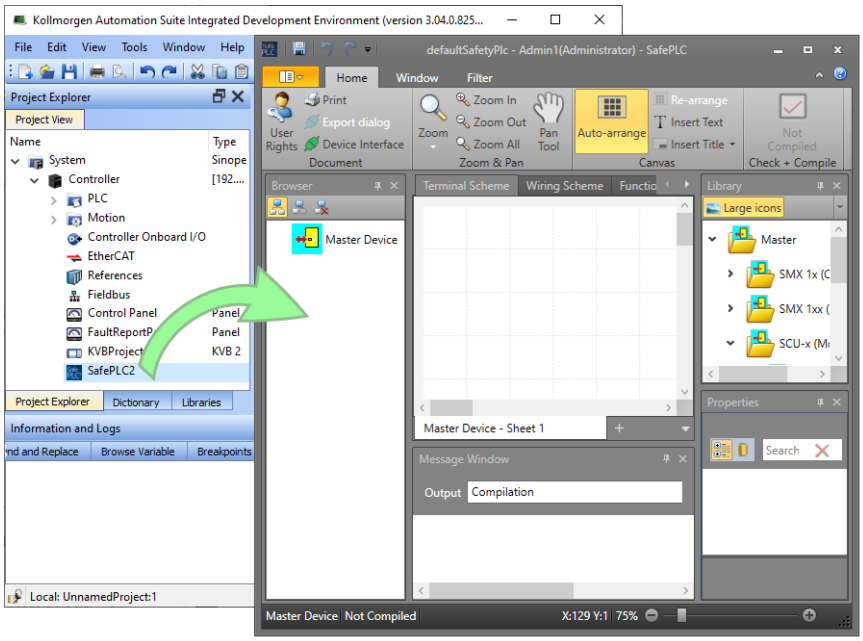
Be sure to close the Kollmorgen Visualization Builder before deleting the KVB Panel from the IDE.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.2 SafePLC2 Projects

A node can be added to the Project tree connected to a SafePLC2 project.

- This is used to synchronize a SafePLC2 project with a KAS project.
- The KAS-IDE automatically generates the PDOs for the connected devices and establishes the Black Channel with the EtherCAT Safety network.

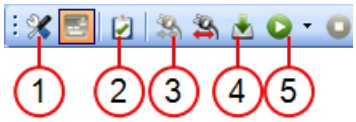


See the online help or the AKT2G I/O Manual for more information.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.3 Overview: Compile, Connect, Download, and Run

The last part of building a project is to compile the code, connect to the controller, download, and then run the application.



1. "Set the Compilation Options" (→ p. 257).
2. "Compile the Project" (→ p. 261).
3. "Connect to the Controller" (→ p. 262).
4. "Download the Project" (→ p. 265).
5. "Start / Stop the Device" (→ p. 267).

6.11.4 Create Project - Advanced

This section provides reference for setting up hardware, performing tasks which may not apply to every user, and other more advanced topics.

6.11.5 Add and Configure a Servo Drive

1. In the Project Explorer, right-click the **EtherCAT** node to open the menu.
2. Select the **Add Device...** command.
   This option is disabled when you are connected to the controller and an application is running.
3. Select the specific servo drive model to add. (Figure 6-63)

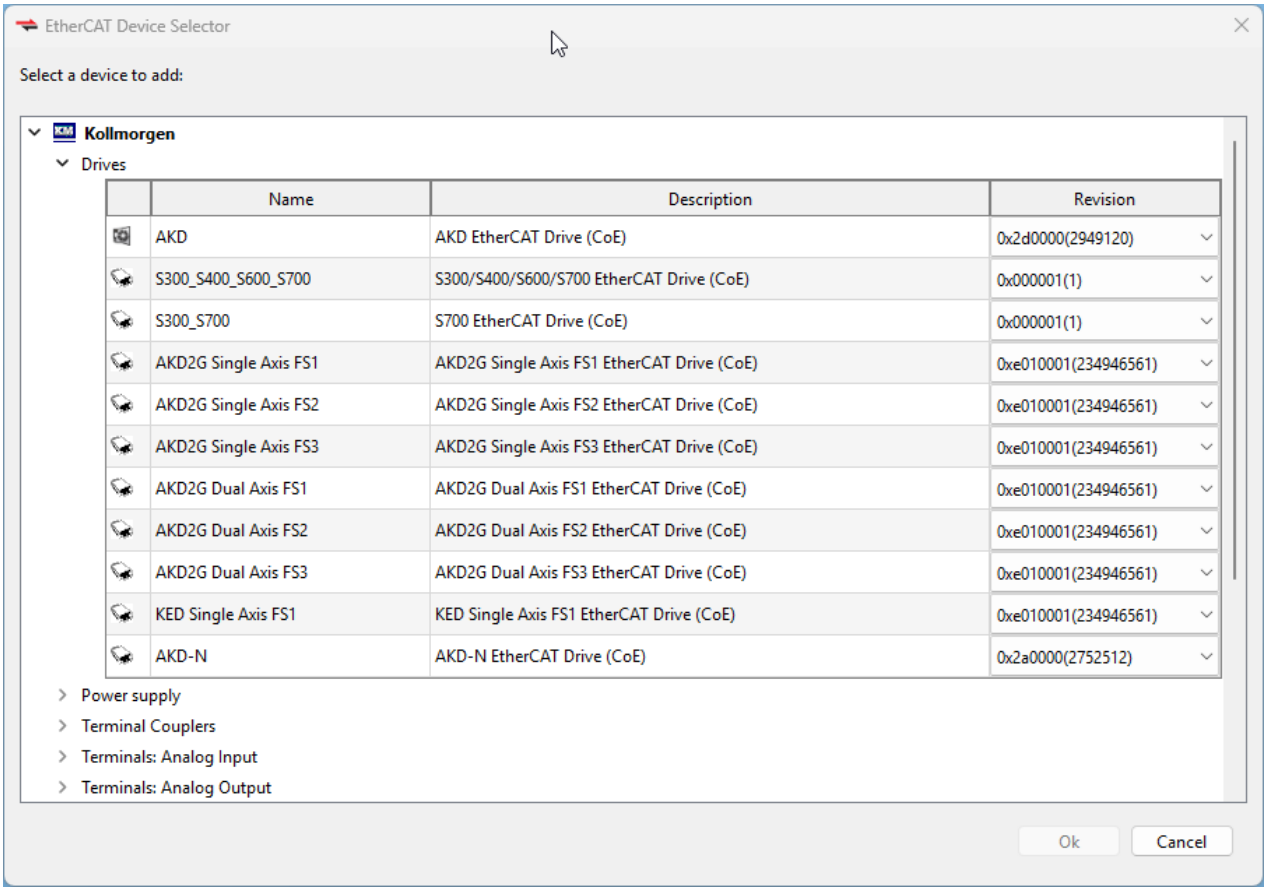


Figure 6-63: EtherCAT Device Selector dialog box

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

When choosing any of the Kollmorgen AKD Series Servo Drives, be sure to select the newest revision.
Older revisions may show in the list of options, but they may not be feature complete.
If a specific revision number is required, select it from the drop-down list in the Revision column.
The default selection is the highest revision number.

4. Click **Ok** when you are done.
   See "Kollmorgen AKD Series Servo Drive" (→ p. 143).
5. In the Project Explorer, right-click the device node and select the **Rename** command. (Figure 6-64)

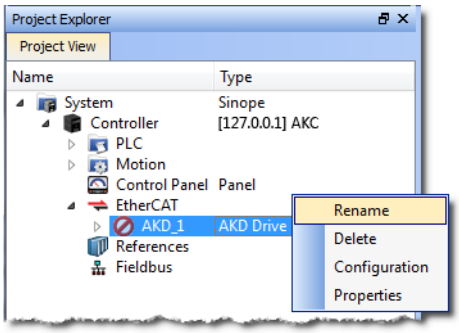


Figure 6-64: Rename

6. Define the name for the new device.
   The name is limited to 10 characters and can only include letters, numbers, and underscores. (Figure 6-65)

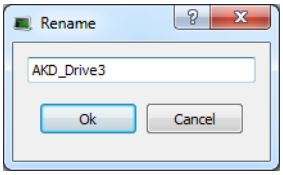


Figure 6-65: Rename dialog box

7. Click **OK**.
8. Continue with "Add and Configure Third-party EtherCAT Devices" (→ p. 234).

When an Kollmorgen AKD Series Servo Drive is added to the project tree, it must be mapped to a physical drive.

TIP

An alternative method to add a drive is to rely on the auto scan feature.

6.11.5.1 Configure the Kollmorgen Servo Drive

The **Configuration** tab under the EtherCAT: AKD tab is used to configure drives once these conditions have been met:

- The "EtherCATScan and Association" (→ p. 164) has been performed.
- Your project is compliant with the physical devices on the EtherCAT network.
- You have activated the Online Configuration mode in the "EtherCAT Toolbar" (→ p. 471).

You now have access to all of the AKD configurations and parameters.

1. In the Project Explorer, double-click the new AKD Drive or select Configuration from the context menu to open all the parameters linked to it. (Figure 6-66)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

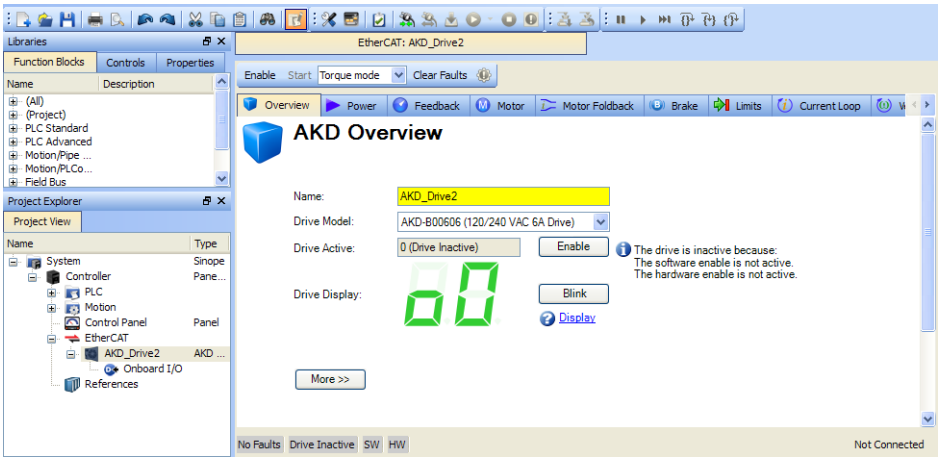


Figure 6-66: AKD Configuration

2. Define the motion parameters in the **Limits** tab to configure the limited motor torque, speed, and acceleration. (Figure 6-67)

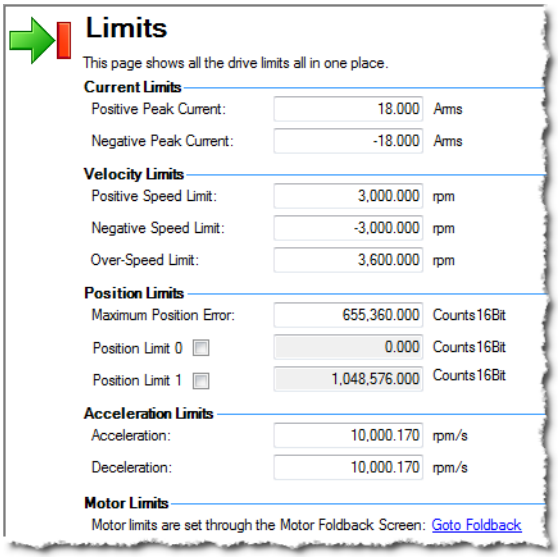
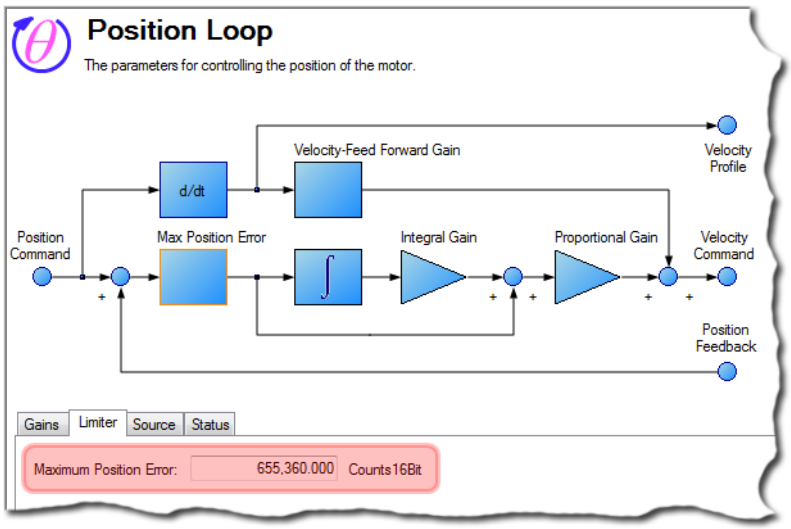


Figure 6-67: Limits tab

3. Define the motion parameters in the **Position Loop** tab to configure the limiting error. (Figure 6-68)



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Figure 6-68: Position Loop tab

4. Define the resolution of the feedback position in the **Feedback** tab.
   For all feedback types, the motor position feedback sent from the AKD drive to the controller through EtherCAT is normalized to 20 bits/rev or 1048576 counts/rev.
5. Define the units used for the motion [1]:
   - For Pipe Network, see "Set the Axis Block Position Units" (➔ p. 190).
   - For PLCopen, see "Axis Data Parameters" (➔ p. 202).

   NOTE

   User units in the PLC language editors are:
   - Position: User unit
   - Velocity User unit/sec
   - Acceleration: User unit/sec^2

   Several AKD tabs contain units that follow the standard AKD format:
   - Position: 16 bits/rev
   - Velocity: RPM
   - Acceleration: RPM/ Sec
6. Define the load for your servo system to ensure high performance.
   KAS-IDE provides several options for performing the drive tuning:
   - Slider Tuning - Use the slider to adjust to a designated bandwidth (pre-calculated tuning).
   - Performance Servo Tuner - Takes the drive through an automatic tuning sequence.
   - Manual Tuning - Used to set gains individually for:
     - Current Loop
     - Position Loop
     - Velocity Loop

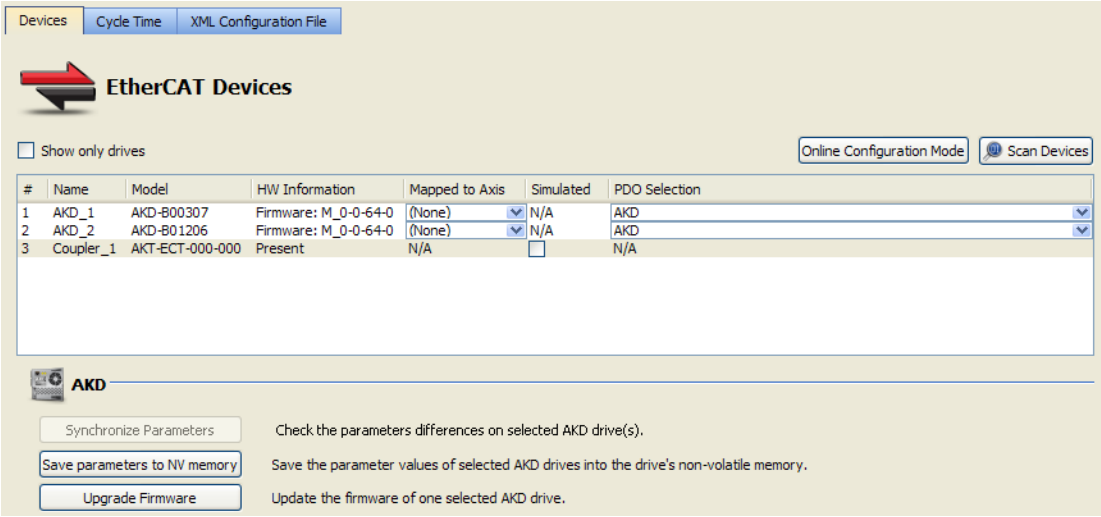
- See "AKD Drive Configuration" (➔ p. 448) about AKD configuration.
- See the "FAQs" (➔ p. 781) about AKD Firmware Download.

NOTE

After the application is downloaded to the controller, activate the "Online Configuration Mode" (➔ p. 471) to configure the drives with the "AKD Setup Wizard" (➔ p. 224).

6.11.5.2 Save Parameters to Non-Volatile Memory

The AKD parameters can be saved to non volatile memory in the drive. For more details, refer to "Save parameters to NV memory" (➔ p. 538)



[1]The normal units screen in the AKD WorkBench GUI is not included in the KAS-IDE.

6.11.5.3 AKD Setup Wizard

The wizard is used to configure drives once these conditions have been met:

- The "EtherCATScan and Association" (➔ p. 164) has been performed.
- Your project is compliant with the physical devices on the EtherCAT network.
- You have activated the Online Configuration mode in the "EtherCAT Toolbar" (➔ p. 471).

You then have access to the AKD parameters that are used when the drive is running.



Figure 6-69: AKD Setup Wizard

6.11.5.4 Configure Onboard I/O

The procedure to define the local I/Os on a Kollmorgen AKD Series Servo Drive is similar to the one for I/O slices, with these exceptions:

- Channel column contains the connector and pin number in brackets.
- PLC variable selection applies to digital inputs and analog inputs and outputs.
  - See "Map I/O from the Project Explorer" (➔ p. 550) for more information.

6.11.5.5 Digital Input Mode

This procedure configures the Digital Input mode setting inside WorkBench.

1. Connect to the controller.
2. Go to "Online Configuration Mode" (➔ p. 471).
3. Open the drive's WorkBench by double-clicking on the Kollmorgen AKD Series Servo Drive node in the project tree view.
4. Open the Digital Inputs and Outputs view. (Figure 6-70)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

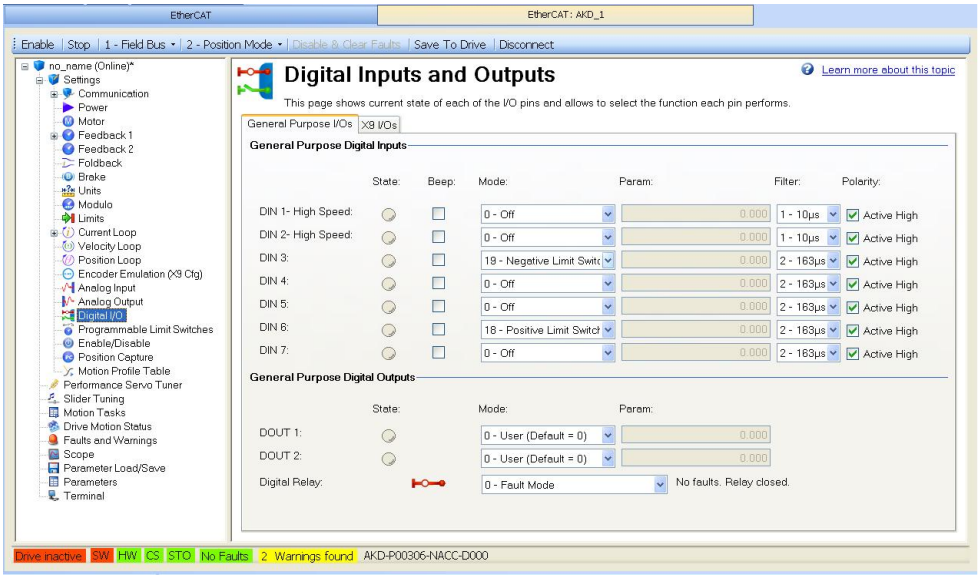


Figure 6-70: Digital Inputs and Outputs view

5. In the Mode column, use the drop-down list to set the mode of each Digital IO on the drive.
   See Digital Inputs and Outputs.

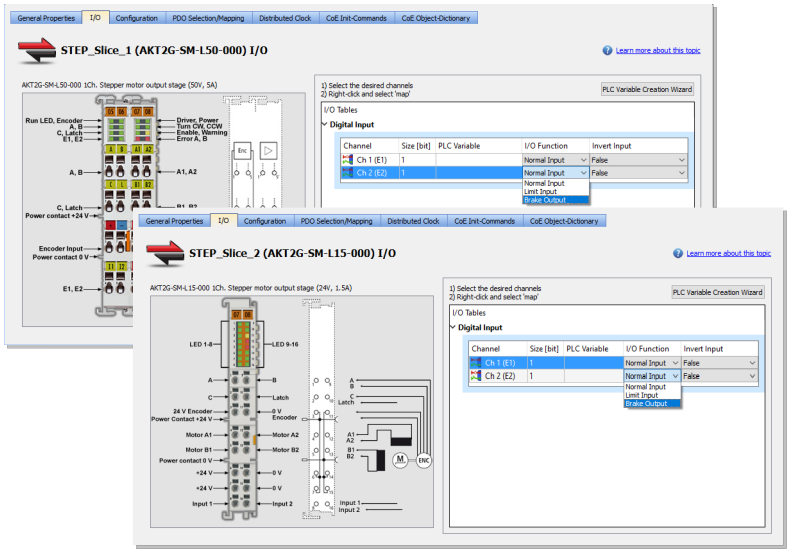
6.11.6 Add and Configure a Stepper Drive

These are the steps for adding an AKT2G-SM-L15-000 or AKT2G-SM-L50-000 stepper drive to the project.

1. Add an AKT2G-IO stepper I/O slice to the project.
   See "Add and Configure I/O Slices" (➔ p. 231) for details.
2. Configure the device.
   See "AKT2G-IO-SM-Lxx Stepper Drive Configuration tab" (➔ p. 226)
3. Assign variables to the I/O.
   See "AKT2G-IO-SM-Lxx Stepper Drive I/O Tab" (➔ p. 225)

6.11.6.1 AKT2G-IO-SM-Lxx Stepper Drive I/O Tab

This tab is used to assign PLC variables to device I/O.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Control	Description
PLC Variable column	A PLC variable may be assigned to each channel. • Double-click (or right-click and select **Map**) to open the PLC Variable Selector to choose or create a variable. • Alternatively, use the "PLC Variable Creation Wizard" (→ p. 530) to automatically create a list of variables for mapping.
I/O Function drop-down	The AKT2G-IO-SM-Lxx stepper slices provide two digital inputs. • **Normal Input**: This selection configures the I/O channel as a normal input. • **Limit Input**: This selection configures the I/O channel as the hardware limit switch. The drive is disabled when a digitial input acting as a source for the hardware limit is triggered. • **Brake Output**: This selection configures the I/O channel as the **Brake Output**. This option is supported only for the second I/O channel on the terminal.
Invert Input drop-down	This option inverts the state of the input when set to True. • The default is False. • This setting is applicable only when the I/O channel is configured as either **Normal Input** or **Limit Input**.

TIP

- See "Add and Configure I/O Slices" (→ p. 231) for adding a stepper drive to your project.
- See AKT2G-IO-SM-L15-000 or AKT2G-IO-SM-L50-000 for information on the devices.
- See AKT2G-IO-SM-Lx Object Description for stepper drive-specific object descriptions.

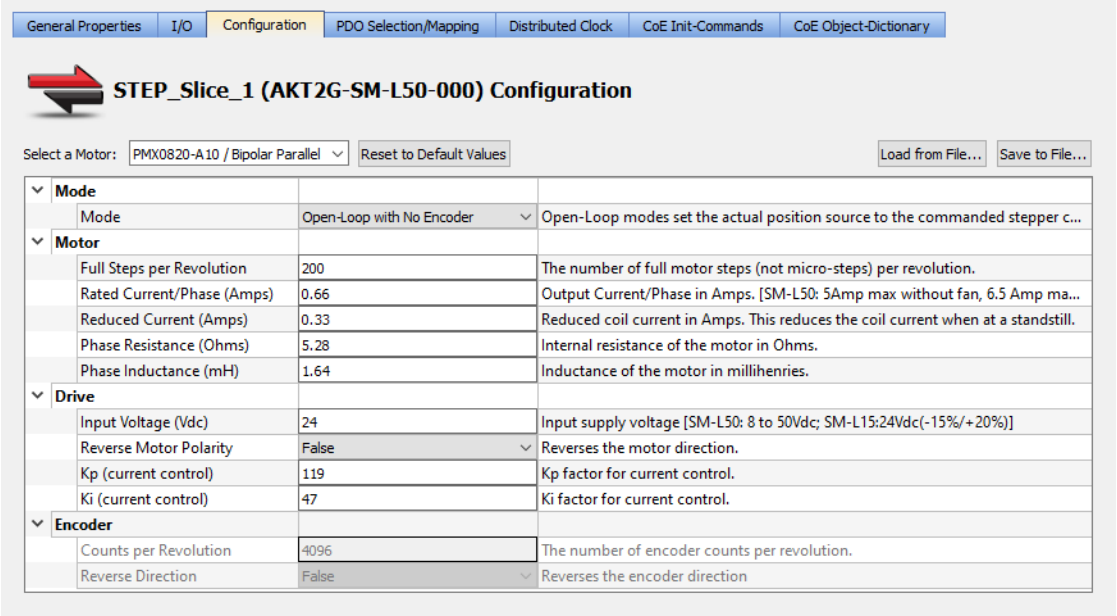
6.11.6.2 AKT2G-IO-SM-Lxx Stepper Drive Configuration tab

This tab is used to configure the device, based on the parameters of the stepper motor being used, and to diagnose any problems with the stepper terminal.

- To make configuration simple, a Kollmorgen stepper motor can be selected from the "Properties tab" (→ p. 226).
  - This automatically populates the stepper motor parameter fields.
- The stepper motor parameters can be replicated to a file and transferred between devices.
  - Once the project is compiled, the "CoE Init Commands tab" (→ p. 570) is populated based on these values.
- Troubleshooting the stepper terminal can begin by gathering information from the "Diagnostics tab" (→ p. 230).

6.11.6.2.1 Properties tab

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



Control	Description
Select a Motor drop-down	The drop-down list contains the Kollmorgen stepper motor models and their specific parameters. • Selecting a model automatically populates the appropriate motor-specific fields. • See the Kollmorgen Stepper Optimizer to match a motor series name to the model number.
Reset to Default Values button	Click to change all of the stepper motor parameters to their default values.
Load from File... button	Click to select an XML file to populate all of the stepper parameters, not just those specific to the motor.
Save to File... button	• Click to create an XML file containing all of the current stepper parameters. • This file can be stored or transferred between devices.

6.11.6.2.2 Mode Section

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Element	Description
Mode	Select the CoE object used for the actual position. • **Open Loop with No Encoder Feedback**: Select this option when there is no position feedback device for the stepper motor. In this mode: • The command position increments sent to the stepper drive are accumulated in an internal counter and returned to the controller as the actual position. • The PLCopen and Pipe Network axis motion function blocks (e.g., MC_ReadActPos, MLAxisReadActPos) return actual position. • This is the stepper drive's internal counter. • The controller commands position without considering position feedback. • There is no assurance that the motor will get to its destination. • **Open Loop with Encoder Feedback**: In this mode: • The command position increments sent to the stepper drive are accumulated in an internal counter. • The encoder feedback is returned to the controller as the actual position. • The PLCopen and Pipe Network axis motion function blocks (e.g., MC_ReadActPos, MLAxisReadActPos) return actual position. • This is the encoder feedback counter. • The controller commands position without considering position feedback. • There is no assurance that the motor will get to its destination because there is no ability to self-correct. • **Configure via CoE Init-Commands**: This mode allows for custom drive/motor configurations. In this mode: • The "Drive Section" (➔ p. 228) and "Motor Section" (➔ p. 228) configurations listed in this table do not apply. • Stepper motor configuration parameters (e.g., Features, Controller, Motor, Encoder, etc.) must be configured manually in the "CoE Init Commands tab" (➔ p. 570). **NOTE** When a custom configuration is required by your application, contact Kollmorgen Customer Support at support@kollmorgen.com.

6.11.6.2.3 Drive Section

Element	Description
Input Voltage (VDC)	Specify the input supply voltage. The range is: • SM-L15 = 24 VDC (-15%/+20%) • SM-L50 = 8 - 50 VDC
Reverse Motor Polarity	Select **True** or **False** from the drop-down menu to reverse the motor direction.
Speed Range	Select the maximum output full-step frequency from the drop-down list.
Kp (current control)	Specify the proportional gain factor for current control.
Ki (current control)	Specify the integral gain factor for current control.
Fan Cartridge Present	Set this to **True** if a AKT2G-AC-FAN-001 fan cartridge is installed. Having a fan present allows you to set the "Rated Current / Phase (Amps)" (➔ p. 229) to a higher value.

6.11.6.2.4 Motor Section

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Element	Description
Full Steps per Revolution	Specify the number of full motor steps per revolution.
Rated Current / Phase (Amps)	Specify the motor's rated current per phase in Amps. • This is also known as the RMS (root mean square) current. • This value is converted from RMS to Peak output current, by multiplying the value by 1.414 (square root of 2) to specify the maximum output current for the stepper drive. • The 64x microstep resolution causes a 30% current loss compared to full-step resolution. • This is due to the output wave form being closer to a sine wave than a square wave. • The rated current/phase value is converted to the peak output current to compensate for the current loss. • Since the peak output current may exceed the motor's rated current per phase at standstill, it is very important to configure the "Reduced Current/Phase (Amps)" (→ p. 229), to prevent the motor from overheating. Maximum Motor Current/Phase (RMS) Values • SM-L15 = • 1.06 Amps without a fan • 1.12 Amps with a fan • SM-L50 = • 3.53 Amps without a fan • 4.59 Amps with a fan Maximum Converted Peak Output Values • SM-L15 = • 1.5 Amps without a fan • 3 Amps with a fan • SM-L50 = • 5 Amps without a fan • 6.5 Amps with a fan
Reduced Current/Phase (Amps)	Specify the motor's current per phase in Amps when at standstill. • This is known as the RMS (root mean square) current. • This value is converted from RMS to Peak output current by multiplying the value by 1.414 (square root of 2) to specify the maximum output current for the stepper drive when the motor is at standstill. • The Reduced Peak Current is applied when there is no motion for > 1 sec. • At a minimum, the Reduced Current/Phase can be set to 70% of the "Rated Current / Phase (Amps)" (→ p. 229). • Setting the Reduced Current/Phase to 50% (or lower) of the Rated Current/Phase is recommended.
Phase Resistance (Ohms)	Specify the internal resistance of the motor in Ohms.
Phase Inductance (mH)	Specify the motor's inductance in millihenries.

6.11.6.2.5 Encoder Section

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Element	Description
Counts per Revolution	Specify the number of encoder counts per revolution.
Reverse Direction	Select **True** or **False** from the drop-down menu to reverse the encoder direction.

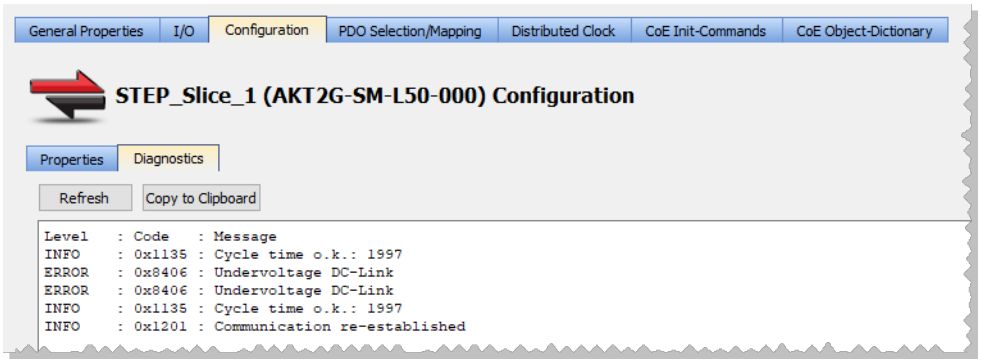
TIP

- See "Add and Configure I/O Slices" (→ p. 231) for adding a stepper drive to your project.
- See AKT2G-IO-SM-L15-000 or AKT2G-IO-SM-L50-000 for information on the devices.
- See AKT2G-IO-SM-Lx Object Description for stepper drive-specific object descriptions.

6.11.6.2.6 Diagnostics tab

This tab gathers the diagnostic messages available on the connected stepper terminal.

- This information can be used for troubleshooting during commissioning or operation.
- The PLC application must be running on a controller or Online Configuration mode must be enabled to acquire diagnostics data.



Refresh	Click this button to read and view the stepper terminal's messages. See AKT2G-SM-Lxx Diagnostic Messages for a comprehensive list of messages. This button is disabled under these conditions. • The PLC application is neither running nor in online configuration mode. • A stepper I/O slice is not present on the EtherCAT network (e.g., offline device). • The PLC application is running on the KAS Simulator.
Copy to Clipboard	Click to copy the message data.

If the application is not running, follow this procedure.

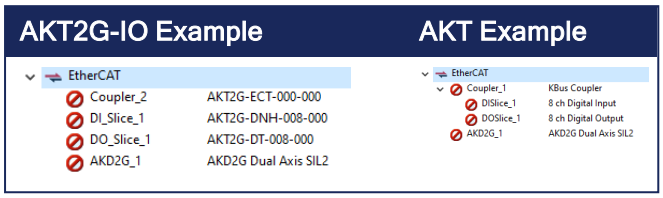
1. Connect to your controller.
2. Compile and download the application.
3. Enable "Online Configuration Mode" (→ p. 471).
   - OR -
   Start the application.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.7 Add and Configure I/O Slices

This topic discusses adding AKT (K-Bus) and AKT2G-IO (EtherCAT/E-Bus) I/O devices. All devices are children of the EtherCAT node in the Project tree.

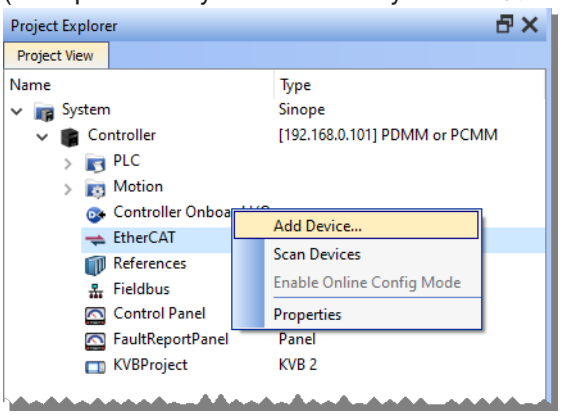
- AKT2G-IO (E-Bus) devices are each directly under the EtherCAT node.
- AKT (K-Bus) slices are children of the coupler, which is a child of the EtherCAT node.



For local I/O, see Configure Onboard I/O (→ p. 224).

6.11.7.1 Add an AKT (K-Bus) I/O Coupler

1. In the Project Explorer, right-click the EtherCAT node to open the menu.
2. Select the **Add Device** command.
   (this option is only enabled when you are **not** connected to the controller)



3. Select the coupler and click **OK**.
4. In the Project Explorer, right-click the coupler node and select the **Rename** command to change the name.
5. Click **OK**.
   The coupler is added to the EtherCAT node.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.7.2 Add an AKT (K-Bus) I/O Slice

1. In the Project Explorer, right-click the Coupler node to open the menu.
2. Select the **Add I/O Slice** command.
3. Choose the I/O slice from the list.

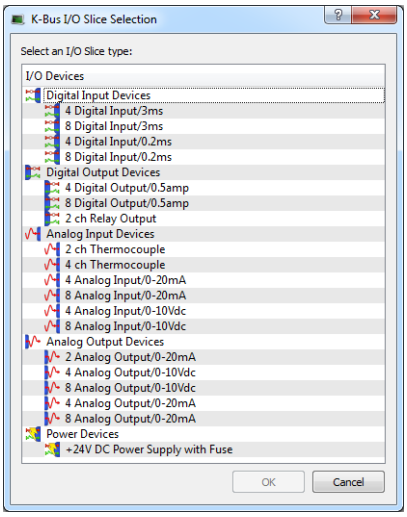
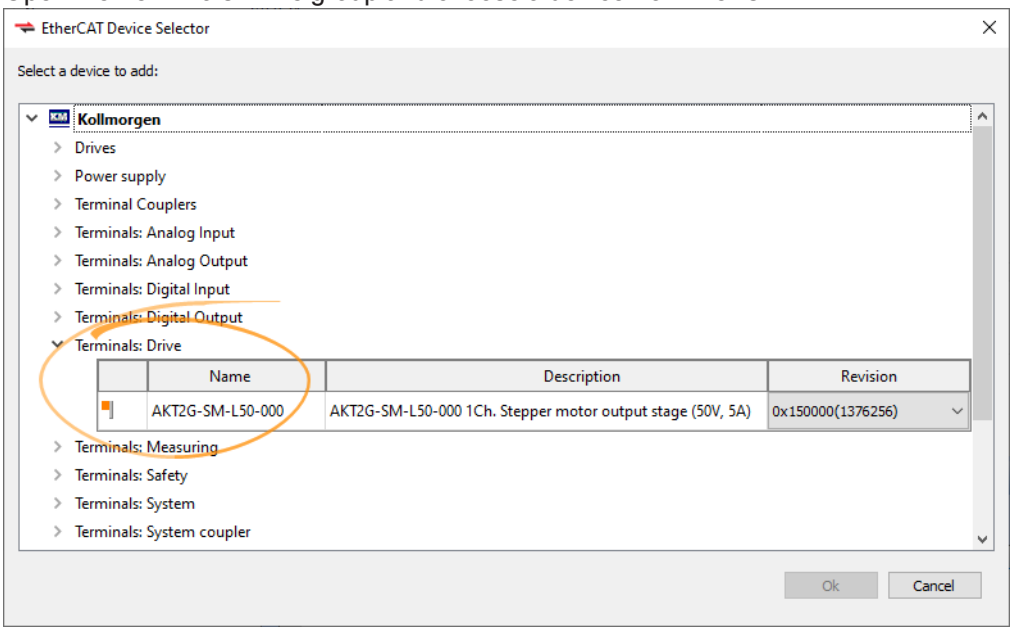


Figure 6-71: Add I/O Slice

4. Click **OK**.
   The device is added below the coupler.

6.11.7.3 Add an AKT2G-IO Stepper Drive

1. In the Project Explorer, right-click the EtherCAT node to open the menu.
2. Select the **Add Device** command.
3. Open the Terminals:Drive group and choose a device from the list.



4. Click **OK**.
   The device is added to the EtherCAT node.

For more information see:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "AKT2G-IO-SM-Lxx Stepper Drive I/O Tab" (➔ p. 225)
- AKT2G-IO-SM-L15-000 for product information
- AKT2G-IO-SM-L50-000 for product information

6.11.7.4 Configure the I/O Slice

See "Map Input and Output to Variables" (➔ p. 526).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.8 Add and Configure Third-party EtherCAT Devices

EtherCAT devices physically connected to the network can be scanned and discovered.

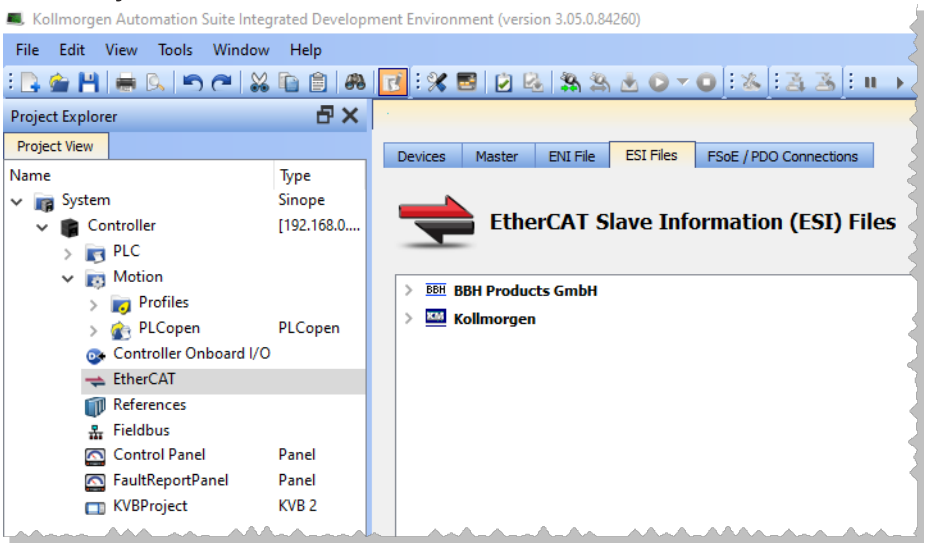
- EtherCAT devices can be manually added to the EtherCAT node in the Project view.
  - Manually added devices allow the project to be configured before connecting.

NOTE

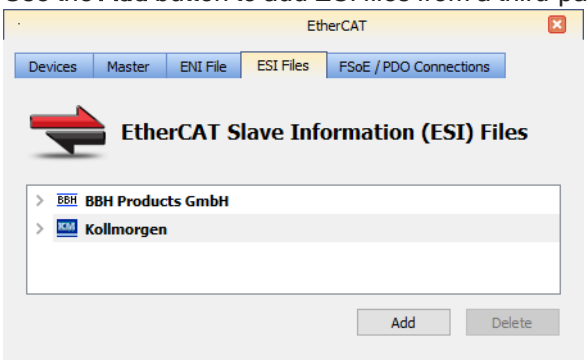
- KAS has limited support for third-party drives that follows the CANopen DS-402 standard.
  - Contact your Kollmorgen representative for details.
- KAS supports third-party FSoE Masters.
- BBH FSoE Master SCU-1-EC support is integrated with the KAS-IDE and is recommended.
  - Contact BBH for the latest ESI file that matches with the SCU-1-EC firmware version.
  - See the BBH web-site for SCU-1-EC.
- Other third-party FSoE Masters are possible but require the FSoE/Connected PDOs to be configured manually by the user.

Procedure

1. In the **Project View**, double-click the EtherCAT node and select the **ESI Files** tab.



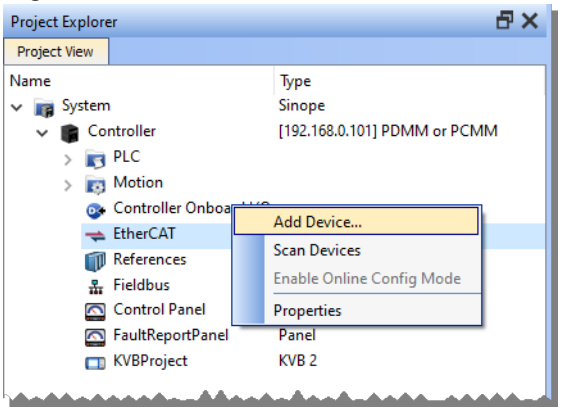
2. Use the **Add** button to add ESI files from a third-party vendor.



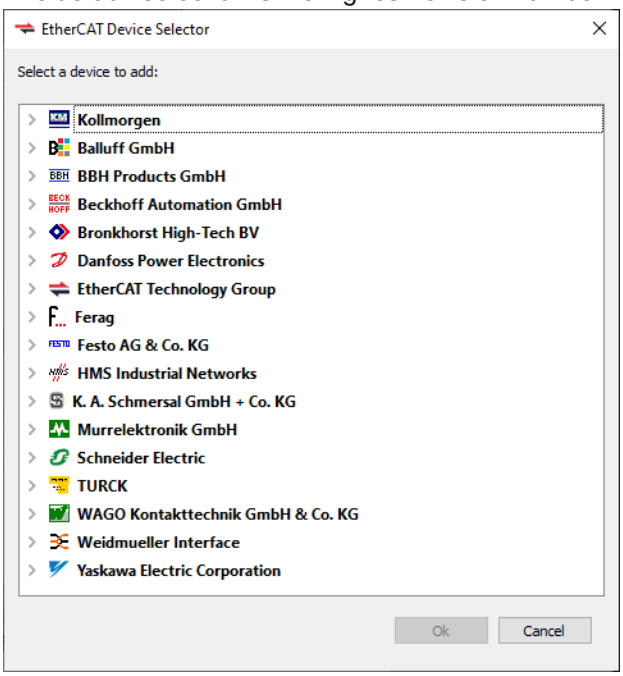
3. Navigate to where you have the third-party ESI file saved on your computer and click **Open**.
   This adds the manufacturer and device to the list of available devices.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4. Right-click the EtherCAT node and select **Add Device...**.



5. Select the device from the **EtherCAT Device Selector** dialog box.
   If a specific revision number is required, it may be selected from the drop-down list in the revision column.
   The default selection is the highest revision number.



6. Click **OK** to save the changes or selections and close the dialog box.
   The device is added to the project tree in the EtherCAT node.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.8.1 Add Third-party EtherCAT Modules

Modules can be added to third party MDP (Modular Device Profile) EtherCAT devices in the Project view.

This helps to pre-configure the EtherCAT network in the project before connecting to the controller.

Procedure

1. Right-click a MDP EtherCAT device and select **Add Module...** from the menu. (Figure 6-72)

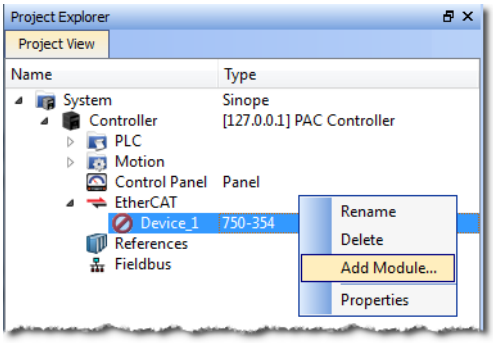


Figure 6-72: Right-click menu - **Add Module...** selected

2. Select a module from the **EtherCAT Module Selector** dialog box.
3. Click **OK** to save the changes or selections and close the dialog box.

The module is added to the project under the selected EtherCAT device.

6.11.8.1.1 Limitations

When adding modules from the Project View, there are limitations:

- Modules cannot be manually added to devices that are mapped to physical devices.
  - Modules are only added to an unmapped, manually added MDP device.
- Manually adding a MDP device does not automatically add mandatory modules.
  - Either:
    - Consult the device documentation to identify the mandatory modules and add them manually.
    - Discover the mandatory modules under the physical MDP devices, using the EtherCAT network scan.

6.11.8.2 Configure a Third-party Drive

NOTE

KAS has limited support for third-party drives that follows the CANopen DS-402 standard.

Restrictions

These restrictions apply when using a third-party drive:

- Only the Cyclic Synchronous Position mode (op-mode = 8) is supported.
  - No other modes (e.g., velocity, torque, etc.) are supported.
- PDOs are not automatically selected.
  - The initial selection of PDOs is determined by the ESI file.
- CoE Init Commands are not added automatically.
  - Initial selection of Init Commands is determined by the ESI file.
- No third-party drive-specific I/O or configuration tabs are displayed.
- ESI files for third-party devices are not included in the KAS-IDE installation package.

Procedure

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. Add the third-party drive to the EtherCAT network.
2. In the KAS-IDE:
   a. Add the ESI file for the third-party drive.
   b. Scan the EtherCAT network to discover the devices.
   The Discovered Devices and Modules dialog box opens. (Figure 6-73)

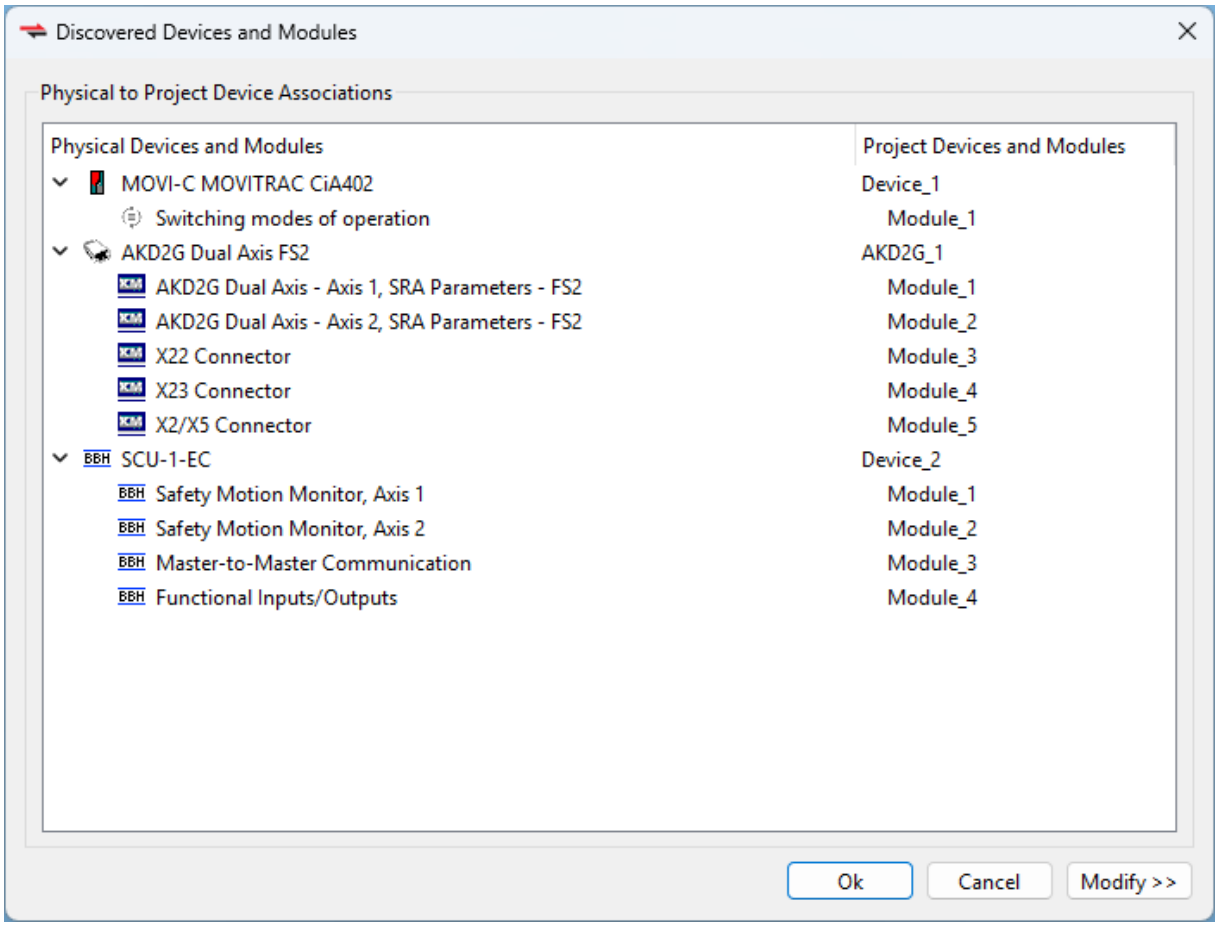


Figure 6-73: Discovered Devices and Modules dialog box

   c. Click **OK** to save the changes or selections and close the dialog box.
   The device is added to the project tree in the EtherCAT node.
3. Optional: In the Project Explorer, right-click the new device node and select **Rename**.
   The **Rename** dialog box opens.
   a. In the **Rename** text box, enter a name for the device.
   b. Click **OK** to save the changes or selections and close the dialog box.
4. In the Project Explorer, double-click the new device node.
   The EtherCAT device configuration tab opens. (Figure 6-74)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

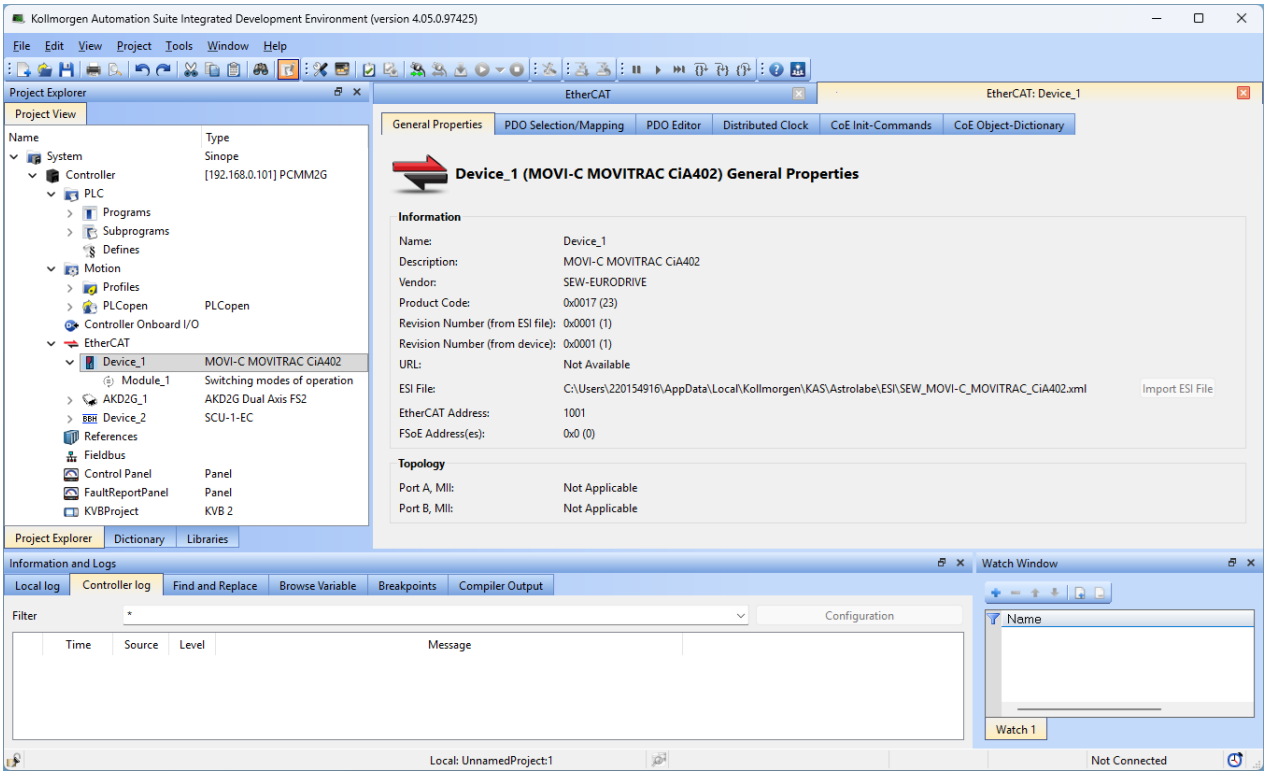


Figure 6-74: EtherCAT device configuration tab

5. Navigate to the "PDO Selection/Mapping tab" (→ p. 554). (Figure 6-75)

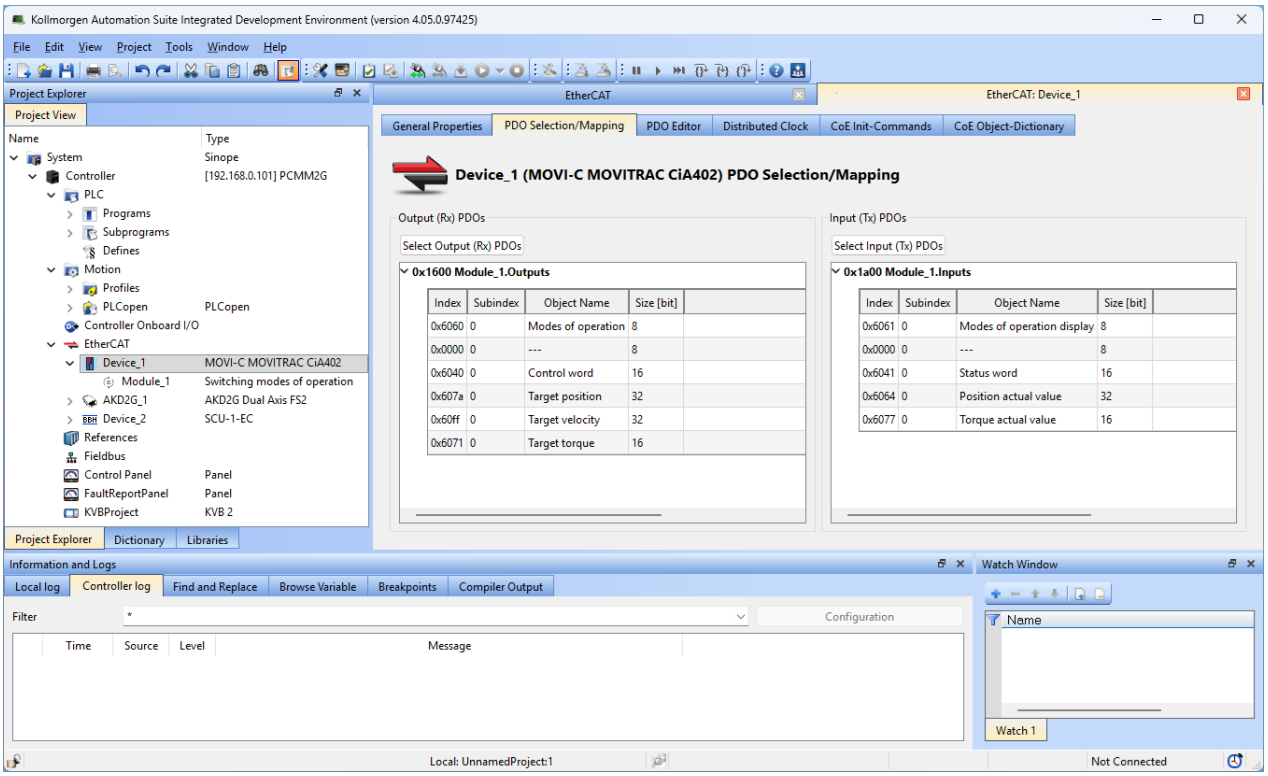


Figure 6-75: PDO Selection/Mapping tab

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6. Verify these PDOs are present:
   a. Input (Tx) PDOs:
      - Status Word: Index 0x6041, Subindex: 0 (zero)
      - Position Actual Value: Index 0x6064, Subindex: 0 (zero)
   b. Output (Rx) PDOs:
      - Modes of Operation: Index: 0x6060, Subindex: 0 (zero)
      - Target Position: Index: 0x607A, Subindex: 0 (zero)

NOTE

If the PDO Selection/Mapping tab does not include these PDOs by default, use the "PDO Editor tab" (➔ p. 560) to included them.

7. Navigate to the "Distributed Clock tab" (➔ p. 568) and "CoE Init Commands tab" (➔ p. 570).
8. Configure the drive as needed for the application.

6.11.8.3 Delete EtherCAT Devices or Modules

One or multiple EtherCAT devices, modules, or slices can be deleted from the project tree simultaneously.

This allows updates to the KAS project by removing these unused items whenever the EtherCAT network configuration is modified.

IMPORTANT

- Deleted devices, modules, or slices cannot be restored (i.e., there is NO <Ctrl+Z>).
- If needed, new devices or modules must be added to replace what was deleted.
- See either:
  - "Add and Configure Third-party EtherCAT Devices" (➔ p. 234).
  - "Add Third-party EtherCAT Modules" (➔ p. 236).

Procedure

1. In the Project tree, select the MDP EtherCAT devices or modules to delete.
   Only the same type of multiple devices or modules can be selected for deletion (e.g., multiple devices (Figure 6-76), multiple modules (Figure 6-77), multiple slices (Figure 6-78), etc.)

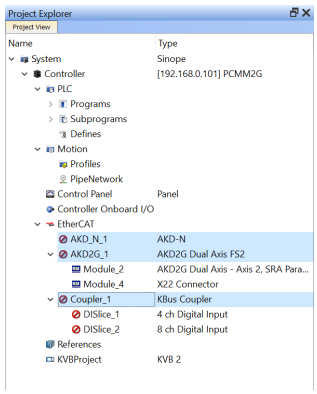


Figure 6-76: Multi-selected devices

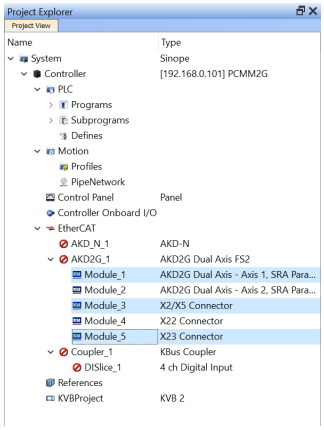


Figure 6-77: Multi-selected modules

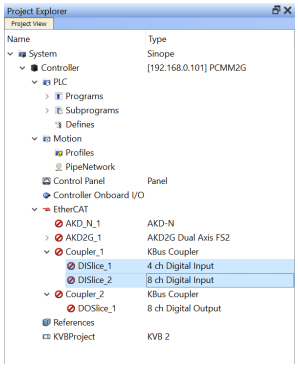


Figure 6-78: Multi-selected slices

2. Press the <Delete> key or
   Right-click and select **Delete** from the menu. (Figure 6-79)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

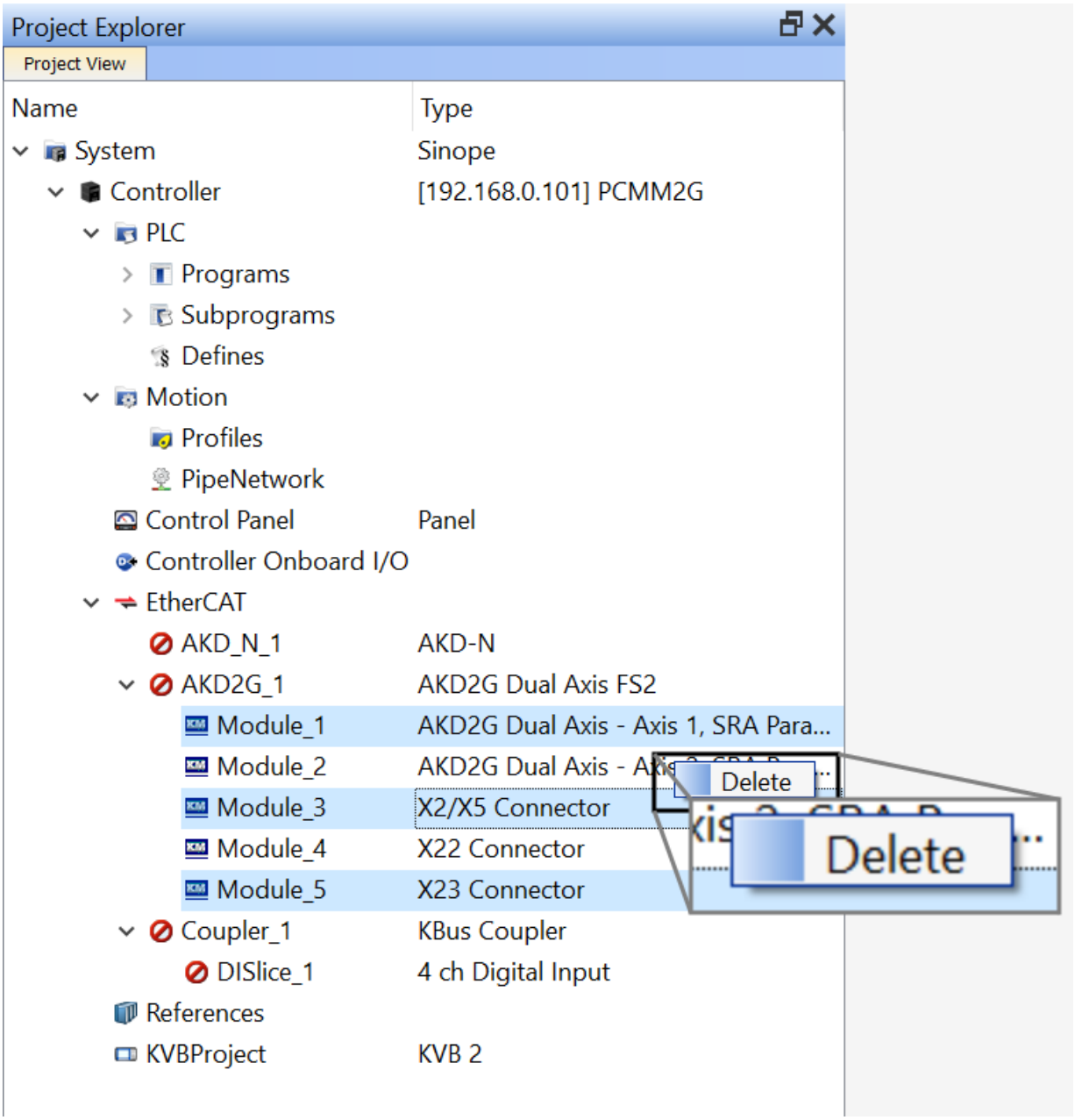


Figure 6-79: Right-click menu - **Delete** selected

A confirmation message appears. (Figure 6-80)

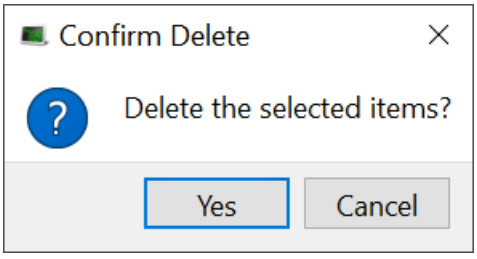


Figure 6-80: Confirmation message

3. Click **Yes** to confirm the deletion.
   The selected devices or modules are removed.
   This action cannot be undone.

6.11.8.3.1 Limitations

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

When deleting devices or modules from the Project View, there are limitations:

- The multi-select delete feature is only applicable to EtherCAT in the tree.
- Only the same type of multiple devices, modules, or slices can be selected for deletion.
- Press <Shift> to select adjacent items in the tree.
- Press <Ctrl> to select specific items in the tree.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.8.4 Modules - General Properties tab

A device may have no modules or it may have several.

- Double-clicking or right-clicking on the module opens the **General Properties** tab.
- This tab provides information about the selected module.

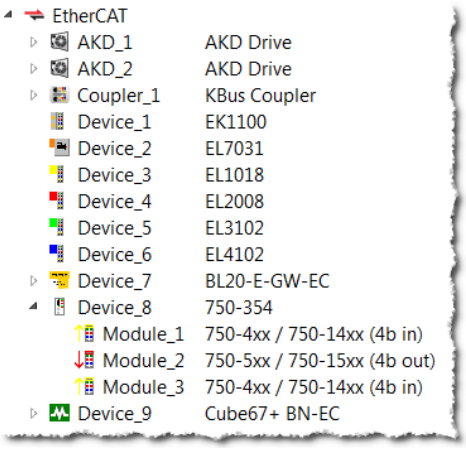


Figure 6-81: Devices in the EtherCAT list

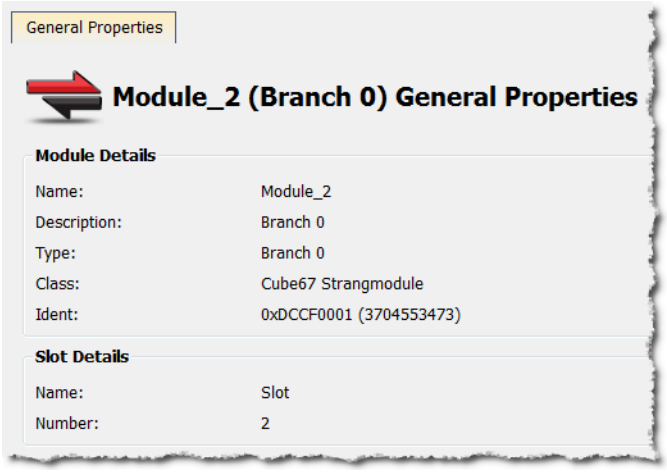


Figure 6-82: Example: General Properties tab - Module and Slot Details

The tab is divided into two sections:

- Module Details
  - Module Details provides the module's basic information.
- Slot Details
  - Slot Details provides the module's slot information.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.9 Use Pre-defined Libraries

The Libraries toolbox is used to select the functions.

TIP

- The **(All)** category at the top enables you to see the full list of available blocks.
- Access a specific function by entering its initial letters on the keyboard (if the elapsed time between two strikes is greater than 1 second, the KAS-IDE considers the last letter as the new initial).

6.11.9.1 Drag-and-Drop into the Editors

1. When the function is selected, move it with a drag-and-drop operation in the program editor
2. In the editor, right-click on a function to set the number of input pins if the block allows an extension.

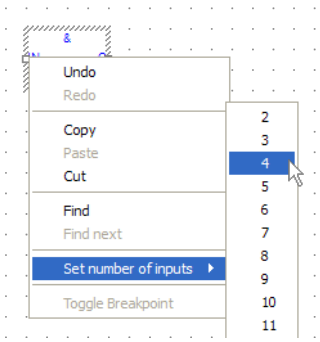


Figure 6-83: Set the Pins Number of the Block

6.11.9.2 Drag-and-Drop into the Dictionary

If you have selected a function block, you can drag-and-drop it in the program declaration within the Dictionary toolbox, to create an instance of that object.

6.11.10 Create and Use Custom Libraries

You need to create a custom library before you can use it to define a new item (e.g., function, function block, or variable).

See "Use the Custom Library" (➔ p. 244).

NOTE

There is a difference between **Libraries** usage and the **Import / Export** commands related to PLC programs.

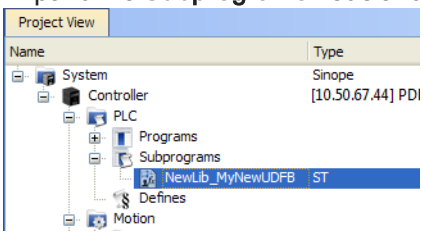
- **Library** is a unique source that can be shared between different projects (like a dll in C): when you modify the library, all the linked projects are impacted.
- **Import/Export** is equivalent to a copy and paste operation of programs: when you update the source of your UDFB, the other programs are not updated because the code has been duplicated.

6.11.10.1 Create a Custom Library

1. In the **File** menu, click the **New** command (save your current opened project if necessary).
2. Select the **Library** application template.

Figure 6-84: Create a Custom Library - Select the Library Template

3. Click the **Finish** button.
4. In the **File** menu, click the **Save As** command.
5. Define the Library Name (extension *.kal) and its Location.
6. Click **OK**.
7. In the Project Explorer, expand the **Controller** and **PLC** nodes.
8. Right-click on **Subprograms** and choose **New UDFB** in the contextual menu.
9. Select the type of programming language.
10. Expand the **Subprograms** node and rename the new UDFB.



NOTE

The name of the variable type is displayed in the dictionary if you use this library in another project.

IMPORTANT

Duplicate UDFB names in a library are not possible.
Only the first instance found is kept when importing the library definitions in a project.
To avoid this situation when designing your libraries, use a prefix to identify the library for all UDFBs and functions in the libraries (e.g., in the this procedure, the prefix is: NewLib_).

11. Create the UDFB program.
    See "Programs: Structure and Syntax" (➔ p. 481) for more information.
12. In the **File** menu, click the **Save** command.

6.11.10.2 Use the Custom Library

1. Open the project where you want to use a library.
2. In the Project Explorer, expand the **Controller** node.
3. Right-click **PLC** and select **Libraries** in the menu.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4. Click **Add**.

NOTE

You can add as many external libraries as you want.

5. Select a previously created *.kal file and click **Open**.

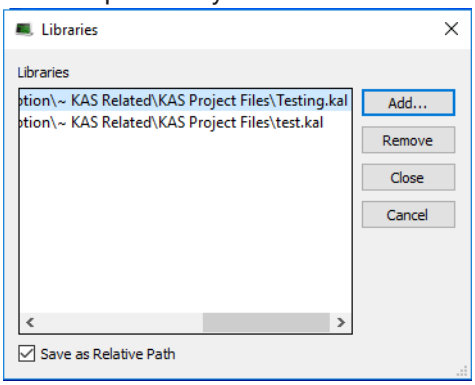


Figure 6-85: Use a Custom Library - Select the Library

TIP

Use the **relative** path to specify the path relative to the working directory where your project is saved.
This ensures consistency when you move your project and your library.
Conversely, the **absolute** path points to the same location on your file system regardless of your project directory.

6. Click **Close**.
The library is displayed in the Library widget and you can now drag-and-drop the UDFB (or any subprogram) of this library in any editor.

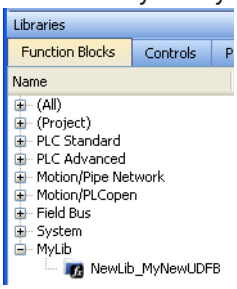


Figure 6-86: Use a Custom Library - Display the Library

7. In the Dictionary toolbox, right-click the program and select **Add variable**.

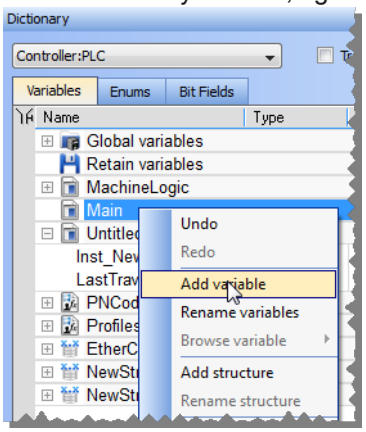


Figure 6-87: Use a Custom Library - Add variable

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8. In the **Type** drop-down menu, select the type defined in the external library (it can be at the bottom of the list).

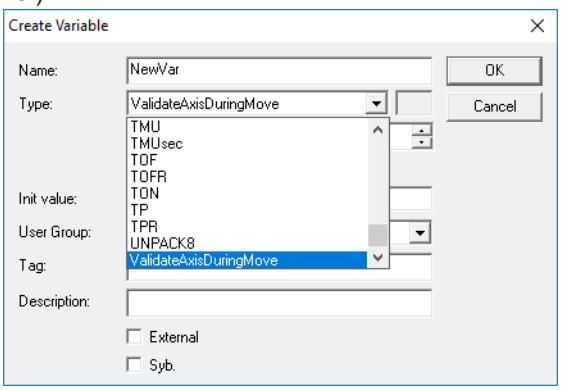
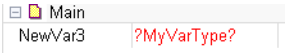


Figure 6-88: Use a Custom Library - Select the Type

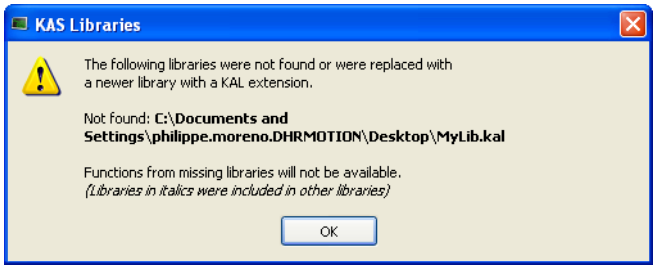
6.11.10.2.1 What happens when you remove a library from your project?

If you remove a library from your project, all its types are removed from your project and all variables based on the library are displayed in the dictionary in red with question marks.



6.11.10.2.2 What happens when a library no longer exists?

If you open a project containing a link on a library which is no longer available, a warning message appears:



These are the options to recover the libraries:

- Enter the new path to this library (assuming it still exists on your machine) using the library dialog (see "Use the Custom Library" (➔ p. 244)).
- Find the missing library and copy the library back to the path originally specified.
  Note that the project has to be closed and re-opened for the library to be read again.

6.11.10.2.3 Broken Link Displayed in Italics

If a library references another library which is no longer available, a dialog with the library link that causes the problem is displayed in italics.

Example: If your project has referenced LIB-4, which in turn references LIB-1-ND, but LIB-1-ND does not exist. To recover your project, open LIB-4 and fix the issue (i.e., LIB-1-ND broken link), then re-open this project again.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.11 Add CAM Profiles

6.11.11.1 Create a CAM Profile

1. In the Project Explorer, right-click the **Profiles** item and click the **New profile** command.

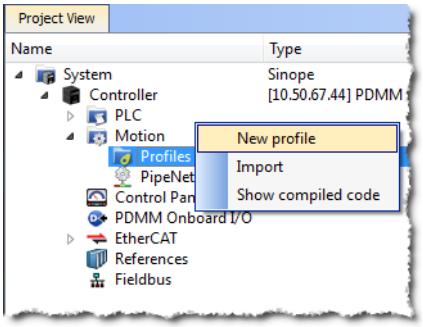


Figure 6-89: Cam - New Profile

2. A new profile named **Profile** is created with default parameters.
3. Right-click the new profile to rename it or change its properties.

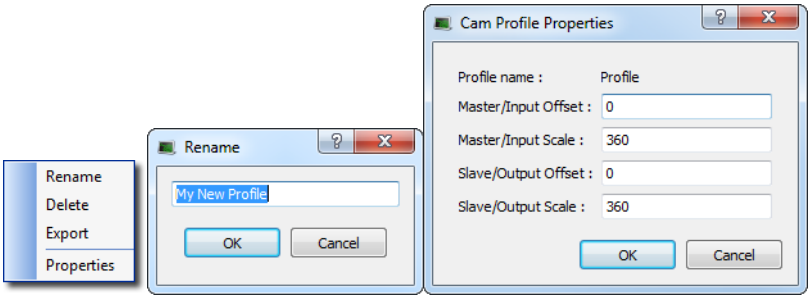
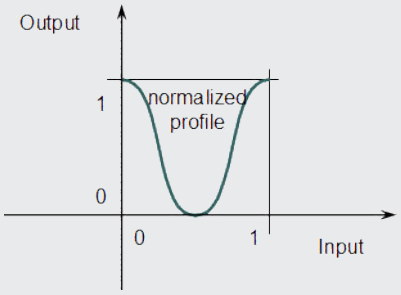
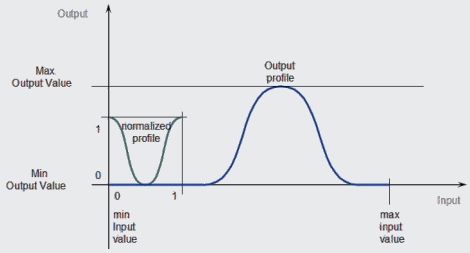


Figure 6-90: Cam - Define Profile Filename

4. Click the new profile to edit it.

6.11.11.1.1 CAM Profile Parameters

Field	Description
Profile name	The name of the Profile which is: • Shown in the Project Explorer • Used in the **Properties** of the CAM Pipe Block. • See "Use Cam Profiles" (➔ p. 250) for more information.

Field	Description
Master/Input Offset **Master/Input Scale** **Slave/Output Offset** **Slave/Output Scale**	Offsets and scales on the X and Y axes transform the normalized profile into the output profile as shown in these images:  **Figure 6-91:** Cam - Normalized Profile  **Figure 6-92:** Cam - Output Profile See "Four Parameters Transforming the CAM Profile" (➔ p. 248) for more information.

6.11.11.1.2.1 Four Parameters Transforming the CAM Profile

Four Parameters Transforming the CAM Profile

Master/Input offset

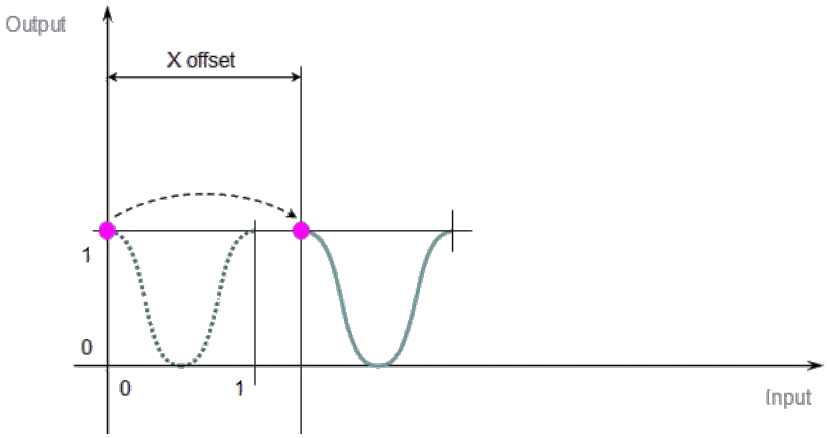


Figure 6-93: CAM Profile Transformation - Step 1

Master/Input scale

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

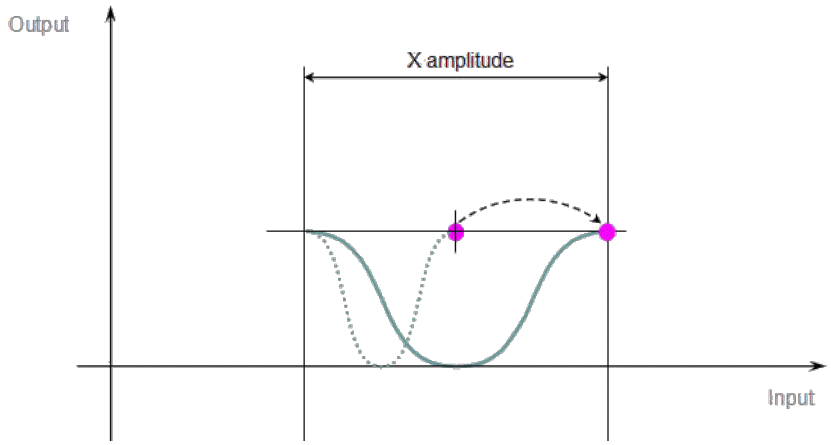


Figure 6-94: CAM Profile Transformation - Step 2

Slave/Output scale

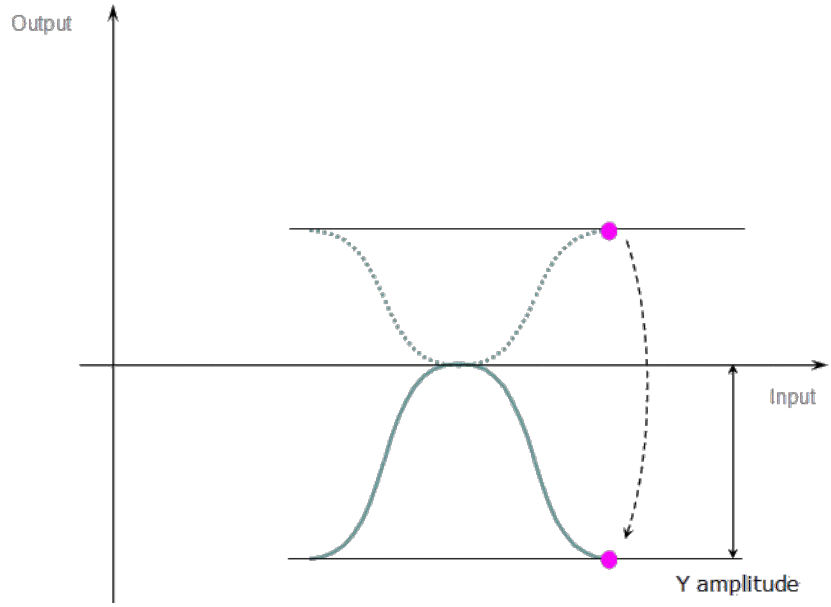


Figure 6-95: CAM Profile Transformation - Step 3

Slave/Output offset

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

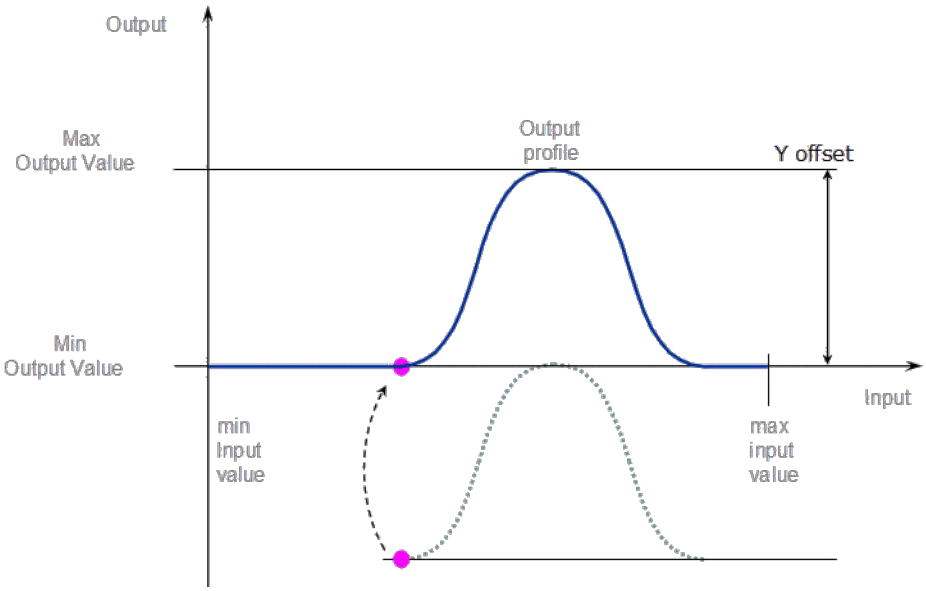


Figure 6-96: CAM Profile Transformation - Step 4

NOTE

When a CAM Profile property is changed, a dialog box indicates the progression of the operation.

See Also

- "Cam Profile Editor" (➔ p. 695)
- "Use Cam Profiles" (➔ p. 250)

6.11.11.2 Use Cam Profiles

Once defined (see "Create a CAM Profile" on page 247), you can associate a cam profile to a cam Pipe Block in the Pipe Network as follows:

1. Right-click on the cam Pipe Block and select **Properties** in the menu
2. In the **Parameters** tab, enter the profile's name

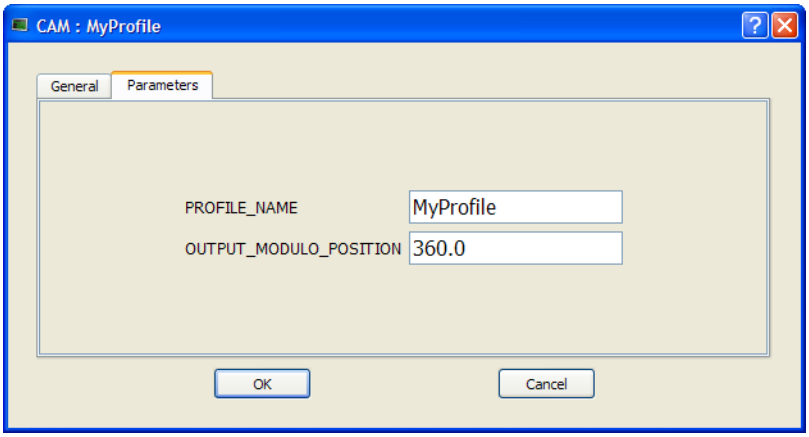


Figure 6-97: Cam - Associate Profile to a Pipeblock

NOTE

Separating the declaration of the cam Pipe Block from the cam profile provides the capability to prepare several different cam profiles and then apply one of them to the cam Pipe Block.

NOTE

If you change the profile's filename, do not forget to update the cam Pipe Block accordingly.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.11.3 CAM Profile Switching

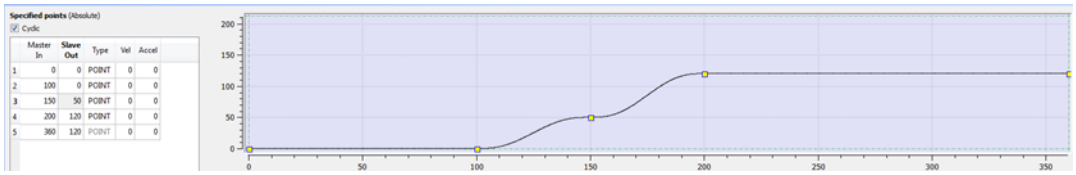
At pipe network start-up or when using the MLCamSwitch function block, a CAM block's output position is set to the output position of the initialized profile.

Assume

- This profile is switched on.
- The present pipe network position is:
  - 150 at the input of the CAM block.
  - 50 at the output of the CAM block.

Result

- As the input to the CAM block increases or decreases, the CAM output value increases or decreases respectively.
- It will go through the modulo without a position jump.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.11.12 Define Scheduling

6.11.12.1 Periodicity

The period of execution of a pipe is the time spent between two successive computations of set values for the same pipe.

The period of execution of a pipe is specified by the PERIOD parameter of the input Pipe Block. (Figure 6-98)

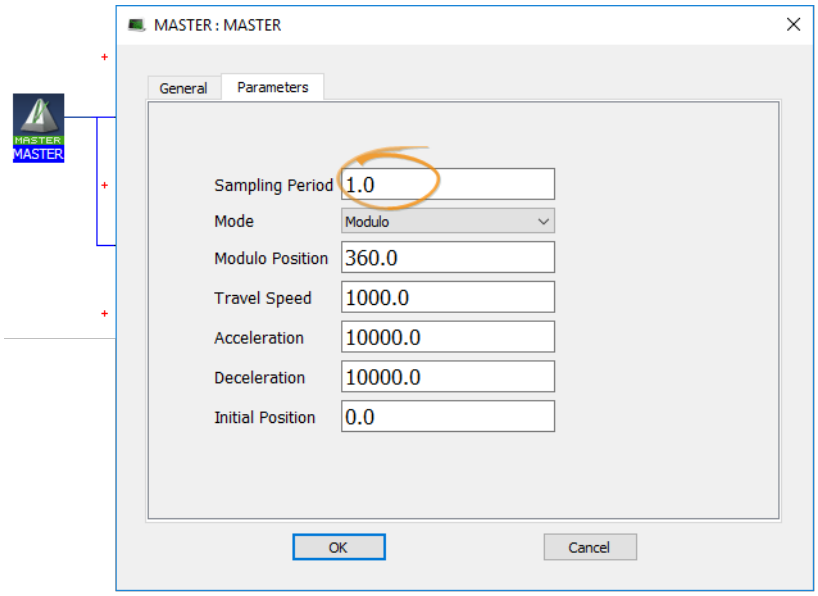


Figure 6-98: Set the Period of Execution

All the pipe values are computed independently of events and sequences execution.

6.11.12.2 Order of Program Execution

The order of program execution in a project is defined in the Cycle dialog box. (Figure 6-99)

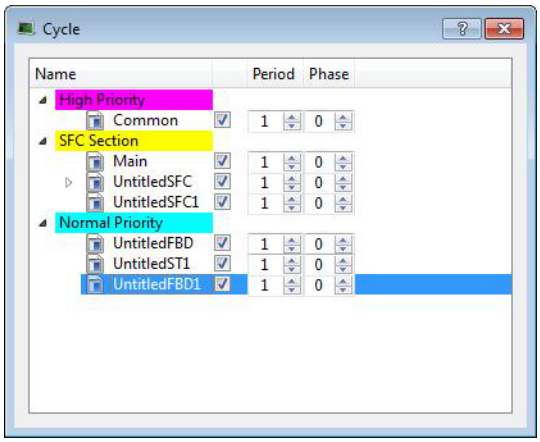


Figure 6-99: Cycle dialog box

The order of execution is always:

1. High Priority
2. SFC Section
3. Normal Priority

The order in each group is determined by the vertical ordering.

In Figure 6-99, the order of execution for Normal Priority programs is:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. UntitledFBD
2. UntitledST1
3. UntitledFBD1

See "Define the PLC Cycle" (➔ p. 253).

6.11.12.3 Define the PLC Cycle

The cycle specification defines the number of cycles between successive executions of the programs.

Procedure

1. In the Project Explorer, expand the PLC node.
2. Right-click on Programs and select **Cycle**. (Figure 6-100)

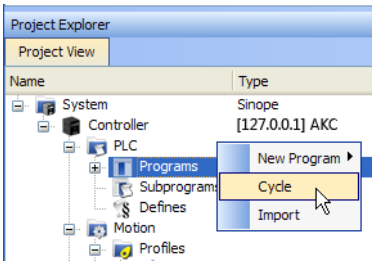


Figure 6-100: PLC > Programs > Cycle

The Cycle dialog box opens. (Figure 6-101)

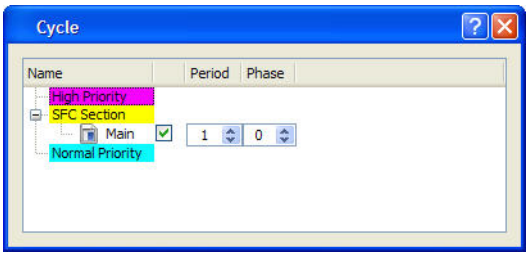


Figure 6-101: Cycle dialog box

The Cycle dialog box is used to:

- Change the Period and Phase parameters.
- Configure the programs priority into the Virtual Machine.

Column	Description
Name	List of PLC programs grouped together by priority level. • The priority defines the order of execution. • The SFC programs have a specific section as they must be grouped together. • High Priority means executed before SFC. • Low Priority means executed after SFC.
Check box	Enables or disables the execution of the corresponding program.
Period	Defines how many cycles are set between two executions of the program. • You can define various sampling periods for programs of the application. • The default period is 1 (i.e., the program is executed on each cycle). • Using a slower period for some programs is an easy way to give higher priority to some other programs.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Column	Description
Phase	Defines an offset that enables you to dispatch slow programs among few cycles. The goal of postponing the program execution is to reduce execution peak loads. **Examples** • A program with period=2 and Phase=1 is executed each **even** cycle. • A program with period=2 and Phase=0 is executed each **odd** cycle

In the **High** and **Normal** Priority sections, adjust the order of the programs with a drag-and-drop operation according to the expected sequence. In each section, the program on the top is executed first.

Select the program to set with a higher priority, then drag and drop it to the relevant priority level. (Figure 6-102)

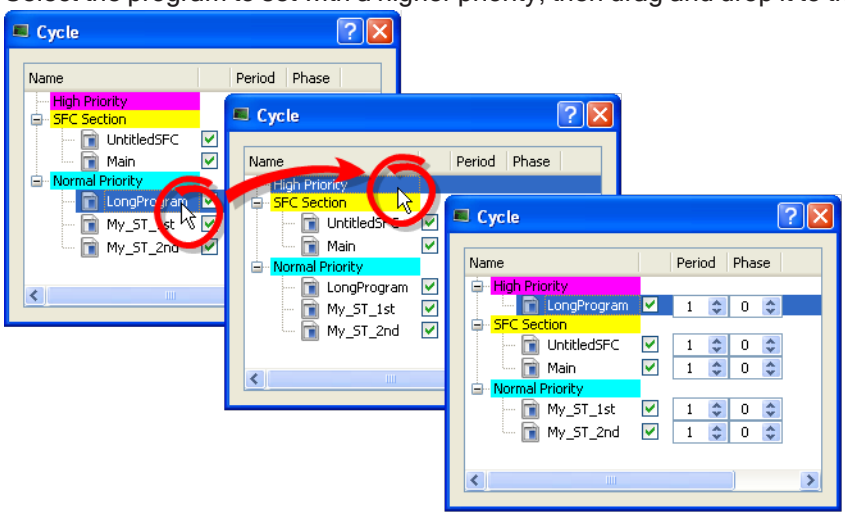


Figure 6-102: Change Priorities by Defining the Cycle

If all programs are with a Period set to 1, the KAS-IDE is more loaded. The choice of the Period for the programs gives you the possibility to distribute the load of the application.

See Also

- "Tasking Model / Scheduling" (→ p. 450)
- "Order of Program Execution" (→ p. 252)

6.11.12.3.1 Specify the Duration of a Cycle

This parameter is defined in "EtherCAT Master Settings tab" (→ p. 539).

6.11.12.3.2 Ensure Variables are Exported

Program Organization Units (POUs) containing variables (see "Map Variables to HMI" (→ p. 216)) must be compiled for the variable to be exported.

Example

In the Figure 6-106 images, a POU (UntitledST) has two variables: NewVar and NewVar1.

1. Only NewVar1 is set to be exported (Figure 6-103).
2. The POU, however, is not set to be executed in the Cycle dialog box (Figure 6-104).
3. This causes a compile error (Figure 6-105).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

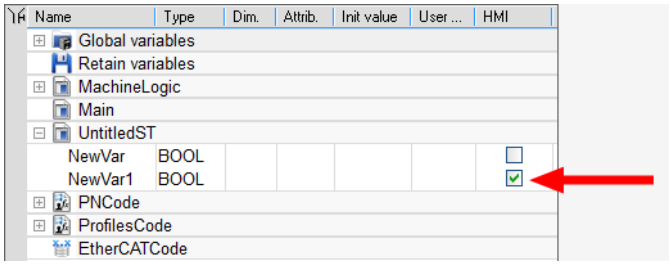


Figure 6-103: NewVar1 is set to be exported

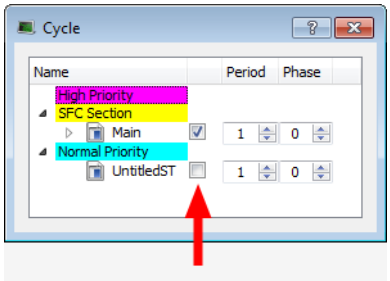


Figure 6-104: POU is not set to be executed

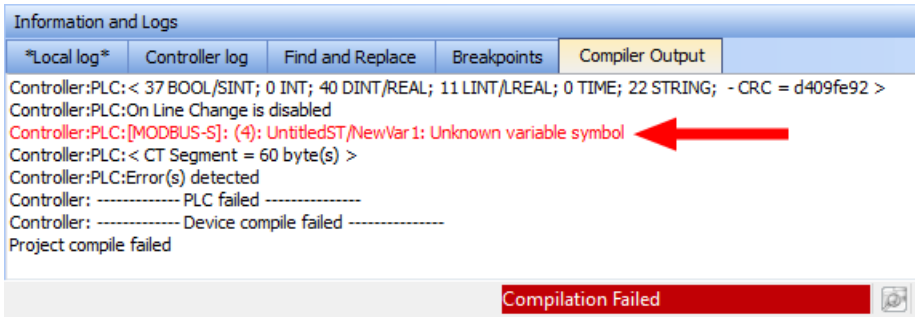


Figure 6-105: Compile error

Figure 6-106: Examples of a variable not being exported and the resulting compile error.

6.11.12.4 Parent-Child Relationships and Execution Order

The SFC Section of the Cycle dialog box is used to edit the period and phase values of parent and child programs.

Parent and child programs follow these rules:

- A parent program can be enabled or disabled.
  - If a parent is disabled the child will also be disabled.
  - A child program cannot be disabled.
- Parent programs are allowed to move across priorities.
  - Child programs follow the movement of a parent.
  - Child programs are not allowed to move independently.
- When a child program is created or imported, it inherits the enabled/disabled state of the parent program.
- The SFCs are executed at the set cycle period and phase.
  - All parent programs are executed first and then the children programs are executed in order.

Example

- In Figure 6-107, there are three parent programs, each with a number of child programs.
- All parents are executed, followed by the children, in order.

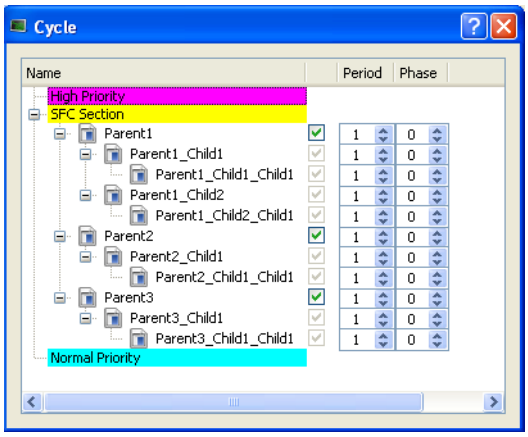


Figure 6-107: Cycle dialog box

6.11.12.4.1 Heirarchy

```
Parent1
  Parent2
    Parent3
      Parent1_Child1
        Parent1_Child1_Child1
          Parent1_Child2
            Parent1_Child2_Child1
              Parent2_Child1
                Parent2_Child1_Child1
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
└▷Parent3_Child1
  └▷Parent3_Child1_Child1
```

IMPORTANT

- **Parent SFCs should run faster than their children.**
- If this is not the case, the stop condition can be vague.
- When a child runs slower than its parent it does not stop when the parent stops, but at the child's next execution.
  - This means the parent could execute more, while the child is still running.

TIP

A child program is initiated at Phase 0 in respect to its parent.

6.12 Run the Project

This section explains how to build, download, and run your project.

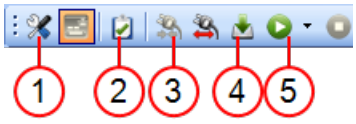


Figure 6-108: Device toolbar

The Steps in this table correspond to Figure 6-108.

Step	Description
1	Set the compilation options to run your project in Debug or Release modes. Choose if you want to activate the Online Changes.
2	Compile the application and see all the remaining warnings and errors. **NOTE** Before step 3, you need to start the KAS software (KAS Simulator or KAS Runtime) on the target device where you want to run your project.
3	Connect the KAS-IDE to the target device.
4	Download the Application compiled on the KAS-IDE to the target device.
5	Start / stop the device, and control your application with the script commands.

6.12.1 Set the Compilation Options

Click the  button to open the **Project Settings** dialog box. (Figure 6-109)

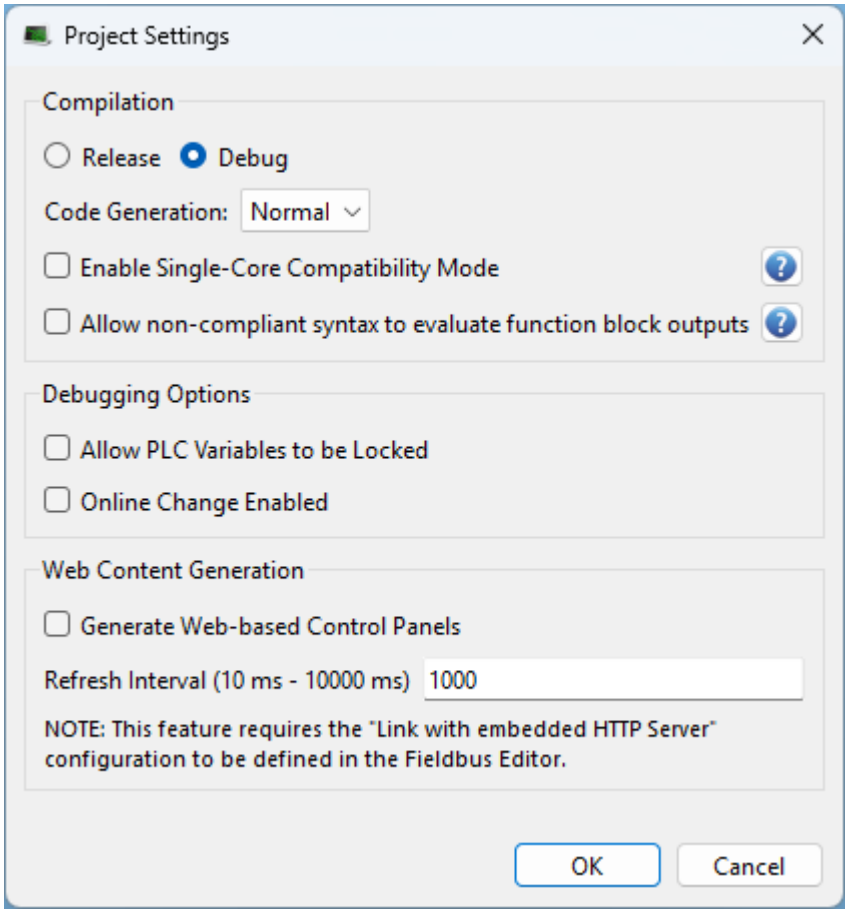


Figure 6-109: Project Settings dialog box

- Select the **Release** option to if step-by-step debugging is not required and to give highest performance to the application.
- Select the **Debug** option to use "Step-By-Step Debugging" (→ p. 268) during simulation or online testing.

When additional statements are incorporated in the code (e.g., trace outputs), you must select the **Debug** compiling mode so that they are taken into consideration (in RELEASE mode, those statements are not included).

See:

- "Conditional Compiling" (→ p. 259)
- "PC Online Change" (→ p. 861)

6.12.1.1 Code Generation

The code compilation options are: "Normal" (→ p. 258) and "Optimized" (→ p. 259).

6.12.1.1.1 Normal

- Normal code:
  - Allows the use of all PLC features in the KAS-IDE and Runtime.
  - Generation creates P-code and executes inside a virtual machine.
  - **Size Constraints**: The P-code is smaller than the same program translated to C-code.
- **Development and Debugging**: Normal code allows breakpoints, step through PLC code, and use online change.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Normal code is interpreted[1] so the interpreter can apply many additional runtime checks that would be harder to implement with Optimized code.

TIP

Using **Normal** code is recommended to take advantage of all KAS PLC features unless the controller is unable to meet performance requirements.

6.12.1.1.2 Optimized

NOTE

Starting with the KAS v5.00 release, the Optimized Code Generation option is supported in PCMM2G projects.

For optimal execution performance, the KAS compiler can generate native machine code for the PLC application by using an integrated a C-language compiler.

In the compilation mode, the process is in this order:

1. The KAS compiler generates P-Code for all PLC programs.
2. The P-Code is translated into C source files.
3. The source files are compiled using the controller-specific C tool chain included in the KAS-IDE.

When optimized code generation is enabled:

- These debugging features are still available:
  - Display of PLC variable values in the KAS-IDE.
  - Editing of PLC variable values by the KAS-IDE.
- Has these limitations:
  - "PC Online Change" (➔ p. 861) is not available.
  - WAIT and WAIT_TIME are not supported.

These features are disabled when optimized code generation is enabled:

- "Breakpoints" (➔ p. 270)
- Debugging (see "Test and Debug the Project" (➔ p. 268)).
- "Step-By-Step Debugging" (➔ p. 268)

6.12.1.2 Conditional Compiling

The compiler supports conditional compiling directives in ST, IL, FFLD, and FBD languages.

- Conditional compiling directives condition the inclusion of a part of the program in the generated code based on pragma.
- Conditional compiling is an easy way to manage several various machine configurations and options in one unique application project.
- Conditional compiling uses definitions as conditions.

This is the main syntax:

```
#ifdef CONDITION
    statementsYES...
#else
    statementsNO...
#endif
```

[1] (The code is read by the KAS Runtime engine that then determines the instructions to run)

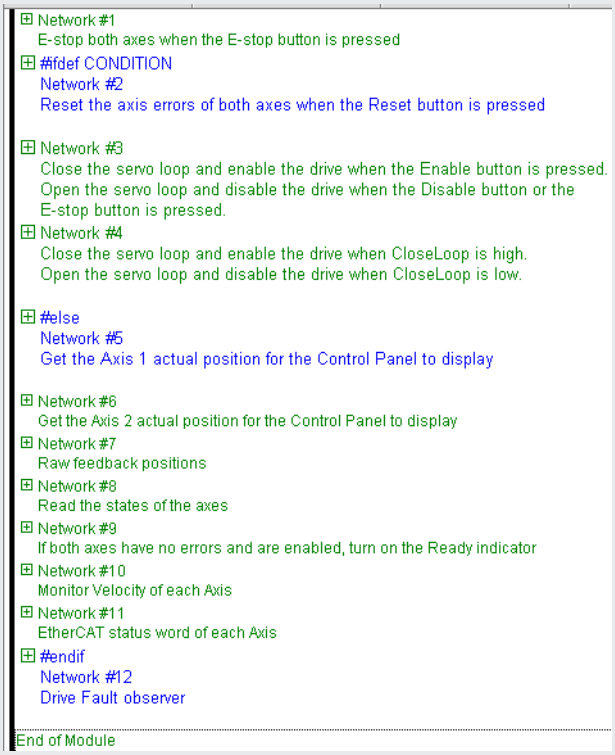
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- If CONDITION has been defined using `#define` syntax, then the **statementsYES** part is included in the code, else the **statementsNO** part is included.
  - The `#else` statement is optional.

TIP

"Intellisense" (→ p. 512) facilitates the reading by coloring in gray the part of the program which is not active.

6.12.1.2.1 Define Conditional Compiling Directives

Language	Description
FBD Language	Directives must be entered as the text of "FBD Network Breaks" (→ p. 491)
FFLD Language	Directives must be entered as a network **pragma** with the icon. In this example image: • If CONDITION has been defined using `#define` syntax, then • The networks **2 to 4** are included in the code, else • The networks **5 to 12** are included. 

IL Language
ST Language
	Directives must be entered alone on one line of text.
	`#ifdef DEF_A1_PeriodicAxis`
	`MLPhaSetPhase(PipeNetwork.PHASE1,DEF_A1_PosPeriod-A1_RefPos4);`
	`#else`
	`MLPhaSetPhase(PipeNetwork.PHASE1,DEF_A1_LinearPeriod -A1_RefPos4);`
	`#endif`

NOTE

Conditional compilation do not apply to actions in an SFC step.

The condition **__DEBUG** is automatically defined when the application is compiled in DEBUG mode.

This allows you to incorporate some additional statements (such as trace outputs) in your code that are not included in RELEASE mode.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
#ifdef      __DEBUG
     Printf('In debug mode', 0, 0, 0, 0);
#endif
```

See Also

also see "Run the Project" on page 257

6.12.2 Compile the Project

After creating all the elements of the project, it is ready to compile. The project must be compiled before it is simulated or downloaded to the target.

Use the Compile button ( ) on the toolbar or the **<Ctrl+B>** shortcut to compile the project.

The compiler reports messages in the **Information and Logs** toolbox.

See the "Compiler Output tab" (→ p. 136).

No other actions are possible when the compilation is in progress.

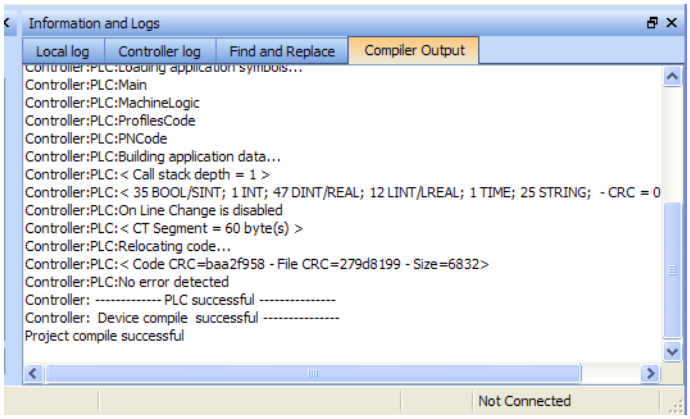


Figure 6-110: Information and Logs toolbox - **Compiler Output** tab

TIP

Errors are easily located using the information and logs window as shown below.

Double-click on an error in the list to open the program and jump directly to the relevant location in the editor.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

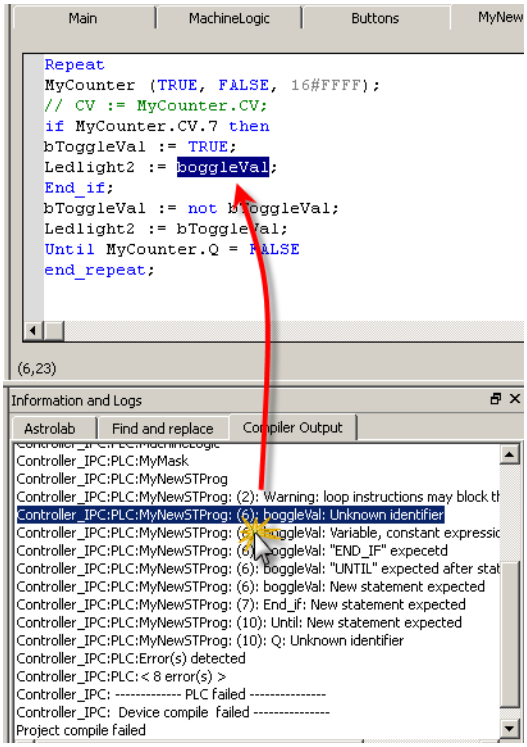


Figure 6-111: Example: Jump to location

To locate source code, use the **Find and replace** feature. See the "Find and Replace tab" (→ p. 129).

NOTE

In FFLD, when a function, function block or UDFB is not connected on the left, then it is ignored (removed at compiling time).
This case only applies for functions - **NOT** function blocks.

6.12.3 Start the KAS Simulator

Click the Simulator button in the toolbar to simulate your application.

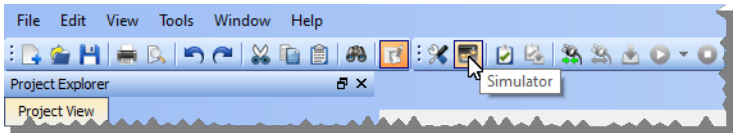


Figure 6-112: Simulator button

Once the program opens, start the simulation using the Control Panel.

See also see "Using the KAS Simulator" on page 586.

6.12.4 Connect to the Controller

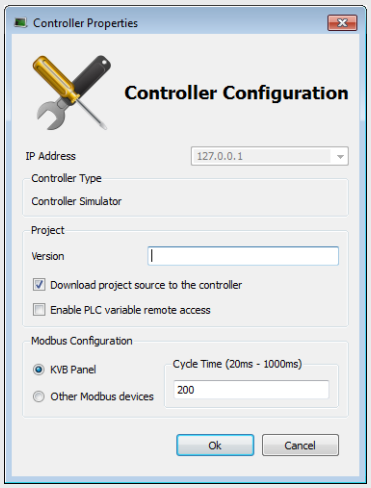
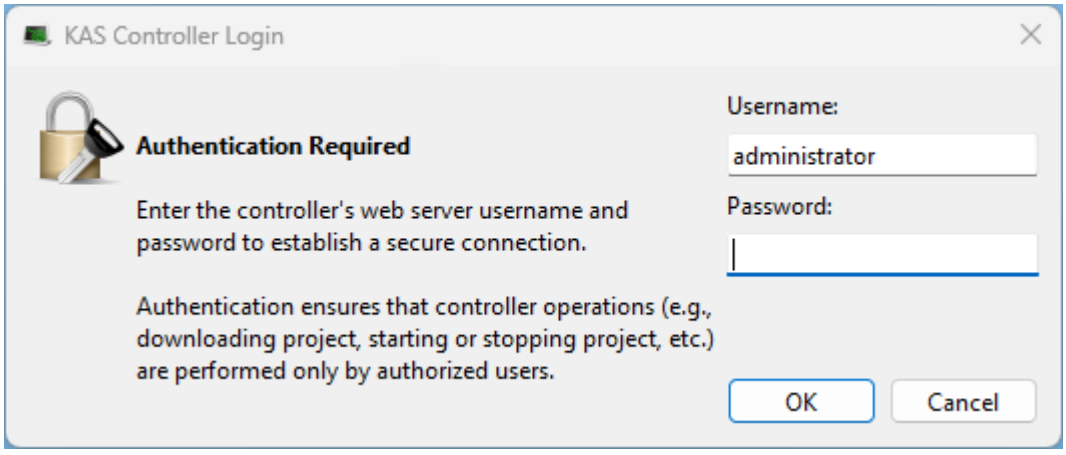
KAS provides all the commands for controlling the target in the **Device** toolbar (Figure 6-113):

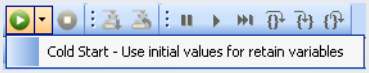


Figure 6-113: Device toolbar

Button	Description
	Show the controller communication properties dialog box (Project Settings).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Button	Description
	Change the controller IP address to connect with KAS Simulator. Example: In this simulated mode, the controller properties change and the Controller Type includes the IP address `127.0.0.1`. (Figure 6-114)  **Figure 6-114:** Controller Properties dialog box See "Using the KAS Simulator" (→ p. 586).
	Compile the project.
	Compile and download the project. This downloads the application and Project Source files.
	Establish a connection with the target controller. An **KAS Controller Login** dialog box opens. (Figure 6-115)  **Figure 6-115:** KAS Controller Login dialog box **NOTE** • The **KAS Controller Login** dialog box does not open when connecting to the KAS Simulator. • See Kollmorgen Automation Suite (KAS) Security. • See the "Status Bar" (→ p. 144) about statuses.
	Close connection with the target controller.

Button	Description
	Download the application to the targeted controller. • The application must **not** be running. • See "Download the Project" (→ p. 265).
	Start the application. It can be either: • **Warm Start (default mode)** • "Retain Variables" (→ p. 292) are loaded at the application startup. • They are **Not** re-initialized; whereas other variables are started with their initial values. • **Cold Start** • Use "Retain Variables" (→ p. 292) with their default values. • Cold starts occurs from time to time but are few. • To choose a Cold Start, click the Start button drop-down (Figure 6-116):  Cold Start - Use initial values for retain variables **Figure 6-116:** Start drop-down - Cold Start button
	Stop the application.

NOTE

If a previous download failed (due to the disconnected Ethernet or the controller is turned off), the application cannot be started.
Additionally:
- No version information is shown in the web server or status bar.
- The Start, Cold Start, and Download buttons are not available on the web server.

Click the  button to verify the Simulated device mode is active.

Click the **Connect Device** button () to establish the connection with the target controller.

NOTE

The device must be configured before connecting.
See "Configure the Controller" (→ p. 156).

6.12.4.1 Actions to Prevent Compatibility Issues

IMPORTANT

- The software versions of the KAS-IDE and the KAS Runtime MUST match to avoid compatibility issues.
- See "KAS-IDE Runtime Compatibility" (→ p. 148).

TIP

- The software versions of the KAS-IDE and the KAS Runtime are available in the local log messages (the level for this message is INFO).
- See "Information and Logs window" (→ p. 123).

When another KAS-IDE is already connected to the controller, a warning opens and the connection is discarded to prevent any conflict.

6.12.4.2 Application Status Bar

The status bar provides global information about the target and the name of the running application currently stored in the device.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

orange background means that the version of the application is different between the KAS-IDE and the target.

See "Status Bar" (➔ p. 144).

6.12.4.3 Message Window

Every log message has this information:

- Timestamp
- ID
- Message

NOTE

Once connected to the device, it is no longer possible to edit the PLC programs, unless Online Change is active.
See "Compile the Project" (➔ p. 261).

IMPORTANT

- Depending on the number of Kollmorgen AKD Series Servo Drives physically present in the EtherCAT network, the KAS-IDE might slow down when getting data.
- The KAS Runtime is **not concerned** with this limitation.

6.12.5 Download the Project

Click the **Compile and Download** ( ) button (**Ctrl + F7**) or the **Download** ( ) button.
The application is compiled and downloaded or just downloaded to the controller.

An alert appears if:

- An attempt to Compile and Download is made without a connection to a controller.
- There is not sufficient space on the controller for the application.
  - Clearing the "User Data tab" (➔ p. 631) is one method to create more space on the controller.
- In the **Controller Properties** dialog box, if the **Download Project Source in the controller** option is:
  - **Enabled** - the source code is copied to the controller.
    See "Configure the Controller" (➔ p. 156).
  - **Disabled** - and the controller has project source loaded, a message appears.
    - The message is different based on whether the current project and the project on the controller have the same name or not.

6.12.5.1 Project Name - Same Name

If the Project Name is the same, the message is (Figure 6-117):

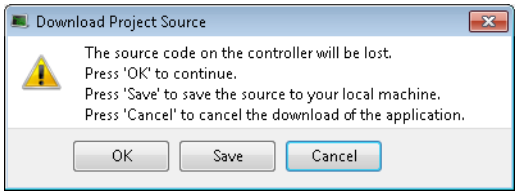


Figure 6-117: Download Project Source message - Project Name = Same Name

6.12.5.1.1 Actions

Click either:

- **OK**: Source on controller is deleted.
- **Save**: Source on controller is overwritten and is saved to "My Documents".
  - If that source file already exists in My Documents, a prompt appears to rename the source file, overwrite that file or not.
  - If the file is not renamed or overwritten, the download to the controller is canceled.
- **Cancel**: Download is stopped.

6.12.5.2 Project Name - Different Name

If the Project Name is different, the message is (Figure 6-118):

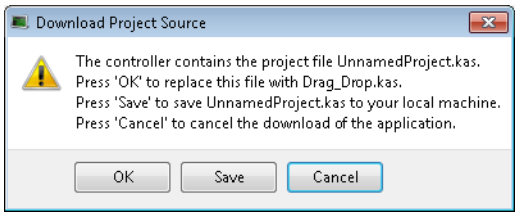


Figure 6-118: Download Project Source message - Project Name = Different Name

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.12.5.2.1 Actions

Click either:

- **OK**: Source on the controller is replaced.
- **Save**: Source on the controller is saved to **My Documents**.
- **Cancel**: Download is stopped.

NOTE

The versions between the KAS-IDE and the KAS Runtime must be the same to debug the application.
Example: To view the "Variable Animation" (→ p. 278) in the editors.

IMPORTANT

If the KAS-IDE version differs from the KAS Runtime after compiling an application, the function blocks defined in the KAS-IDE and those implemented in the virtual machine of the runtime can be different.
To prevent this potential mismatch, compile and download the application again.

TIP

The project's source code can be downloaded to store the project.
See the "File menu" (→ p. 455) about retrieving saved projects.

6.12.5.3 Application Status Bar

The tooltip of the application status bar gives more information about the application stored in the target: name of the project, name of the device, version of the application, its build number and date of compilation.

To view the tooltip, hold the mouse over the application status bar and wait for 1 or 2 seconds without moving the mouse.

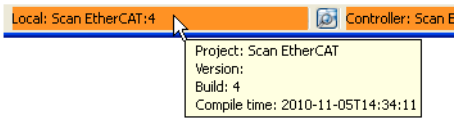


Figure 6-119: Device Tooltip displays Version

6.12.6 Device Control

6.12.6.1 Start / Stop the Device

With the KAS-IDE

Start / stop the device with the buttons and .

With the KAS Runtime

In the KAS Runtime menu, click the **start** / **stop** command.

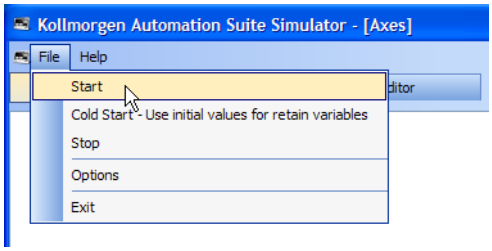


Figure 6-120: Start Device with the KAS Runtime

6.12.6.2 Log Window

The Log window displays all run-time messages issued by the device or by the KAS Simulator when testing the application.

- The log area of the KAS-IDE and the KAS Runtime Simulator are the same.
- It contains the log messages as described in "Information and Logs window" (→ p. 123).

6.13 Test and Debug the Project

During system validation it is essential that the KAS-IDE allows you to monitor the application program execution and to capture critical events and their data when they occur.

A Control Panel, designed with an internal editor, can be used to provide a basic interface.

See "Design the HMI with the Internal Editor" (→ p. 715).

6.13.1 Step-By-Step Debugging

To minimize risk, the KAS-IDE, in conjunction with the KAS Simulator, allows checking and validating the application program prior to deployment of the machine/system in production. This is achieved by capturing critical events in a **Step-By-Step Debugging** mode.

In addition to the cycle-by-cycle execution mode, the debugger has a collection of features for **Step-By-Step Debugging** in the source code of the application.

NOTE

Step-By-Step Debugging is only available if the project has been compiled with the **Debug** option.

This option is selected from the project compiling **PLC options** dialog box (accessed using the  button).

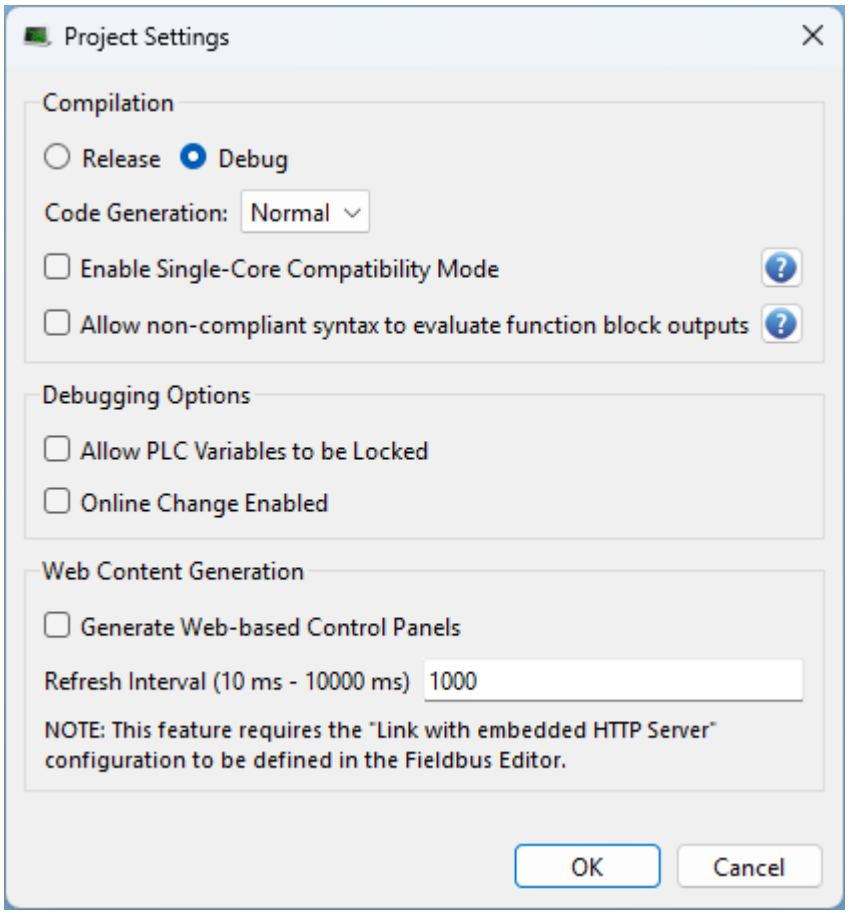


Figure 6-121: PLC Options - Debug Compiling Mode

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- An application compiled in **Debug** mode includes additional information for stepping.
  - This leads to a larger code size and reduced performance.
- When debugging is finished, it is recommended to compile the application in **Release** mode to give highest performance to the application.

Step-By-Step Debugging is available in:

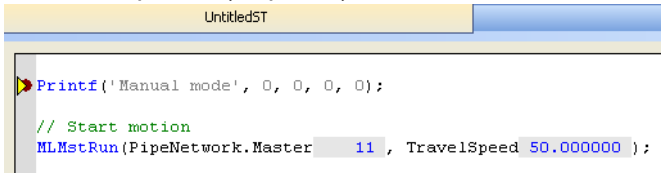
- FBD (a step is considered as a graphic symbol corresponding to an action).
- FFLD programs (a step is considered as a rung).
- ST and IL text programs (a step is considered as a statement).

IMPORTANT

Step-By-Step Debugging is **not** possible in SFC programs.
See "About SFC" (→ p. 272) for more information.

These are the options for entering the **Step-By-Step Debugging** mode:

- When a breakpoint is set in a program.
  - See "Breakpoints" (→ p. 270) for more information.



  - When you start your application and the breakpoint is reached, the execution stops at the specified location.
    You can run one step further in the program with the stepping commands.
- When the target is in cycle stepping mode (STOP).
  - You can step to the beginning of the first program.

TIP

- Pausing a program does not interrupt the current VM cycle.
- The current cycle finishes and execution is paused before the beginning of the next VM cycle.

These commands are available from the "Debug toolbar" (→ p. 470) for stepping:

Button	Description
	**Step Over** the next instruction: If the next instruction is a call of a function block or a sub-program, the execution passes over to the following instruction.
	**Step Into** the next instruction: • The next step is at the beginning of the called block. • If the next instruction is **not** a call of a function block or a sub-program, the **Step Into** behaves like the **Step Over**.
	**Step Out** the current block: • If the current stepping position is in a called function block or a sub-program, the execution continues up to the end of the current block. • Otherwise, the **Step Out** behaves like the **Step Over**.

These commands are available at any time:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Button	Description
	Execute a cycle step (from the current position up to the end of the last program).
	Restart the target in **Normal** execution mode (RUN).

6.13.2 Breakpoints

The step-by-step debugging feature is enabled by setting breakpoints in the source code of the application.

NOTE

This feature is only available in **DEBUG** mode.
See "Set the Compilation Options" (→ p. 257).

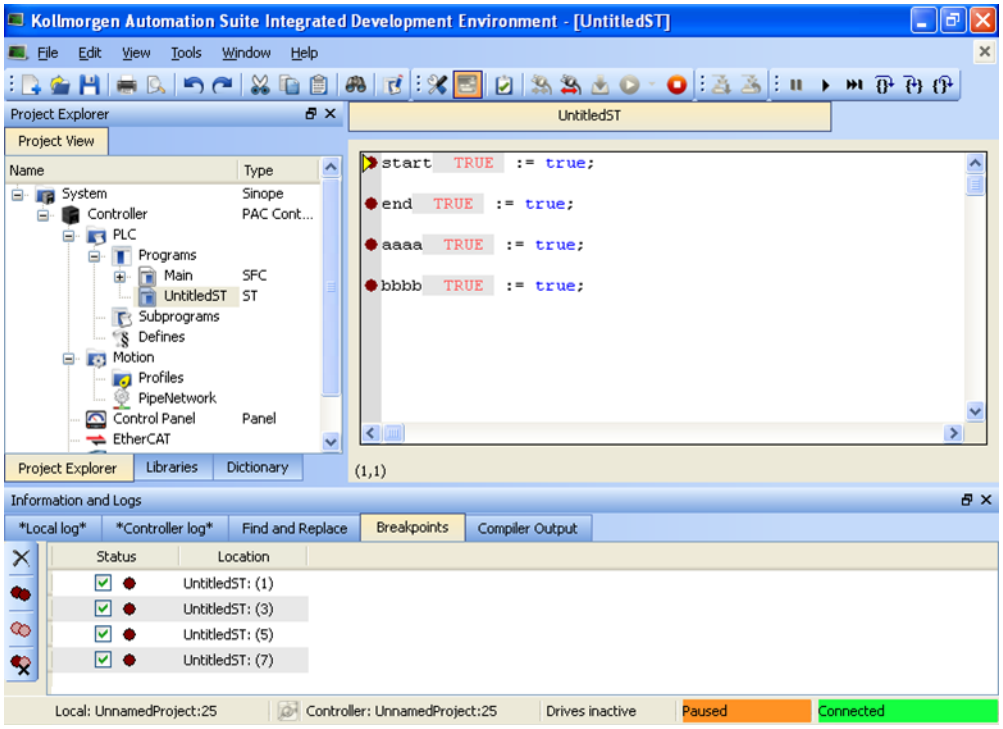


Figure 6-122: Example: Breakpoints tab

- Breakpoints are a marker set in code which, when reached, stops the code's execution at that location.
  - This lets you run one step further in the program with stepping commands.

  TIP

  - Pausing a program does not interrupt the current VM cycle.
  - The current cycle finishes and execution is paused before the beginning of the next VM cycle.
- Breakpoints are shown as a red circle (dark or light) in the left margin.
- Breakpoints may be active ( ) or inactive ( ).

6.13.2.1 Active or Inactive Breakpoints

- Breakpoints are active only when the KAS-IDE is connected to a target running an application that is compiled from the exact code displayed in the editor.
- Breakpoints are inactive if the KAS-IDE is:
  - Not connected to a target.
  - Connected but not running.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Connected to a different version of the code.
- Connected to the code but a modification has been made in Edit mode.

6.13.2.2 Breakpoint Usage

- Breakpoints are always applied to the target, based on their position in the editor.
  - If a breakpoint is moved in the editor, and you reconnect to a target, the breakpoint in the target is moved to the new position.
- A Breakpoint that has been hit has a yellow triangle (and) to indicate it has been reached in the code.
- Breakpoints are saved when saving KAS and are reloaded when loading a KAS application.
- Projects support a maximum of 16 breakpoints.
  - This includes both enabled and disabled breakpoints.

See:

- "Set, Remove, Enable, and Disable Breakpoints" (→ p. 271) about working with breakpoints.
- "Breakpoints tab" (→ p. 135) about information in the **Information and Logs** widget.

TIP

- Breakpoints can significantly increase the PLC cycle time execution.
- This is because the VM must evaluate the breakpoint condition at every cycle.

6.13.2.3 Online Change

- Online Change cannot be enabled when the KAS Runtime is paused due to a breakpoint.
  - Online Change can only be activated when the target is running. (Figure 6-123)



Figure 6-123: Online Change message

- Every breakpoint is activated if an Online Change is performed successfully.

NOTE

Breakpoints are not activated synchronously but in a reasonable time.

- All breakpoints become inactive when an Online Change is reverted.

6.13.3 Set, Remove, Enable, and Disable Breakpoints

This section discusses working with breakpoints within the editor.

See "Breakpoints tab" (→ p. 135) for information on the **Information and Logs** widget, including modifying breakpoints in bulk.

For background information see "Breakpoints" (→ p. 270).

6.13.3.1 Set Breakpoints

1. Open your program in the IEC 61131-3 Editor.
2. Click on the line (for ST/ IL) or diagram (for SFC [1] , FBD or FFLD) where you want to set the breakpoint.
3. Press **F9** or right-click and select **Add/Remove Breakpoint** from the menu.

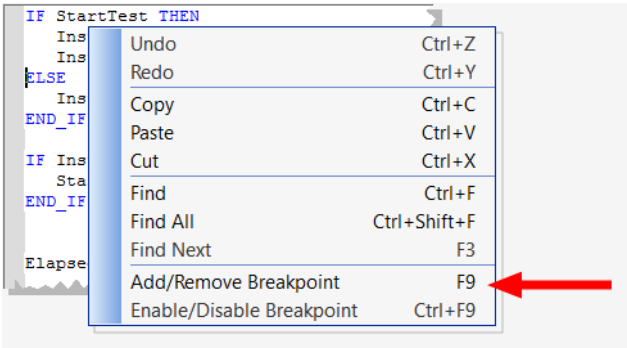


Figure 6-124: Setting Breakpoints

4. A Breakpoint circle is added to or removed from the left margin.
   The Breakpoint is set as either active ( ) or inactive ( ), based on the KAS-IDE's connectivity.

Even when you are **not** connected to the controller, breakpoints can be placed in programs, sub-programs or UDFBs.

NOTE

When you start your application, if the current position is not on a valid line for stepping, the breakpoint is automatically moved to the nearest valid position.

IMPORTANT

When you close the connection with the target, all the breakpoints are removed in the KAS Runtime.

6.13.3.1.1 About SFC

There are several things to note about breakpoints in SFC programs:

- In **SFC** programs, breakpoints can only be set on transitions (i.e., in First Level diagram), and not in steps or conditions. With a breakpoint set on a transition, you can debug cycle-by-cycle. Please remember that P1, N and P0 placeholders are designed to contain very simple code.

  TIP

  The recommended way to proceed for SFC sub-level programs is to rely on subprograms, where debugging is allowed.
- Breakpoints can be set and removed in SFC programs, they cannot be enabled and disabled.

6.13.3.2 Enable/Disable a Breakpoint

Enable a Breakpoint

- Right-click on an **inactive** breakpoint and select **Enable / Disable Breakpoint** (or <Ctrl + F9>).
- This is only available when the KAS-IDE and runtime are connected.

Disable a Breakpoint

- Right-click on an **active** breakpoint and select **Enable / Disable Breakpoint** (or <Ctrl + F9>).
- Selecting this option removes the breakpoint from the runtime.
- The breakpoint remains in the editor and can be changed to an inactive state.

6.13.4 Printf Function

[1] See limitation explained in paragraph below: **About SFC**

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Function - Used to display string in debug mode. (Figure 6-125)

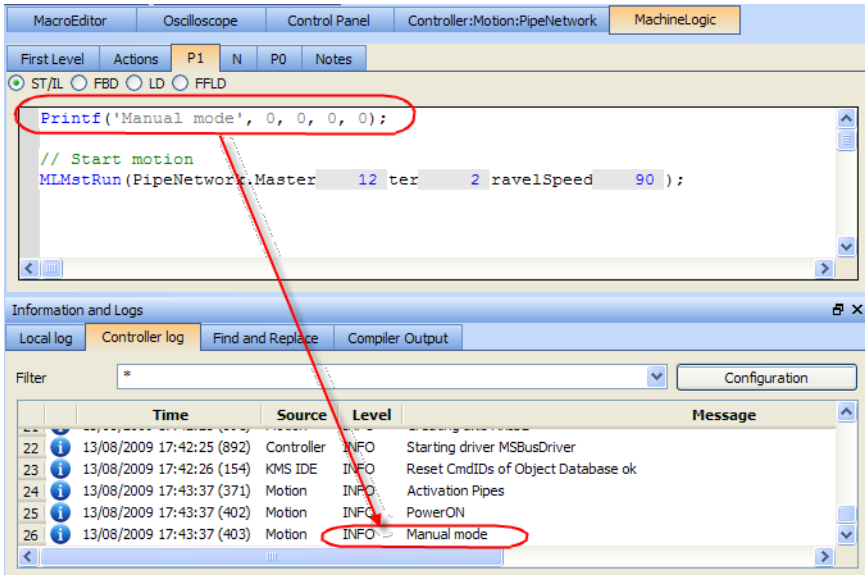


Figure 6-125: Printf

This is a good way to trace your SFC programs or use the PrintMessage.

6.13.4.1 Customize Output in the Log window

Raise warnings or errors icons.

- The first column in the log window has an information icon.
- This can be replaced with a warning or error icon:

```
@W
@E
```

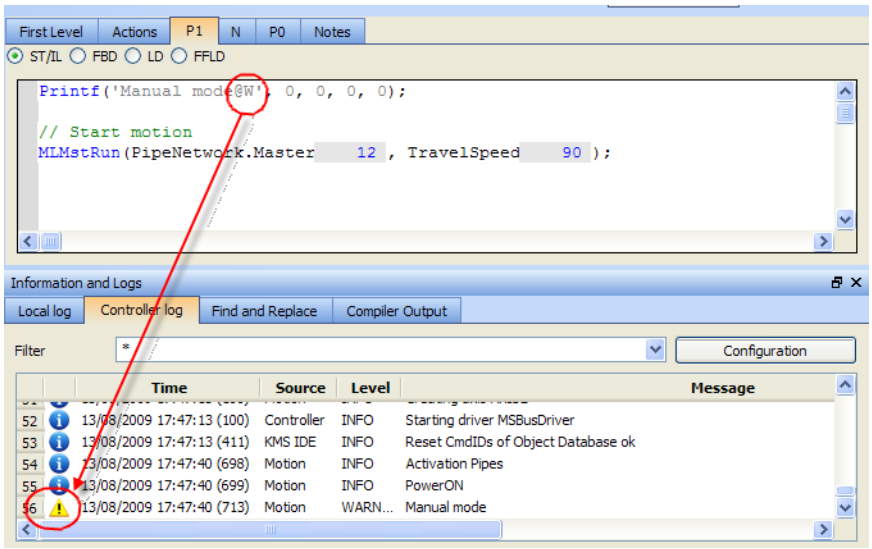


Figure 6-126: Customizing Output for Printf

6.13.5 Soft Oscilloscope Debugging

Access the values of the variables of the application is important during development and debugging.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Use "IEC 61131-3 Editor Debugging" (→ p. 281) to visualize and change variables via the Graphics HMI panel.
  - You can also access and change variables via the Variable Dictionary (see "Variable Monitoring" (→ p. 278)).
- However, these two methods can only access and change variables from the PLC part and not from the Motion part of the application.
  - The temporal evolution of the Motion variables is not be very intuitive.
  - The ideal tool to trace the Motion variables is a softscope.

Other typical areas for using the softscope are:

- Recording when an input is sensed in a cycle.
- Recording how much correction is being made in each cycle.
- Checking the settling time of an axis.

To open the Softscope, click the **Oscilloscope** command in the Tools menu.

See "Soft Oscilloscope" (→ p. 737) for description and usage.

6.13.5.1 Plug Motion Variables

NOTE

The Oscilloscope retrieves the variable values from the Motion Simulator.
You can only plug objects which exist in the Motion Simulator.
While the PLC variables exist all the time, the Motion objects are only created after the start of the application.

1. Verify the application is running.

   NOTE

   Your application **must be connected and running** to plug a variable to the channel.
2. Open the **PipeNetwork** of the controller in the Workspace.
3. Right-click on Gear1 and click the Map to Oscilloscope channel... option. (Figure 6-127)

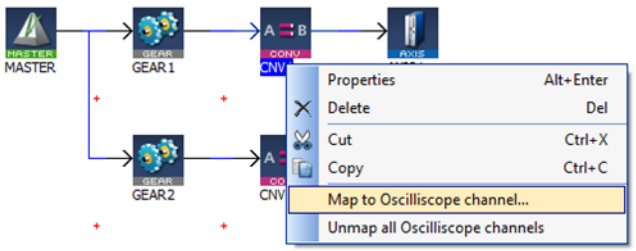


Figure 6-127: Selected menu option: **Map to Oscilloscope channel...**

The Plug on a channel dialog box opens.

4. Click the Channel drop-down arrow and select **1**.
5. Click the Data drop-down arrow and select the relevant option. (Figure 6-128)

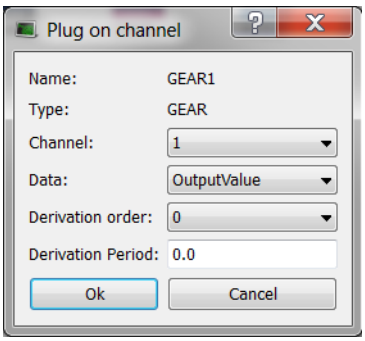


Figure 6-128: Plug on a channel dialog box - Parameters

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

The complete list of data are only visible when the application is running.

6.13.5.1.1 Usage Example with the Pipe Network

The Oscilloscope allows the recording and display of motion at points any where in a Pipe Network.

This example shows the difference between the input and output of the Phaser Pipe Block (called AngleAdvance).

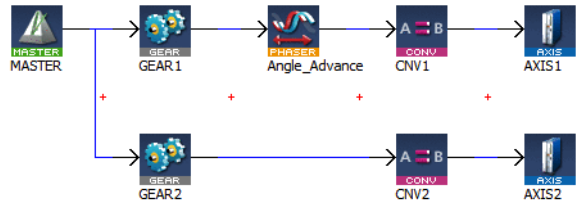


Figure 6-129: Example: Input / Output of Phaser Pipe Block

- The red line is the input.
- The green line is the output.
- The blue line shows when the phase advance change was active.

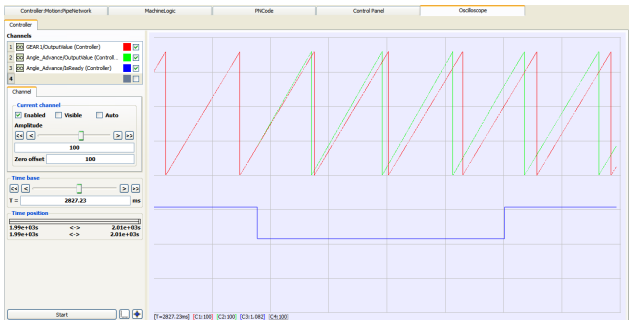


Figure 6-130: Example: Plugging a Pipe Block

See Also

- "Plug PLC Variables" (➔ p. 275)
- "Plugging Probes, Mapping Variables" (➔ p. 749)

6.13.5.2 Plug PLC Variables

1. In the Variable Dictionary, right-click on the variable **lastMachineSpeed** and click **Plug on channel...**. (Figure 6-131)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

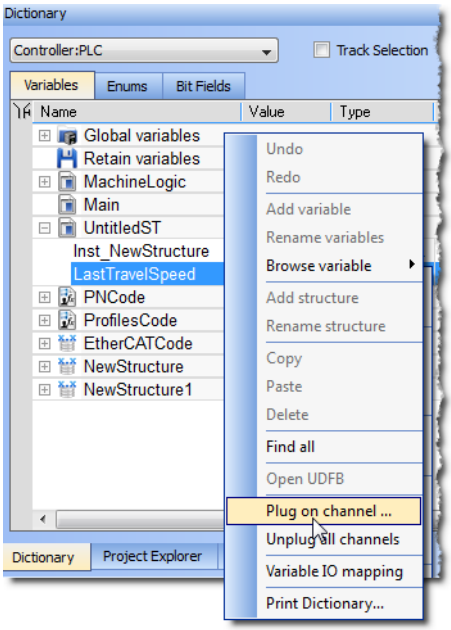


Figure 6-131: Selected menu option: **Plug on channel...**

The Plug on channel dialog box opens.

2. Set Channel to **2**(because channel 1 is already plugged). (Figure 6-132)

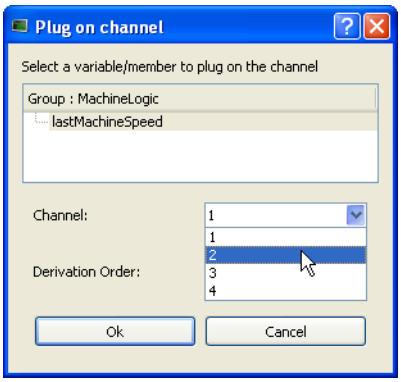


Figure 6-132: Plug on channel dialog box - PLC Variable Parameters

You can start the Oscilloscope now to see traces. (Figure 6-133)

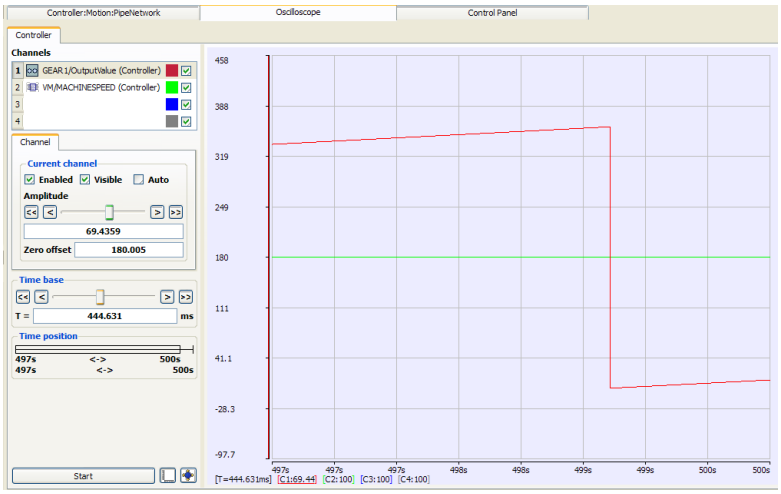


Figure 6-133: Oscilloscope Traces

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TIP

Easy probe plugging is assured since you do not need to unplug a probe from a channel before plugging a new probe into the same channel.

See Also

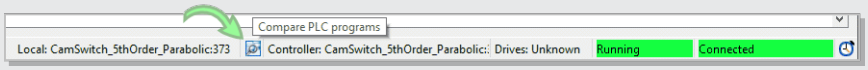
"Plug Motion Variables" (→ p. 274)

6.13.6 Compare PLC Programs

KAS provides a tool to show the differences between the Local Project and the project currently on the Controller or another project file.

TIP

- Use the Compare Projects to compare project files.
- The function for comparing the local project file to that on the controller is accessed from a button on the Status Bar, between the Local and Controller versions.
  - It is active when KAS is connected to a controller.



Once files have been selected to compare, a window opens listing the components of each project and any differences between them.

- How the projects differ is color coded.
  - Red item indicates where there is a mismatch.
  - Double-click to open an item. The **<<** button brings you back to the list.

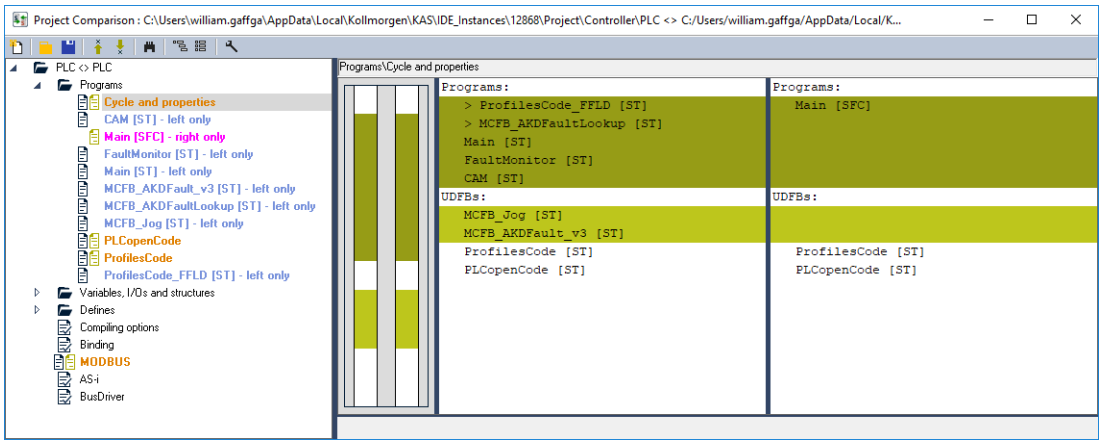
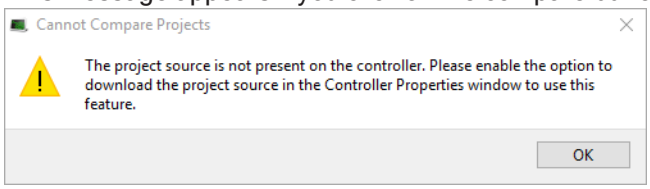


Figure 6-134: Listing the Differences

This message appears if you click on the compare button but the project is not present on the controller.



6.13.6.1 Limitations of FFLD Diagrams

If one or more components within a network differ, then the entire network is displayed in the diff view with a gray background.

Differences between individual components within the network are not specified in any way.

Example: Figure 6-135 has a comparison of two FFLD programs.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Network #1 differs, but the actual difference is not called out or highlighted.
  - The difference is the last element: a coil on the left diagram and an inverted coil on the right diagram.
- Network #2 is the same on the left and right, so it has a white background.

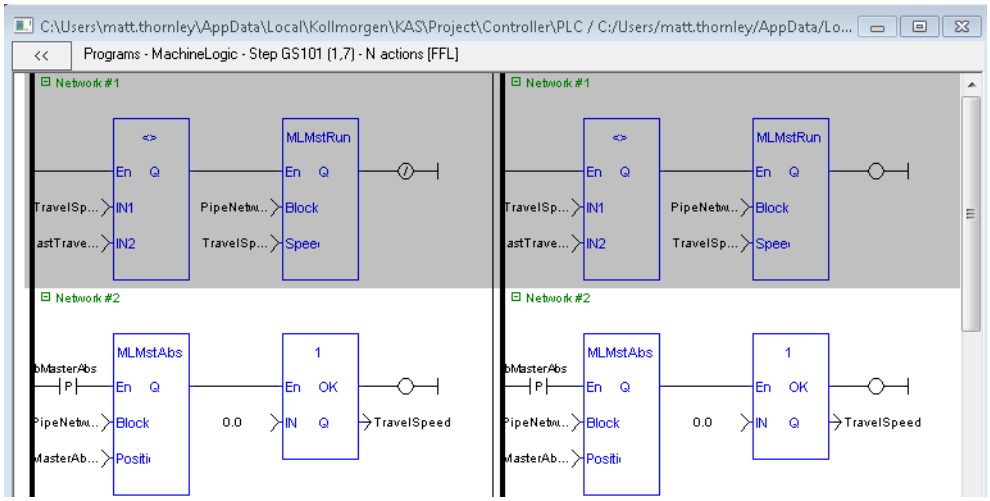


Figure 6-135: Example: Comparison of two FFLD programs

6.13.7 Variable Animation

When your application is running, all variables in the "IEC 61131-3 Editors" (➔ p. 481), Dictionary, and Watch Window are animated.

This means the value of each variable is displayed dynamically.

NOTE

- When the value of a variable is displayed, only the value computed at the end of the cycle is displayed.
- If the same variable is set in different programs, the animation in all those programs displays the same value for the variable, which corresponds to the latest program executed within the cycle.

6.13.7.1 Online Change

When Online Change is enabled, the animated values only take place when you are in Debug mode (and not edit).

6.13.7.1.1 Limitations

- The versions on the KAS-IDE and the KAS Runtime must be the same
- Animation does not apply to actions in an SFC step

6.13.7.2 Variable Monitoring

The Variable Dictionary contains all the IEC 61131-3 variables needed by the application. The variables are listed by categories corresponding to the declared programs, functions and function blocks.

When your application is running: (Figure 6-136)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- All variables in the Dictionary are animated [1] with real-time values displayed in the **Value** column (see call out (1)).
- A specific column is used to indicate the initial values of all variables (2).

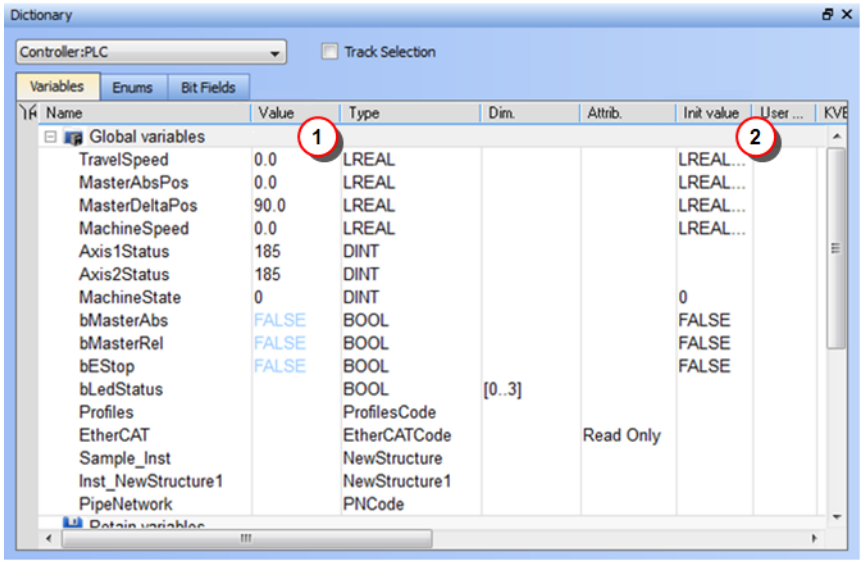


Figure 6-136: Variable Dictionary

6.13.7.2.1 Monitoring UDFBs

The real time values for UDFBs can be shown in the workspace while the program is running.

- Double clicking on a UDFB in the Project View's Subprograms list opens the UDFB in the workspace.
- A list of the UDFB's instances are listed if it is instantiated more than once or declared inside of a structure. (Figure 6-137)

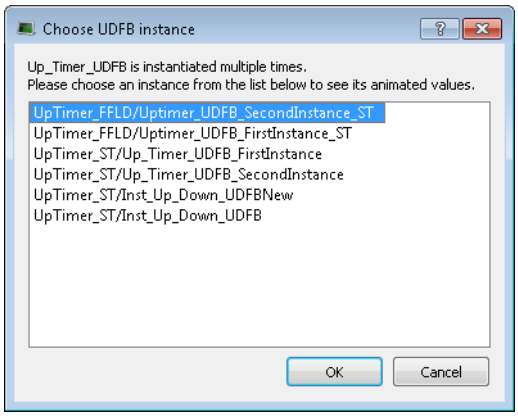


Figure 6-137: Choose UDFB instance dialog box

- If the UDFB is open in the Workspace before running the program, it does not automatically animate because there may be more than once instance of the UDFB.
- To start the animation double-click the entry in the Project View.
- If there is only one instance, it will open; if there are multiple instances the list is presented.

6.13.7.2.2 Force a Variable

---

[1] To better track variables and expressions of the PLC programs in Test mode, the KAS-IDE dynamically computes their value along with the program execution and display the result in gray boxes beside their usage in the instruction lines of the IEC 61131-3 editor.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

At runtime, all variables in the table widget are animated [1] with real-time values.

At runtime, double-click on the value of the variable in the list or press the **ENTER** key when it is selected.

A popup window appears and allows you to "Force" (→ p. 280), "Lock" (→ p. 280), or "Unlock" (→ p. 281) a variable.

6.13.7.2.3.1 Force

- Change the value of the selected variable.
- Depending on the variable type, you have the possibility to define its value either in the text field or with the check boxes. (Figure 6-138)

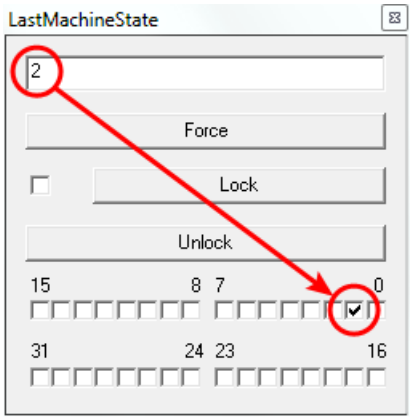


Figure 6-138: Forcing a variable

6.13.7.2.4.2 Lock

- Variables, member variables of a structure and variables in an array may be locked.
- When a variable is locked, its value is no longer changed by the runtime.
  - You can then force its value from the debugger independently from the runtime operations.
- All variables can be locked and forced at runtime.

NOTE

The value of a locked variable is displayed with square brackets. (Figure 6-139)

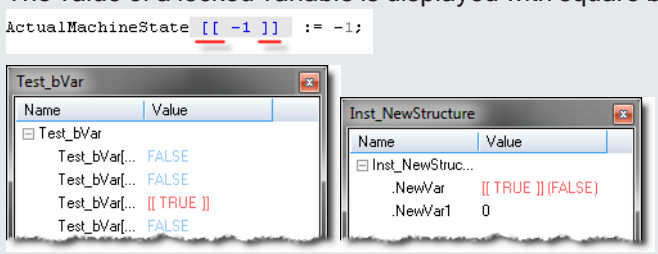


Figure 6-139: Example: Lock variable

The variable locking feature can be enabled or disabled via the PLC Options device toolbar button.

If enabled, the Lock and Unlock buttons are accessible (Figure 6-141):

[1] To better track variables and expressions of the PLC programs in test mode, the KAS-IDE dynamically computes their value along with the execution and displays the result

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

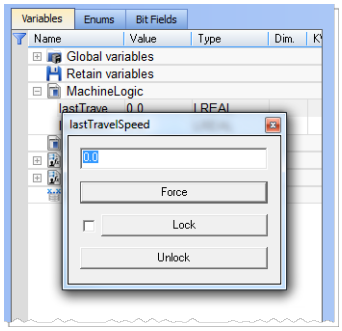


Figure 6-140: Enabled locking

If disabled (default), the Lock and Unlock buttons are not accessible (Figure 6-140):

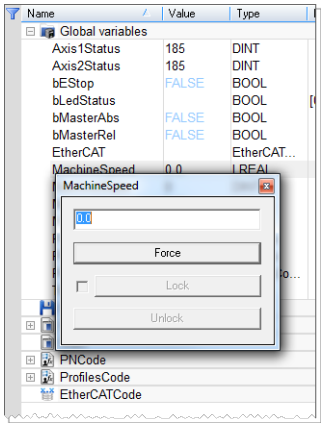


Figure 6-141: Disabled locking

IMPORTANT

If PLC variable locking is enabled, the controller Runtime requires an additional 3% to 10% CPU processing power to manage the PLC variable locking.
For the best controller performance, disable PLC variable locking.

6.13.7.2.5.3 Unlock

- Remove the lock on a variable so that it can be changed again by the runtime.

6.13.7.3 IEC 61131-3 Editor Debugging

In Test mode (Online or Simulation), all editors are animated [1] with real-time values of the edited objects:

- Values of variables, contacts and coils are displayed in FBD diagrams.
  - Double-click a variable name to force or lock the variable.
- Values of variables, contacts and coils are displayed in FFLD diagrams.
  - Double-click a variable name to force or lock the variable.
- Step activities (tokens) are displayed in the SFC editor.
- In the text (ST or IL) editor, place the mouse cursor on a variable name to display its real-time value in a tooltip.
  - Double-click the variable name with the **Shift** key pressed to force or lock the variable.

[1] To better track variables and expressions of the PLC programs in Test mode, the KAS-IDE dynamically computes their value along with the program execution and display the result in gray boxes beside their usage in the instruction lines of the IEC 61131-3 editor.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
Repeat
MyCounter (TRUE, FALSE, 16#FFFF);
// CV := MyCounter.CV;
if MyCounter.CV.7 FALSE then
bToggleVal TRUE := TRUE;
Ledlight2 TRUE := bToggleVal TRUE ;
End_if;
bToggleVal TRUE := not bToggleVal TRUE ;
Ledlight2 TRUE := bToggleVal TRUE ;
Until MyCounter.Q FALSE = FALSE
end_repeat;
```

Figure 6-142: Animation in Editors

See "Force a Variable" (→ p. 279).

6.13.7.4 Pipe Network Editor Debugging

See "Plug Motion Variables" (→ p. 274).

6.14 Manage a Project

- The **New** command in the File menu uses a wizard to help you to define the project.
- The **Open...** command opens a window to let you navigate your system and retrieve previous projects.
- The **Save** command saves your entire project.
- The **Save As...** command allows you to save your project with a custom name and location.

IMPORTANT

Choose a safe folder for your project.
Never select the Installation repository.

The **Close** command prompts you to save first if some modifications have not been saved.

NOTE

When a project is already open, and you try to create or open another one, KAS-IDE shows a message asking you to save the project before it is closed.

The **Print...** command is used to create documentation containing editor's programs or diagrams.

See either "Menus Overview" (→ p. 455) or "Toolbar Overview" (→ p. 469).

TIP

Use the **Recent Projects** command in the File menu to open the last four projects.

When editing your project, the KAS-IDE has these restrictions:

- You cannot work with several projects in parallel.
  - Multiple instances of KAS-IDE can be used as a work-around.
- Modifications that impact the project structure cannot be reversed with the **Undo** command.
  - Use the **Save As** command first to create a backup.
- No guarantee is provided by the KAS-IDE with respect to the project file's integrity.
  - This means that, if you modify your data from outside the KAS-IDE, you can spoiled your project.

6.14.1 Use a Version Control System

To ensure integrity of your project files, you have to rely on tools to control versions.

Generally, such tools also have facilities for:

- Backup management.
- Multi-users or multi-site development.

6.14.2 Print

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.14.2.1 Printable Elements

The elements that you can print are:

- All PLC programs (see "PLC" (→ p. 88) node in the Project Explorer).
- Individual programs.
- Level 2 SFC.
- Level 2 SFC of single transition/state.
- The Pipe Network editor.
- The Dictionary.

You can either print one specific program or all the project (PLC, Motion, Dictionary variables).

6.14.2.2 Page Setup

This dialog enables you to define the following settings:

Page Setup tab

- Orientation:
  - Used to choose between portrait or landscape.
  - Because the orientation can be set in both the page setup and the printer driver, it is recommended to have both settings synchronized.
- Scaling:
  - Select the **Fit to** option to fit on the specified number of pages.
  - You have to enter one of the two values (either Wide or Tall) and the other are filled in automatically to keep a 1:1 aspect ratio of the print.

NOTE

These settings are not applicable when printing a project.

6.14.2.2.1 Margin and Header/Footer tab

If you specify new margins or header/footer for a program, it affects the entire project when printed.

- Special items can be inserted into the header/footer string as **{@item}**.
  - They are converted to the correct format on printing or for print preview.
- About the **Filename** field:
  - If an SFC level 2 program is being printed, the filename contains the SFC program name, Step or Transition number, and the action tab name (e.g., Main, GS3, P1).

NOTE

All the settings defined in the Page Setup are saved in the project and are applied to each printed program.

This dialog box contains two buttons:

- **Print...** displays the Printer dialog box.
- **Print Preview** displays a printout on the screen to see how it looks like before printing it.

6.14.2.3 Print

This dialog is used to:

- Set the output (a printer, a PDF).
- Set the output preferences to set-up the printer options.
- Look for a printer on the network.
- Set the number of copies.
- Set the page area to be printed.
- Start the print.

To print an SFC level 2 program, open it in the SFC editor and click the Print icon  <Ctrl + P>.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6.14.2.4 Print Preview

This dialog box is used to view a printout on the screen before printing.

NOTE

Print preview limits the number of pages to display to the first 30 pages.

6.14.2.5 Print the Project

A Print Project dialog lists all the items that are printable.

Select those items to include in the output and click **OK**.

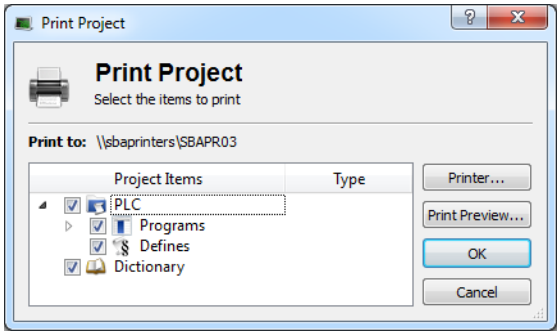


Figure 6-143: Print Project

NOTE

Selecting an SFC program prints the SFC chart as well as SFC level 2 programs.
Automatic scaling is applied for best readability.

6.14.3 Use the Reference Folder

Using the Reference item, you can link as many files as you want to your project.

1. Right-click on the Reference item and select the **Insert Reference** command.

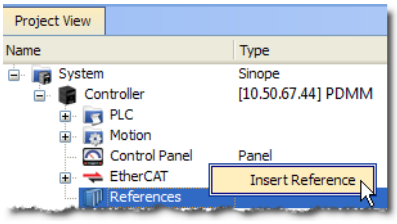


Figure 6-144: Inserting a Reference

2. Define the **Name** and choose a valid **URL**.

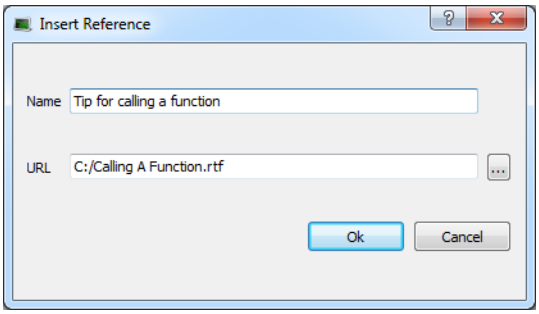


Figure 6-145: Defining the Reference

3. Double-click the new reference to open it in the workspace.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

- You can link files that are on your local machine (or to a server shared with a mapped drive) and of these types:
  - PDF, doc, xls, drawings, etc.
- You must ensure the link is not broken if you want the KAS-IDE to open it correctly.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7 Understanding KAS

This section provides information about the most important concepts needed to use KAS.

IMPORTANT

To take full advantage of KAS functions, a basic understanding of automation (programming languages and motion control) is required.

7.1 IEC 61131-3

To create programs for the implementation of the PLC part of your application, the KAS-IDE complies with IEC 61131-3.

- This standard defines programming languages for programmable control systems.
- The KAS-IDE implements this standard to provide you with well-defined and well-known programming languages.

7.1.1 Data Types

Data types are defined within the common elements of IEC 61131-3.

Why Data Typing?

Data typing is implemented to define the type of any parameter used. This helps prevent errors early on in the programming phase (e.g., avoids dividing a Date by an Integer).

When you have defined whether the data is a string, a date, an integer or a 16-bit Boolean input, there is no longer any confusion, nor any conflict between different people using the textual representation (i.e., the name of the variable).

7.1.1.1 Different Data Types

- **Common Data Types**: Boolean, Byte, Date, Integer, Real, String, Time_of_Day, Word.
- **Derived Data Types**: Define personal data types based on the Common data types.
  - Example: Define an analog input channel as a data type and re-use it.

7.1.1.2 List of Data Types

Type	Prefix	Description	Values
BOOL		Boolean (bit)	• FALSE, TRUE • Stored in 1 byte.
BYTE		Same as "USINT" (→ p. 287).	
DINT		Signed double precision integer in 32-bits.	-2147483648 to 2147483647
DWORD		Same as "UDINT" (→ p. 287).	
INT	INT#	Signed integer in 16-bits.	-32768 to +32767
LINT	LINT#	Long signed integer in 64-bits.	
LREAL ‡	LREAL#	Double precision floating point stored in 64-bits.	• -1.7E308 to 1.7E308 • 14 to 15 significant digits of accuracy
LWORD		Same as "ULINT" (→ p. 287).	
REAL ‡	REAL# ‡‡	Single precision floating point stored in 32-bits.	• -3.4E38 to 3.4E38 • 6 to 7 significant digits of accuracy.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Type	Prefix	Description	Values
SINT	SINT#	Small signed integer in 8-bits.	-128 to +127
STRING		• Variable length string with declared maximum length. • Each character is store on 1 byte (i.e., on 8-bits).	Maximum length cannot exceed 255 characters.
TIME	T# or TIME#	Time data type is used to specify a time variable. • Accuracy is 1ms.	0ms - 24hr
UDINT	UDINT#	Unsigned integer in 32-bits.	0 to +4294967295
UINT	UINT#	Unsigned integer in 16-bits.	0 to 65535
ULINT	ULINT#	Long unsigned integer in 64-bits.	
USINT	USINT#	Small unsigned integer in 8-bits.	0 to 255
WORD		Same as "UINT" (➔ p. 287).	
WSTRING		• Do not use! • The WSTRING is **not** supported.	

NOTE

- ‡ **REAL** variables are limited to 6 digits of accuracy.
  - To achieve greater accuracy, a longer mantissa may be specified by prefixing **LREAL** with #.
- Example: To achieve an accuracy of 20 digits for the value of Pi, rather than what **REAL** provides (3.14159), set the type to `LREAL#3.14159265358979238`.

IMPORTANT

‡‡ **REAL** is restrictive, but because it is the default, it is recommended to explicitly declare real constants with the **LREAL#** prefix.

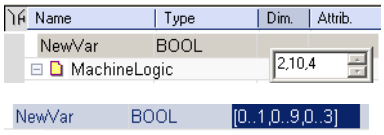
TIP

Use **2#**, **8#**, or **16#** prefixes to specify an integer in binary, octal or hexadecimal basis, respectively.

7.1.1.3 Arrays

You can declare arrays for internal variables by specifying dimension(s).

To declare an array, enter the number of elements in the **Dim.** column on the Variables tab.



For a multi-dimensional array (arrays have a maximum of three dimensions), enter the number of elements for each dimension separated by commas:

- Using **2,10,4** as the 3D array:
  - The first dimension has 2 elements.
  - The second dimension has 10 elements.
  - The third dimension has 4 elements.

Arrays are accessed by variable indices.

```
myArray[myIndex1, myIndex2]
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

IMPORTANT

- All indexes are 0 based.
- Example: In a single dimension array, the first element is always identified by `ArrayName`**[0]**.

7.1.1.4 Bitfields

A bitfield is a user-defined data type that combines multiple Boolean or integer values into a single variable.

- Bit Fields are used to define custom variable types and values.
- A bitfield packs multiple pieces of data together in one variable.
- Each field represents an independent piece of information.
  - These fields can be referenced individually within the application.
- Bitfields provide a compact and organized way to group related status and control information while maintaining easy access to individual fields.

Example

A bitfield type named AxisStatus may contain these fields:

Initialized
Enabled
Moving
Faulted

See the Bit Fields tab about creating bit fields and examples of their use.

7.1.1.5 Enumerations

An enumeration (enum) is a user-defined data type that contains a predefined set of named values.

- An enum variable can hold only one value from the defined list at a time.
- Enumerations are useful when a variable should only represent one value from a known set of states, modes, or selections.

Example

Enumeration can be used to represent machine states such as:

Idle
Homing
Running
Faulted

- Using enumerations improves program readability and maintainability by replacing numerical values with meaningful names.
- This makes application logic easier to understand, debug, and maintain.
- See the Enums tab about creating enumerations and examples of their use.

7.1.1.6 Structure

A structure is a complex data type defined as a set of members.

- Members of a structure can have various data types.
  - A member of a structure can have dimensions or can be an instance of another structure.
- When a structure is defined, it can be used like other data types to declare variables.
- Members of a structure can have an initial value.
  - Corresponding members of all declared variables having this structure type are initialized with the initial value of the member.
- To specify a member of a structured variable in PLC languages, use this notation:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
VariableName.MemberName
```

7.1.1.6.1 Limitation

If a member of a structure is an instance of another structure, the nested structure must be declared **BEFORE** in the list.

7.1.2 Variables

The scopes of the variables are normally limited to the organizational unit in which they are declared, e.g., local.

- This means that their names can be re-used in other parts without any conflict, eliminating another source of errors, e.g. the scratchpad.
- If the variables have global scope, they must be declared as such.
- Parameters can be assigned an initial value to have the right setting at start up and cold restart.

7.1.2.1 Retain Variables

A retain variable is a PLC variable which:

- Is non-volatile and stored in non-volatile controller memory (PCMM2G).
  - When using the KAS Simulator, the retain variables are stored in a normal disk file.
- Is known by all programs (when its content is changed, the change is propagated to all equations in which this variable is used)
- Normally does not contain real-time critical data.

TIP

- Retain variables are saved when you stop the PLC.
- The next time the PLC starts, the retained values are used.
- Retains are auto-saved every two seconds.
  - Changed variables stored in non-volatile memory are updated every two seconds.
  - Unchanged variables are not updated.

Retain variables are typically used to store application-specific persistent state.

Example: Counting cutting edge cycles so maintenance is triggered after a preset number of operations:

- "Retained Variable Initialization Behavior" (➜ p. 289)
- "Best Practices Retained Variables" (➜ p. 293)
- "Copy Retained Variable Values to Init Value" (➜ p. 294)
- "Copy Retained Variables Values to Init Value from another Controller" (➜ p. 298)

7.1.2.2 Retained Variable Initialization Behavior

- On an **Application Start**, KAS initializes the retain variables using values stored in non-volatile memory (NVRAM).
- On an **Application Cold Start**, KAS initializes the retain variables using their Init values.
  - See "Connect to the Controller" (➜ p. 262).

7.1.2.3 Warning - Dependency on Project Compatibility

- If the declaration of the retain variables is different between the non-volatile memory and the project, the retain variables are also reset to their Init values.

KAS Runtime determines compatibility using:

- The number of variables of each type.
- The length of a STRING type variables.

7.1.2.4 Non-volatile Storage Considerations

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The non-volatile memory size is hardware dependent.
- If the size of the retained variables is larger than the non-volatile storage space:
  - An error is logged.
  - Data is **not** stored in non-volatile memory.
- See "NVRAM" (→ p. 771).

For the KAS Runtime Simulator, the retained variables are saved in a file in the project repository.

7.1.2.5 Actions that Reset Retained Values to Init Values

These changes invalidate existing retained data and reset to their Initial values.

- Change the type of a retain variable.
- Change the length of a string retain variable.
- Change the size of an array variable.
- Change any element of a structure variable.
  - This includes adding new members.

⚠WARNING

Even small structural changes (e.g., adding one field) cause a complete reset of the affected retained variable.

7.1.2.6 Variables

All variables used in programs must be declared first in the variable editor.

Each variable belongs to a group and must be identified by a unique name in its group.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.1.2.6.1 Groups

A group is a set of variables.

A group refers to a physical class of variables or identifies the variables local to a program or user-defined function block.

This table lists the possible groups.

Groups	Description
%I...	Channels of an input board. • Variables with same data type are linked to a physical **input** device.
%Q...	Channels of an output board. • Variables with same data type are linked to a physical **output** device.
GLOBAL	Internal variables known by all programs.
PROGRAMxxx	All internal variables local to a program. The name of the group is the name of the program.
"Retain Variables" (➔ p. 292)	Non volatile internal variables known by all programs. • The latest values from RETAIN variables are stored from the Runtime into a file on the hard disk drive (HDD). • In case of a warm (re)start, or a cold start, the Runtime initializes the variables with these stored values. • The Runtime stores these values periodically per default, triggered every 10ms in an own, lower priority thread. • The storage can be configured using either: • The menu tab entry **Project/Settings.../Runtime/Cycle** time. • The function **F_SAVERETAIN** used in the program code. • If a device with zenon does not have a robust HDD, it is recommended to deactivate the periodical storage and store seldom. • This can be done via a manual function call.
UDFBxxx	• All internal variables local to a User-Defined Function Block plus its IN and OUT parameters. • The name of the group is the name of the program.

7.1.2.6.2 Data Type and Dimension

Each variable must have a valid data type.

- It can be either a basic data type or a function block.
  - In a function block, the variable is an instance of the function block.
- Physical I/Os must have a basic data type.
- Instances of function blocks can refer either to a standard block or to a User Defined Function Block.

If the selected data type is STRING, you must specify a maximum length.

This cannot exceed 255 characters.

- See the list of "Data Types" (➔ p. 286).
- See Call a Function Block about using a function instance.
- Specify dimensions for an internal variable to declare "Arrays" (➔ p. 287).

7.1.2.6.3 Name a Variable

A variable must be identified by a unique name within its parent group.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The variable name cannot:
  - Be a reserved keyword of the programming languages.
  - Have the same name as a standard or C function or function block.
- A variable must not have the same name as a program or a user-defined function block.
- The name of a variable must begin by a letter or an underscore (_), followed by letters, digits, or underscore marks.
  - Two consecutive underscores in a variable name is **not** allowed.
- Naming is case-insensitive.
  - Two names with different cases are considered as the same.

7.1.2.6.4 Variable Attributes

Physical I/Os are marked as either **Input** or **Output**.

- Each internal variable can be configured as Read / Write or Read-only.
  - See "Attributes" (➔ p. 106).
    - Read-only variables can be mapped to Outputs but not to Inputs.
    - Inputs can change state and a Read-only variable cannot change its value to match the input state.
- Parameters of User-Defined Function Blocks are marked as either **IN**or **OUT**.

7.1.2.7 Retain Variables

A retain variable is a PLC variable which:

- Is non-volatile and stored in non-volatile controller memory (PCMM2G).
  - When using the KAS Simulator, the retain variables are stored in a normal disk file.
- Is known by all programs (when its content is changed, the change is propagated to all equations in which this variable is used)
- Normally does not contain real-time critical data.

TIP

- Retain variables are saved when you stop the PLC.
- The next time the PLC starts, the retained values are used.
- Retains are auto-saved every two seconds.
  - Changed variables stored in non-volatile memory are updated every two seconds.
  - Unchanged variables are not updated.

Retain variables are typically used to store application-specific persistent state.

Example: Counting cutting edge cycles so maintenance is triggered after a preset number of operations:

- "Retained Variable Initialization Behavior" (➔ p. 292)
- "Best Practices Retained Variables" (➔ p. 293)
- "Copy Retained Variable Values to Init Value" (➔ p. 294)
- "Copy Retained Variables Values to Init Value from another Controller" (➔ p. 298)

7.1.2.7.1 Retained Variable Initialization Behavior

- On an **Application Start**, KAS initializes the retain variables using values stored in non-volatile memory (NVRAM).
- On an **Application Cold Start**, KAS initializes the retain variables using their Init values.
  - See "Connect to the Controller" (➔ p. 262).

7.1.2.7.2 Warning - Dependency on Project Compatibility

- If the declaration of the retain variables is different between the non-volatile memory and the project, the retain variables are also reset to their Init values.

KAS Runtime determines compatibility using:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The number of variables of each type.
- The length of a STRING type variables.

7.1.2.7.3 Non-volatile Storage Considerations

- The non-volatile memory size is hardware dependent.
- If the size of the retained variables is larger than the non-volatile storage space:
  - An error is logged.
  - Data is **not** stored in non-volatile memory.
- See "NVRAM" (➔ p. 771).

For the KAS Runtime Simulator, the retained variables are saved in a file in the project repository.

7.1.2.7.4 Actions that Reset Retained Values to Init Values

These changes invalidate existing retained data and reset to their Initial values.

- Change the type of a retain variable.
- Change the length of a string retain variable.
- Change the size of an array variable.
- Change any element of a structure variable.
  - This includes adding new members.

⚠WARNING

Even small structural changes (e.g., adding one field) cause a complete reset of the affected retained variable.

See Also

- "Best Practices Retained Variables" (➔ p. 293)
- "Copy Retained Variable Values to Init Value" (➔ p. 294)

7.1.2.7.5 Best Practices Retained Variables

ⓘIMPORTANT

- **Do not use retained variables as outputs of Function Blocks.**
  - Doing so may cause problems with retaining the value if the FB is executed with the Enable input off.
  - The output structure can cause it to be conditionally updated to zero or the old value.

A work-around solution is to use a `copy (1 gain)` function to selectively enable updating the retain variable from an intermediate variable, which is the output of a function block. (Figure 7-1)

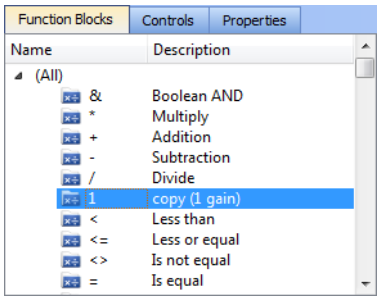


Figure 7-1: Example: `copy (1 gain)`

Work-around example

In Figure 7-2:

- The `ReadActualPositionX` is continually reading position feedback from an axis.
  - The copy block is saving a specific value to a retain variable.
- `ActualPositionX` is the intermediate variable and `SavedPositionX` is the retain variable.

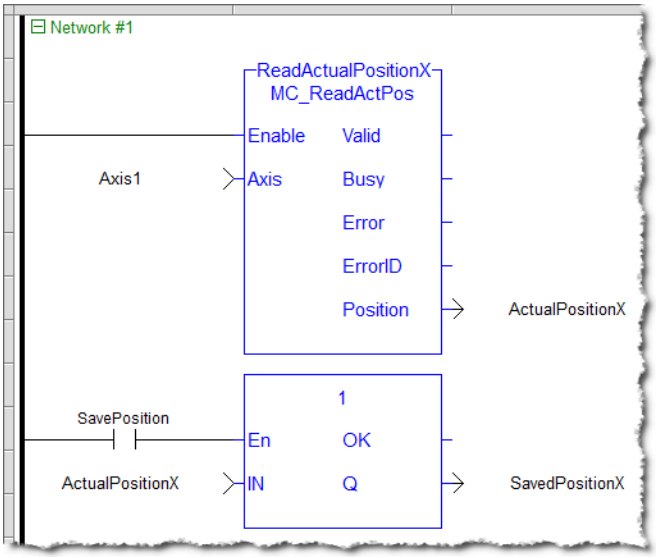


Figure 7-2: Example: Workaround

7.1.2.7.6 Copy Retained Variable Values to Init Value

Once a retained variable's runtime value has been validated, it can be copied into its Init Value.

The Copy Retained Variable Runtime Values to Init Values feature ensures that runtime validated values are preserved inside the project itself, not only in controller memory.

This provides several benefits:

- Protects against retained data loss due to project changes.
- Captures commissioning, tuning, or calibration results.
- Makes the application deterministic across deployments.
- Enables deploying the application to new or replacement machines.
- Eliminates manual reentry of tuned values.

7.1.3 Preconditions

Before using this procedure, verify:

- KAS-IDE is connected to the controller.
- The application is in a valid runtime state.
- The compiled project version in KAS-IDE matches the version running in KAS Runtime.

Procedure

1. Open the KAS project in KAS-IDE.
2. Connect to the controller.
3. Compile and Download the project, if necessary.
4. Start the project, if not running already.
5. Wait until the system reaches a stable operational state.
6. Navigate to the Variables tab in the KAS-IDE.
7. Right-click on the **Retain Variables** entry and select **Copy variable value to init value**. (Figure 7-3)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

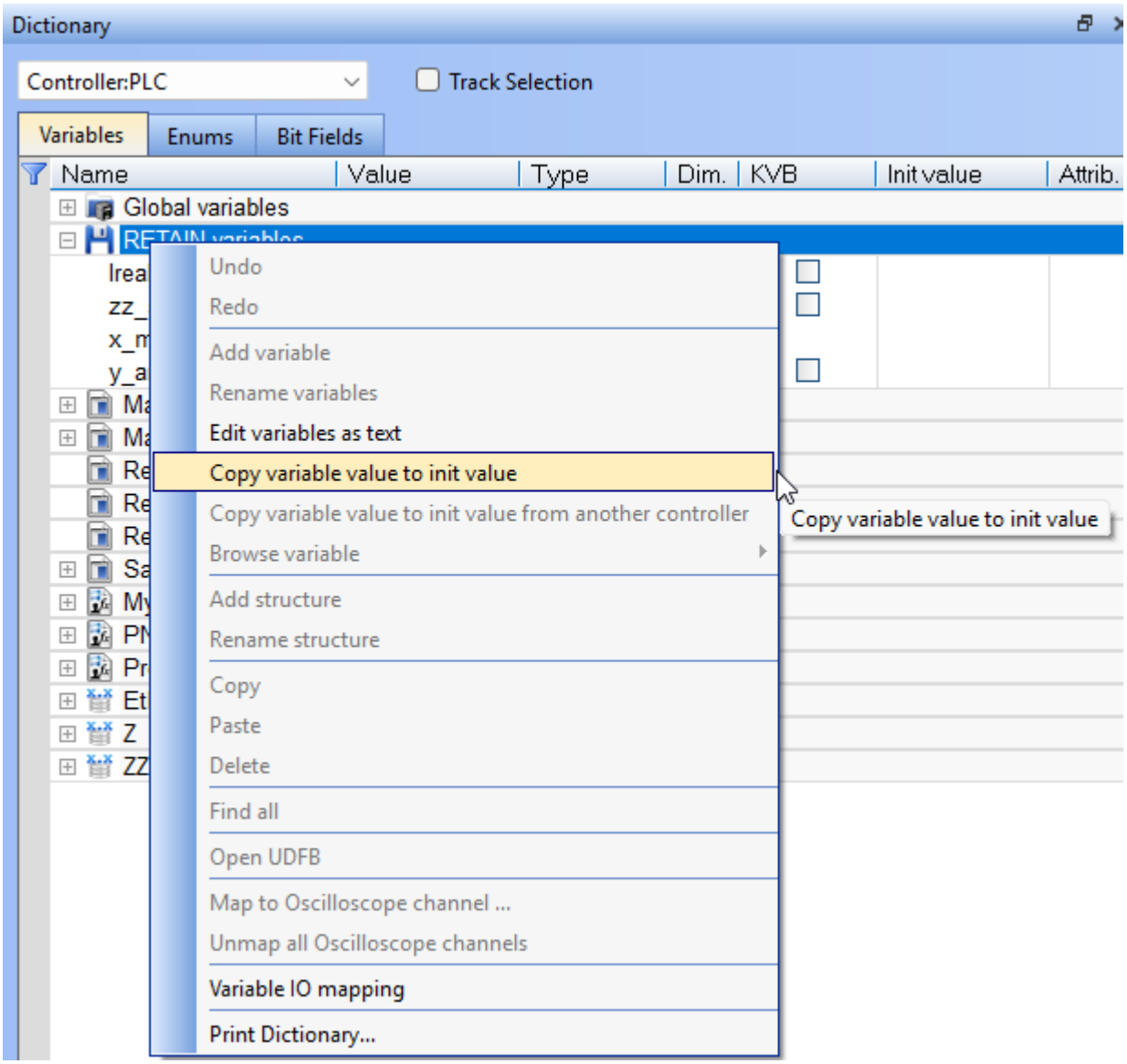


Figure 7-3: Copy variable value to init value

NOTE

This option is also available for individual variables in the Retain Variables section.

8. Confirm the operation when prompted.

After Completion

The current runtime value of each retained variable is written to its Init Value in the project configuration.

9. Save the project to avoid losing changes.
10. Compile and download the updated project to the controller if you want the new init values to take effect.

Example Procedure

This example determines a retain variable's value and saves the value to the retain variable's initial value.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. Run the program to determine the retain variable's value, which is FALSE. (Figure 7-4)

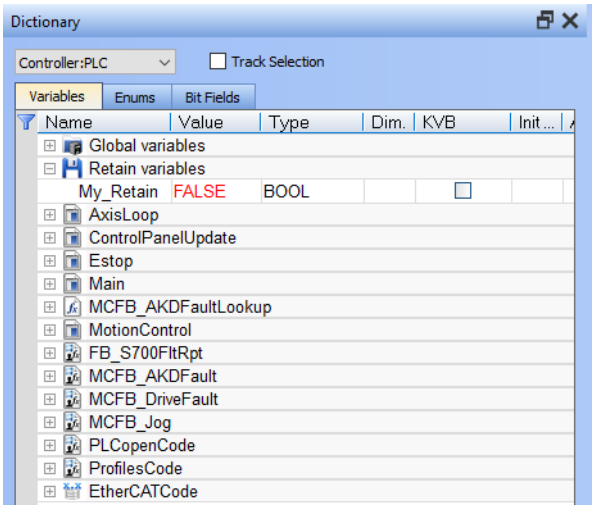


Figure 7-4: Example: Retain Variable = FALSE

2. Right-click the variable and select **Copy variable value to init value**. (Figure 7-5)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

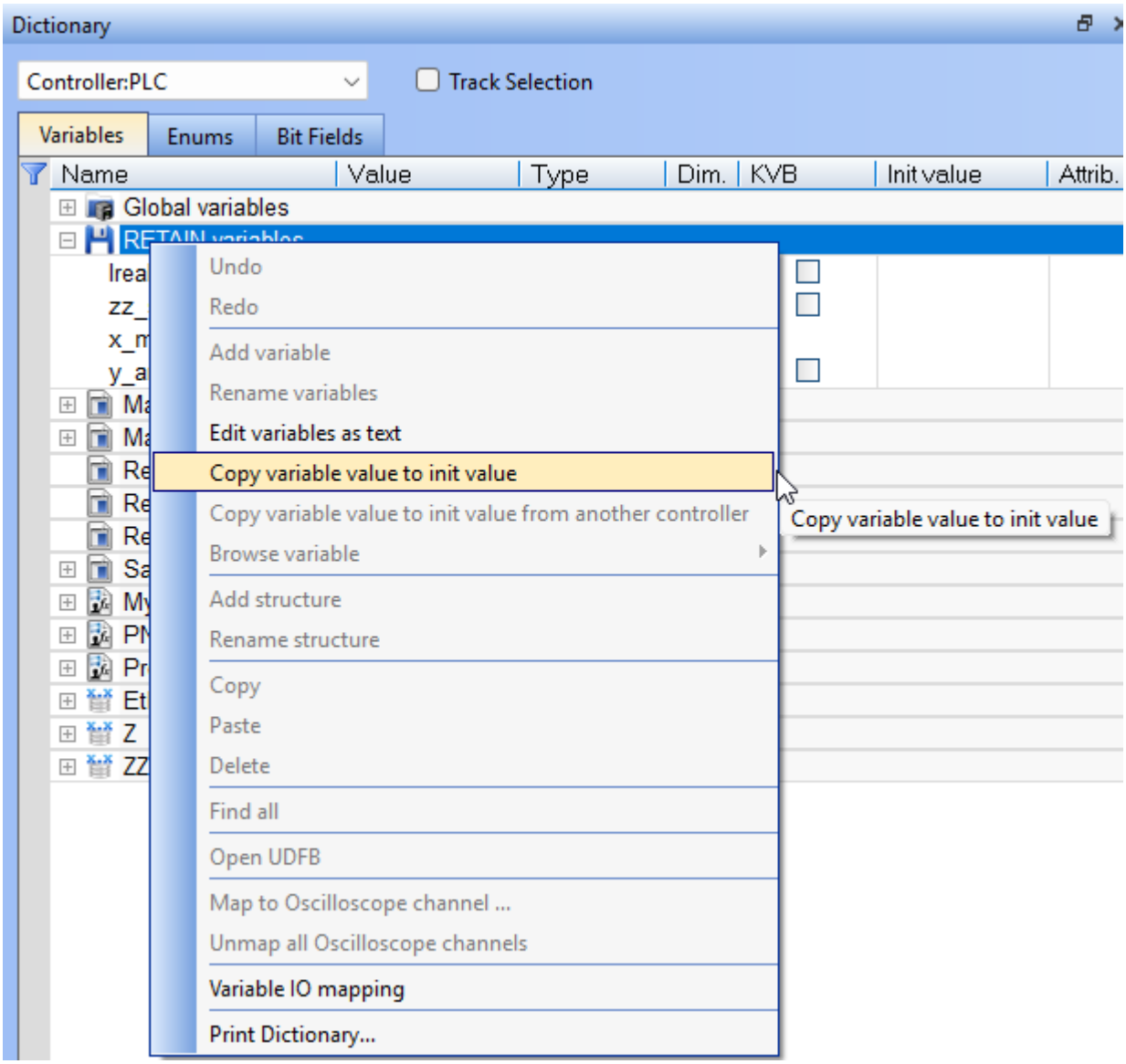


Figure 7-5: Right-click, select **Copy variable value to init value**

The Init value field is now FALSE. (Figure 7-6)

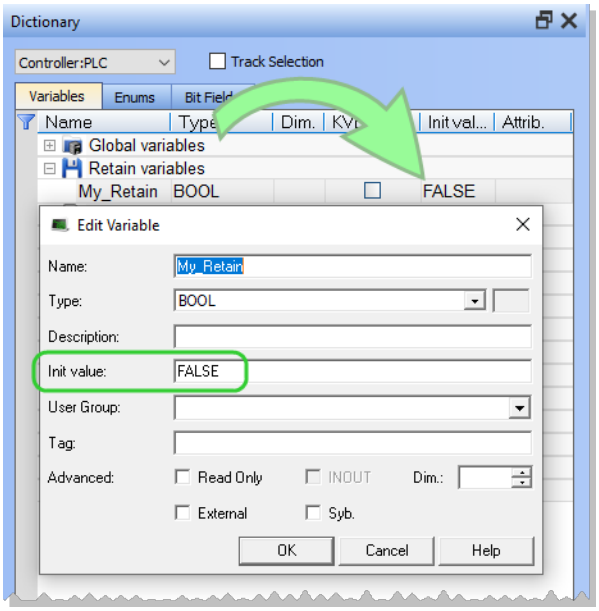


Figure 7-6: Init value = FALSE

7.1.3.0.1 Copy Retained Variables Values to Init Value from another Controller

The Copy Variable Values to Init Values from Another Controller is used to:

- Transfer retained variable values from a source controller.
- Store them as initial values within the current KAS project.
  - These values can then be applied to a replacement controller during project download and application startup.
- This feature can be used for any controller replacement scenario where retained variable definitions are compatible between the source controller and the target project.

Typical use cases include:

- Replacing a failed controller with a new controller.
- Migrating an application to replacement hardware.
- Preserving machine-specific settings, calibration data, production counters, and configuration parameters during controller replacement.

NOTE

- This feature is particularly valuable when migrating from legacy AKD PDMM or PCMM controllers to a PCMM2G controller because it provides a simple method to preserve retained data during platform migration.
- For PCMM2G-to-PCMM2G replacement scenarios, this feature remains fully supported.
  - Alternative methods (e.g., project backup and restore) are also be available.

The migration process has these basic steps:

A. "Transfer Retain Variables from Source Controller to the KAS Project" (➔ p. 299)
B. "Transfer Retained Variables values to the Target Controller" (➔ p. 301)

Prerequisites

Before transferring retained variable values, verify:

- The source controller is online and reachable over network.
- The KAS project on the source controller matches the project currently opened in the KAS-IDE.
- The retained variable definitions in the source controller and project are identical.
- The project is running on the source controller.
  - See Run the Project.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Remote access to PLC variables is enabled in the project running on the source controller.
  - See Configure the Controller.

7.1.3.0.2.1 Transfer Retain Variables from Source Controller to the KAS Project

1. Start the KAS-IDE.
2. Open the KAS project associated with the source controller.
3. Navigate to the Variables tab.
4. Right-click the Retain Variables and select **Copy variable value to init values from another controller**. (Figure 7-7)

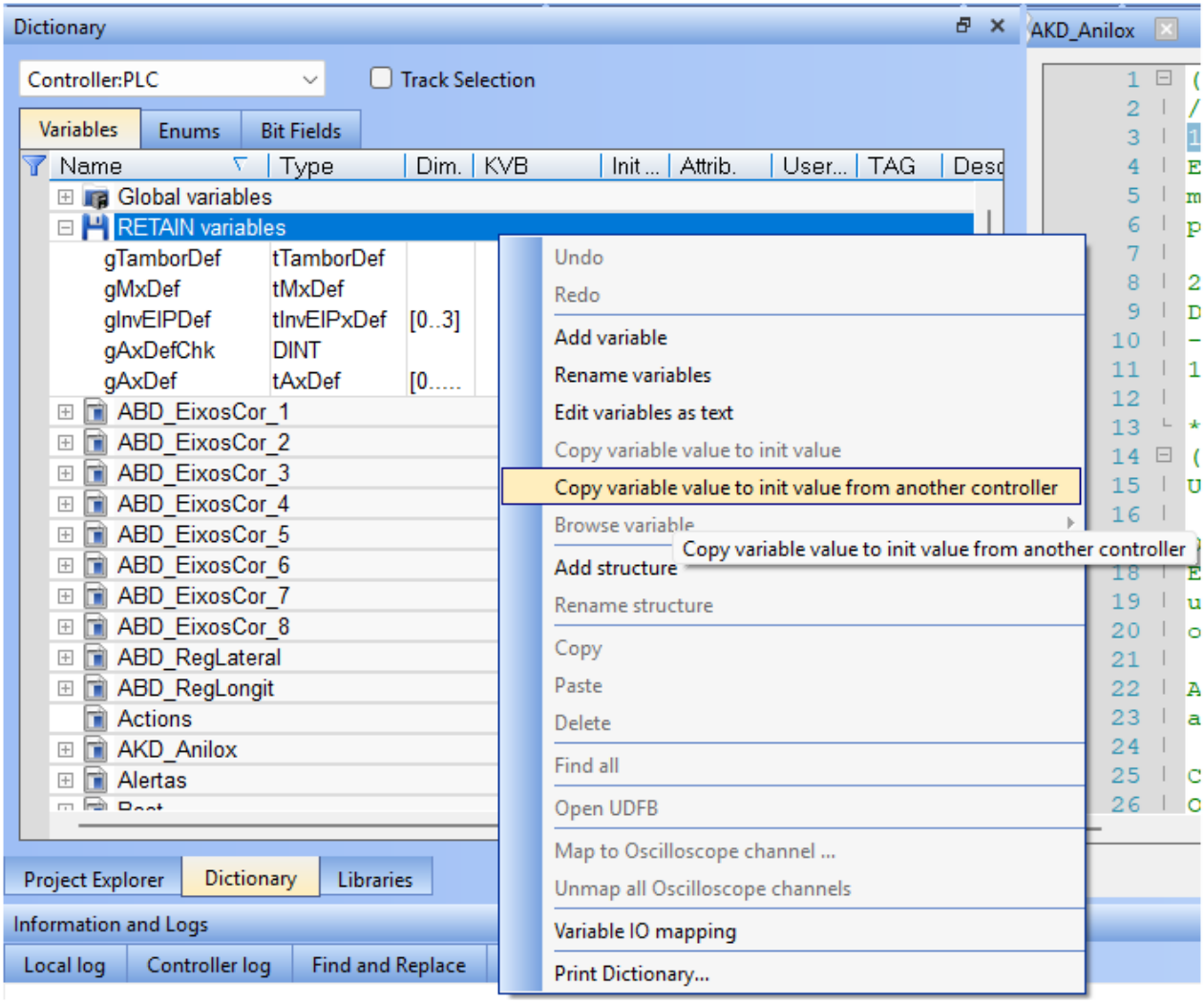


Figure 7-7: Select **Copy variable value to init values from another controller**

The **Copy Variable Values to Init Values** dialog box opens. (Figure 7-8)

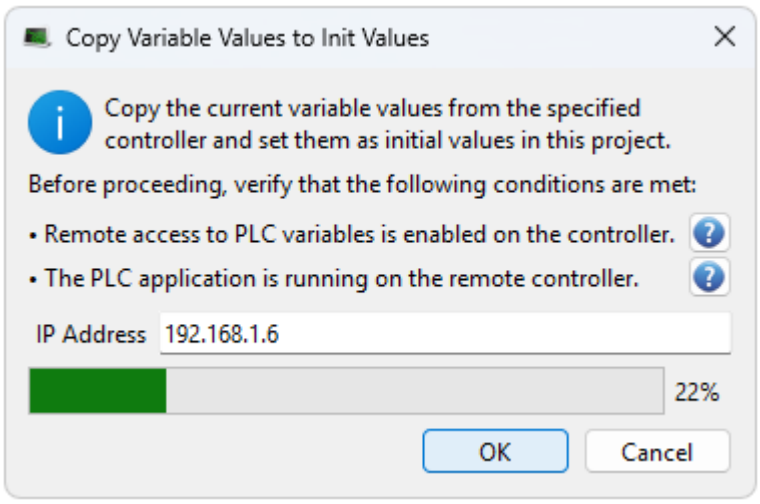


Figure 7-8: Copy Variable Values to Init Values dialog box

5. In the **IP Address** text box, enter the address of the source controller.
6. Click **OK** to save the changes or selections and close the dialog box.

   The KAS-IDE retrieves the retained values from the source controllers and stores them as initial values in the current project.

7.1.4 Validation

To ensure data integrity:

- KAS-IDE validates that the retained variable definitions on the source controller match those in the current project, including:
  - Array dimensions
  - Data type
  - Structure definition
  - Variable name
- Only variables that pass validation are eligible for transfer.

7.1.5 Transfer Failures

If the KAS-IDE is cannot retrieve all retained variables values, a warning message opens identifying the variables. (Figure 7-9)

Typical causes of transfer failures include:

- Communication interruptions.
- Missing variables on the source controller.
- Mismatched variable definitions between the source controller and the project.
- Unsupported variable types.

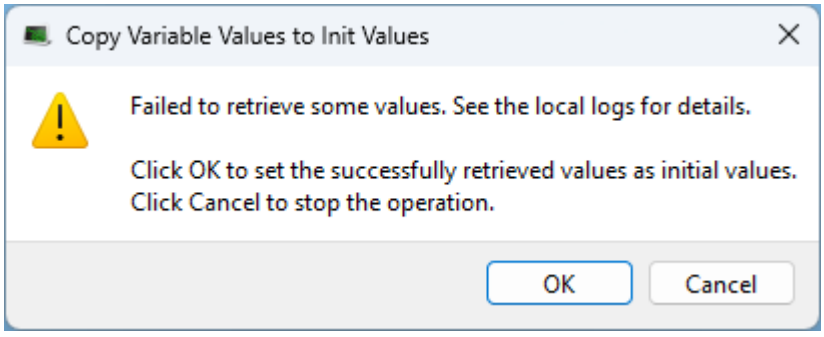


Figure 7-9: Copy Variable Values to Init Values message

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The options are:

- Copy partially retrieved values and continue the operation.
- Cancel the transfer and resolve the reported issues before retrying.

7.1.5.0.1.1 Transfer Retained Variables values to the Target Controller

After the retained variable values have been imported into the KAS project:

1. Connect to the target controller.
2. Compile and download the updated project.
3. Perform a Cold Start of the application:
   a. Click the Start button drop-down.
   b. Select Cold Start - Use Initial values for retain variables. (Figure 7-10)

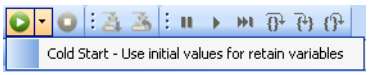


Figure 7-10: Cold Start - Use Initial values for retain variables

During the Cold Start, the controller initializes the retained variables using the values stored in the project, restoring the machine configuration on the replacement controller.

NOTE

A Cold Start is required to apply the imported initial values to the retained variables on the target controller.

7.1.6 Constant Expressions

Constant expressions can be used in all languages for assigning a variable with a value.

All constant expressions have well-defined **Data Types** according to their semantics.

IMPORTANT

If you program an operation between variables and constant expressions having inconsistent data types, it leads to syntactic errors when the program is compiled.

These are the syntactic rules for constant expressions according to possible data types:

Type	Prefix	Description
BOOL		**Boolean** • These are the Boolean reserved keywords: • `TRUE` • `FALSE` • See "Valid Constant Expressions" (→ p. 305) for an example.
DINT		**32-bit (default) Integer** • 32-bit integer constant expressions must be valid numbers between -2147483648 to 2147483647. • DINT is the default size for integers: such constant expressions do not need any prefix. • Use **2#**, **8#**, or **16#** prefixes to specify an integer in binary, octal or hexadecimal basis, respectively. • See "Valid Constant Expressions" (→ p. 305) for an example.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Type	Prefix	Description
INT	INT#	**16-bit Integer** • 16-bit integer constant expressions are valid integer values (between -32768 and +32767). • Must be prefixed with `INT#`. • All integer expressions having no prefix are considered "DINT" (→ p. 301) integers. • See "Valid Constant Expressions" (→ p. 305) for an example.
LINT	LINT#	**Long (64-bit) Integer** • Long integer constant expressions are valid integer values. • Must be prefixed with LINT#. • All integer expressions having no prefix are considered "DINT" (→ p. 301) integers. • See "Valid Constant Expressions" (→ p. 305) for an example.
LREAL	LREAL#	**Double Precision Floating Point Value** • Real constant expressions must be valid numbers, must include a dot (**.**). • If you need to enter a real expression having an integer value, add **.0** (dot zero) at the end of the number. • You can use **F** or **E** separators for specifying the exponent in case of a scientific representation. • LREAL constants are limited to 14-15 digits of accuracy. • Any digits after these significant digits are lost, leading to a loss of precision. • See "Valid Constant Expressions" (→ p. 305) for an example.
REAL		**Single Precision Floating Point Value** • REAL is the default precision for floating points: such expressions do not require a prefix. • **Important**: REAL is restrictive, but because it is the default, it is recommended to explicitly declare your real constants with the LREAL# prefix. • Real constant expressions must be valid numbers, must include a dot (**.**). • If you need to enter a real expression having an integer value, add **.0** (dot zero) at the end of the number. • You can use **F** or **E** separators for specifying the exponent in case of a scientific representation. • REAL constants are limited to 6-7 digits of accuracy. • Any digits after these significant digits are lost, leading to a loss of precision. • See "Valid Constant Expressions" (→ p. 305) for an example.
SINT	SINT#	**Small (8-bit) Integer** • Small integer constant expressions are valid integer values (between -128 and +127). • Must be prefixed with `SINT#`. • All integer expressions having no prefix are considered "DINT" (→ p. 301) integers. • See "Valid Constant Expressions" (→ p. 305) for an example.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

<table>
<tr><th>Type</th><th>Prefix</th><th>Description</th></tr>
<tr><td>STRING</td><td></td><td>Character String
String expressions must be written between single quote marks (' ').
The length of the string cannot exceed 255 characters.
See "Valid Constant Expressions" (→ p. 305) for an example.
Use these sequences to represent a special or not-printable character within a string:
<table>
<tr><th>Sequence</th><th>Description</th></tr>
<tr><td>$$</td><td>A "$" character</td></tr>
<tr><td>$'</td><td>A single quote</td></tr>
<tr><td>$T</td><td>A tab stop (ASCII code 9)</td></tr>
<tr><td>$R</td><td>A carriage return character (ASCII code 13)</td></tr>
<tr><td>$L</td><td>A line feed character (ASCII code 10)</td></tr>
<tr><td>$N</td><td>Carriage return plus line feed characters (ASCII codes 13 and 10)</td></tr>
<tr><td>$P</td><td>A page break character (ASCII code 12)</td></tr>
<tr><td>$xx</td><td>Any character (xx is the ASCII code expressed on two hexadecimal digits</td></tr>
</table></td></tr>
<tr><td>TIME</td><td>T#
or
TIME#</td><td>Time of Day
Time-constant expressions represent durations that must be less than 24 hours.
Expressions must be prefixed by either <code>T#</code> or <code>TIME#</code>.
They are expressed as a number of:
hours followed by h
minutes followed by m
seconds followed by s
milliseconds followed by ms
The order of units (hour, minutes, seconds, milliseconds) must be respected.
Blank characters are not allowed in the time expression.
There must be at least one valid unit letter in the expression.
See "Valid Constant Expressions" (→ p. 305) for an example.</td></tr>
<tr><td>UDINT/DWORD</td><td>UDINT#</td><td>Unsigned 32-bit Integer
Unsigned 32-bit integer constant expressions are valid integer values (between 0 and 4294967295).
Must be prefixed with <code>UDINT#</code>.
All integer expressions having no prefix are considered "DINT" (→ p. 301) integers.</td></tr>
</table>

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Type	Prefix	Description
UINT/WORD	UINT#	**Unsigned 16-bit Integer** • Unsigned 16-bit integer constant expressions are valid integer values (between 0 and +65535). • Must be prefixed with `UINT#`. • All integer expressions having no prefix are considered "DINT" (→ p. 301) integers.
ULINT/LWORD	ULINT#	**Unsigned Long Unsigned (64-bit) Integer** • Unsigned 64-bit integer constant expressions are valid integer values. • All integer expressions having no prefix are considered "DINT" (→ p. 301) integers.
USINT/BYTE	USINT#	**Unsigned 8-bit Integer** • Unsigned small integer constant expressions are valid integer values (between 0 and 255). • Must be prefixed with `USINT#`. • All integer expressions having no prefix are considered "DINT" (→ p. 301) integers.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.1.6.1 Examples

7.1.6.1.1 Valid Constant Expressions

These are examples of valid constant expressions.

Constant Expression	Type	Description
'hello'	Character String	Character string.
'I$'m here'	Character String	Character string with a quote inside (I'm here).
'name$Tage'	Character String	Character string with two words separated by a tab.
'x$00y'	Character String	Character string with two characters separated by a null character (ASCII code 0).
0.0	REAL	0 expressed as a REAL number.
1.002E3	REAL	1002 expressed as a REAL number in scientist format.
2#1000100	DINT	DINT integer in binary basis.
8#34712	DINT	DINT integer in octal basis.
16#abcd	DINT	DINT integer in hexadecimal basis.
123456	DINT	DINT (32-bit) integer.
FALSE	BOOL	FALSE Boolean expression.
INT#2000	16-bit Integer	16-bit integer.
LINT#1	Long (64-bit) Integer	Long (64 bit) integer having the value 1.
LREAL#1E-200	Double Precision Floating Point Value	Double precision real number.
SINT#127	Small (8-bit) Integer	Small integer.
T#1h123ms	Time of Day	TIME value with some units missing.
T#23h59m59s999ms	Time of Day	Maximum TIME value.
TIME#0s	Time of Day	Null TIME value.
TRUE	BOOL	TRUE Boolean expression.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.1.6.1.2 Invalid Constant Expressions

These are examples of errors in constant expressions:

Invalid Constant Expressions	Description
'I'm here'	Quote within a string with "$" mark omitted.
1a2b	Basis prefix ("16#") omitted.
1E-200	"LREAL#" prefix omitted for a double precision float.
BooVar := 1;	0 and 1 cannot be used for Booleans.
hello	Quotes omitted around a character string.
T#12	Time unit missing.

NOTE

- There are pre-defined constants.
- See "Use the Defines List" (→ p. 523), "Internal Defines" (→ p. 524), and "Global Defines" (→ p. 525).

7.1.7 Program Organization Units

Within IEC 61131-3, the "Action Blocks (POU)" (→ p. 310), "Functions (POU)" (→ p. 307), "Function Blocks (POU)" (→ p. 307), and "Programs (POU)" (→ p. 308) are called Program Organization Units (POU).

In addition to the IEC standard, you can write you own code as either a sub-program or UDFB.

Types	IEC 61131-3	Written by End-user
• Basic functions. • Has no memory.	"Functions (POU)" (→ p. 307)	"Action Blocks (POU)" (→ p. 310) / "Programs (POU)" (→ p. 308) / "Sub-programs" (→ p. 308)
• Instantiated functions. • Keep track of the past.	"Function Blocks (POU)" (→ p. 307) (FB)	"User-Defined Function Blocks (POU)" (→ p. 315) (UDFB)

7.1.7.1 Differences between Functions and Function Blocks

- Functions are expected to complete in one cycle.
- Function Blocks can take several cycles to complete.

7.1.7.1.1 Description of FB Operation

Rather than halt the application or waiting for operations to complete, the FB typically gives control back to the application but does not set its **Done** output.

7.1.7.1.2 Examples of Operations Overrunning the Cycle Duration

- A motion command to move from one location to another can take several cycles to complete.
- Operations like reading/writing to files or reading and writing over TCP/IP can also take several cycles to complete.

7.1.7.1.3 Operation Sequence

1. When an FB is called, it starts an operation and possibly does not complete it.
2. The FB is called in the next cycle and it checks to determine if the operation is done.
3. If it is done, it sets the **Done** output.
   If not, it continues on.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4. Now the application knows that the operation is complete and can do what ever other processing it needs based on the FB being done.

7.1.7.2 Functions (POU)

IEC has defined standard functions and also allows you to create your own functions (called user-defined functions). Typically, functions take several inputs and return a single output as the result of processing.

- Standard functions are for example ADD (addition), ABS (absolute), SIN (sine), COS (cosine), GT (Greater Than),….
- User-defined functions, as in this example, can be used repeatedly once defined.

```
FUNCTION SIMPLE_FUN : REAL
       VAR_INPUT
               A, B : REAL;
               C : REAL := 1.0;
       END_VAR
       SIMPLE_FUN := A*B/C;
END FUNCTION
```

7.1.7.3 Function Blocks (POU)

Function Blocks (FBs) take several inputs and return a group of values as the output as the result of processing.

- Function Blocks are the equivalent to Integrated Circuits (IC), representing a specialized control function.
- They are specified at such a level that you quickly recognize the functionality of the function block and specifically what happens if it is activated or connected to other blocks in a sequence of motion commands.
- They contain data as well as an algorithm, so they can keep track of the past (which is one of the differences from Functions).
    - They have a well-defined interface and hidden internals, like an IC or a black box.
    - The user only sees the interface, being the inputs and outputs.
    - The code itself is hidden.
- Function Blocks can be used in any of the IEC languages.
    - In an SFC program, function blocks can be part of a step or transition created in FBD, FFLD, IL Language, and ST Language.
- Once defined, they can be used repeatedly, in the same program, different programs, or even different projects. This makes them highly re-usable.
- There are predefined function blocks (e.g., timers, counters, or triggers) and additional function blocks that can come from libraries produced by you or other suppliers (e.g., a temperature control-loop or PID loop.).

Example of Function Blocks

- The function block is based on the programming language function block Diagram and has the name Hysterisis. (Figure 7-11)
- It has three inputs (XIN1, XIN2, and EPS) of datatype REAL on the left and one output (called Q) of type BOOL on the right-hand side. (Figure 7-12)

TIP

Input names are not very usable; use meaningful names.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

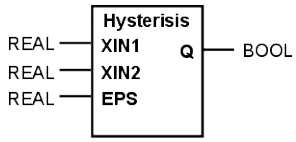


Figure 7-11: Example Inputs

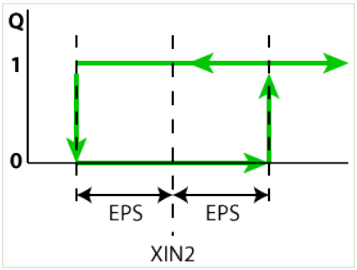


Figure 7-12: Example Outputs

Internally, the FB contains this body code:

```
FUNCTION_BLOCK HYSTERISIS
VAR_INPUT
       XIN1, XIN2 : REAL;
       EPS : REAL;  (* Hysterisis ban *)
END_VAR
VAR_OUTPUT
       Q : BOOK := 0
END_VAR
IF Q THEN
       IF XIN1 < ( XIN2 - EPS ) THEN
                Q := 0 (* XIN2 decreasing *)
       END_IF;
       ELSIF XIN1 > ( XIN2 + EPS ) THEN
                Q := 1; (* XIN2 increasing *)
       END_IF;
END_FUNCTION_BLOCK
```

In this example, the body code is written in ST Language:

- The first part deals with the data structure.
- The second with the algorithm.
- No additional data is used.

Whatever name was used for this local data inside the body does not conflict with matching names in other functions, function blocks, or with global expressions.

This example of data encapsulation removes a major source of errors.

7.1.7.4 Programs (POU)

With the basic building blocks, a program can be seen as a network of functions and function blocks. Each of them being written in any of the defined Programming Languages.

TIP

Programs and Subprograms can be reordered by drag-and-drop.
Dragging an item up the list places it above the program you drop it on, while dragging down places it below.

7.1.7.4.1 Sub-programs

The list of programs is completed with Sub-programs.

- Sub-programs are written in FBD, FFLD, IL Language, or ST Language.
  - They are called by the programs of the application.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Input and output parameters plus local variables of a sub-program are declared in the variable editor as local variables of the sub-program.
- A function type sub-program can call another function type subprogram.
- A UDFB type sub-program can call another UDFB-type subprogram or a function type subprogram.

Unlike UDFB, local variables of a sub-program are not instantiated.

- This means that the sub-program always works on the same set of local variables.
  - Local variables of a sub-program keep their value among various calls.
  - The code of a sub-program is not duplicated when called several times by parent programs.
- A sub-program cannot have more than 32 input parameters or 32 output parameters.
- A good programming practice is to break up your programs into smaller modules.
- See "Application Software Structure - Definitions" (➔ p. 1008).

7.1.7.4.2 Program Guidelines

An application is a list of programs. Programs are executed sequentially within the target cycle, according to this model:

```
 Begin cycle
| exchange I/Os
| execute first program
| ...
| execute last program
| wait for cycle time to be elapsed
End Cycle
```

Programs are executed according to the order defined by the user.

- All SFC programs must be grouped (it is not possible to insert a program in FBD, FFLD, IL Language, or ST Language between two SFC programs).
- Each program is entered using a language chosen when the program is created.
- Possible languages are:
  - Free Form Ladder Diagram (FFLD)
  - Function Block Diagram (FBD)
  - Instruction List (IL Language).
  - Sequential Function Chart (SFC)
  - Structured Text (ST Language)
- The maximum number of programs in an application is 32767.

7.1.7.4.3.1 Program Name

Programs must have unique names.

- A program name cannot:
  - Be a reserved keyword of the programming languages.
  - Have the same name as a standard or "C" Function or function block.
  - Have the same name as a declared variable.
- The name of a program must begin by a letter or an underscore ("_") mark, followed by letters, digits or underscore marks.
  - It is not allowed to put two consecutive underscores within a name.
- Naming is case-insensitive.
  - Two names with different cases are considered as the same.

7.1.7.4.4.2 Child SFC Programs

- You can define an hierarchy of SFC programs, entered as a tree in the list of programs.
  - See SFC Program Hierarchy.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- A child program is controlled within action blocks of the parent SFC program.
  - See Control an SFC Child Program.

TIP

- If you do not want to split your FFLD program, separate FFLD from SFC.
  - Make a sub-program in FFLD called from the SFC step.
  - Keep only the state machine in the SFC program.
- This makes it simpler for editing and debugging.

7.1.7.4.5 Program Limitations

When creating your application you have to consider these important limitations.

7.1.7.4.6.1 FFLD Programs

- Width of any network is limited to 255 columns.
- Height of any network is limited to 255 rows.

7.1.7.4.7.2 Other Programs

For any program, sub-program or UDFB written in **other** languages:

- Jump limit is 64KB.
  - Example: In an FFLD program, if you create a UDFB or program which is over 64KB and then add a jump to label in the first network to the last network, this jump reaches the limit.
- Total PLC code size of the program, sub-program or UDFB cannot exceed 64KB.

NOTE

- If the PLC code size of the program exceeds 64KB, split the logic into smaller chunks using Action Blocks (POU).
- Acton Blocks function similarly to subprograms but share the parent program's variables and execution context.

7.1.7.4.8.3 SFC Programs

- Actions in SFC steps cannot be more than 32KB.
- Condition in SFC transition cannot exceed 32KB.
- Total PLC code size of the program cannot exceed 64KB.

7.1.7.4.9 Action Blocks (POU)

Action Blocks are a structuring feature that allows automation logic to be segmented into named, reusable units that can be called by programs, steps, or execution logic.

Action Blocks help structure automation logic by separating what is done from when it is done.

They are designed to:

- Improve readability of large IEC 61131-3 projects.
- Reduce code duplication.
- Make applications easier to maintain, debug, and scale.

TIP

Action Blocks are especially useful when developing complex sequences, state based logic, or modular machine behaviors.

See:

- "Concept and Purpose" (→ p. 312)
- "Supported Environments" (→ p. 312)
- "Action Block Ownership and Scope" (→ p. 312)
- "Creating an Action Block" (→ p. 312)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Implementing Logic Inside an Action Block" (➔ p. 313)
- "Using Action Blocks" (➔ p. 313)
  - "Unqualified Invocation" (➔ p. 313)
  - "Qualified Invocation (Fully Qualified Name)" (➔ p. 314)
  - "Name Disambiguation Example" (➔ p. 314)
- "Debugging and Maintenance" (➔ p. 315)
  - "Online Monitoring" (➔ p. 315)
  - "Maintenance Benefits" (➔ p. 315)
- "Action Blocks have an Unexpected Variable Selector Prompt in FFLD Editors" (➔ p. 315)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.1.7.4.10.1 Concept and Purpose

An Action Block represents a logical operation or behavior (e.g., StartMotor, CheckSafety, HandleFault).

Instead of embedding large logic sections directly inside programs or SFC steps, these operations are encapsulated and referenced as needed.

Key characteristics are:

- Executed on demand, not automatically cyclic.
- It can be organized independently from program structure.
- No internal instance memory.
  - Works with the variables of its parent POU.
- Ideal for actions, procedures, or conditional tasks.
- Complements IEC 61131-3 languages rather than replacing them.

This abstraction layer helps structure automation projects without changing the execution model of the PLC runtime.

7.1.7.4.11.2 Supported Environments

Action Blocks are available in the KAS-IDE and are compatible with standard IEC 61131-3 workflows:

- Programming languages include:
  - Function Block Diagram (FBD)
  - "Free Form Ladder Diagram (FFLD) Editor" (➔ p. 496)
  - Instruction List (IL)
  - Structured Text (ST)

They are executed by the KAS runtime without any additional scheduling mechanisms.

7.1.7.4.12.3 Action Block Ownership and Scope

Each Action Block is owned by a parent POU (typically a Program).

- This ownership defines the block's visibility and how it must be referenced when invoked.
- Conceptually, an Action Block is uniquely identified by: ActionBlockName + ParentPOUName.

7.1.7.4.13.4 Creating an Action Block

1. Open a project in KAS-IDE.
2. In the Project Explorer > Programs section, right-click a program.
3. Select Add Action Block and choose a language (e.g., FBD, FFLD, IL, or ST). (Figure 7-13)

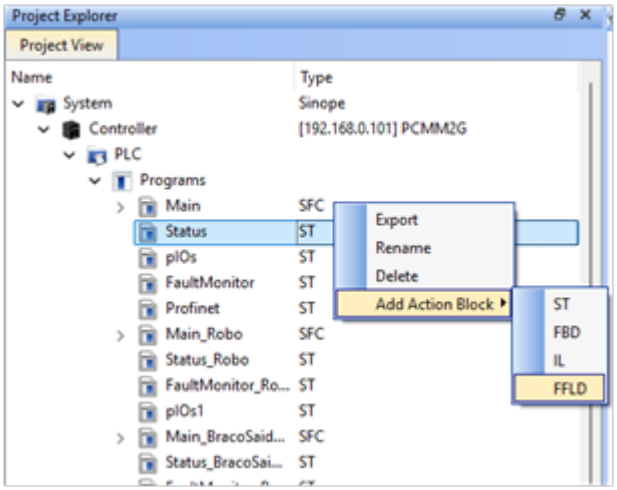


Figure 7-13: Example: Project Explorer > Project View > Programs > Status > Add Action Block

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The New Program dialog box opens.

4. In the text box, enter a unique name for the action block. (Figure 7-14)

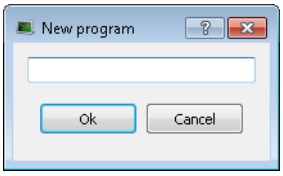


Figure 7-14: New Program dialog box

5. Click **OK** to save the changes or selections and close the dialog box.

TIP

- Naming Guidelines
  - Use verb-based names (e.g., StartPump, ResetAlarm).
  - Avoid generic names (e.g., Action1, Program 1).
  - Match the naming conventions used for programs and function blocks.

The Action Block editor opens automatically, allowing you to implement the logic

7.1.7.4.14.5 Implementing Logic Inside an Action Block

An Action Block contains standard IEC 61131-3 logic, identical in syntax and behavior to program level code.

Example (Structured Text)

```
IF CmdStart AND SafetyOK THEN
    MotorRun := TRUE;
END_IF;
```

Important Notes

- Action Blocks share local variables of its parent POU and has no separate private variables.
- Action Blocks can access:
  - Global variables
  - Functions
  - Function blocks
  - UDFBs
- They do **not** define tasks or execution cycles themselves.
- Execution depends entirely on where and how they are called.

7.1.7.4.15.6 Using Action Blocks

Action Blocks never execute automatically.

- They must be called explicitly from a program.
- Invocation of Action Blocks is supported only from Structured Text (ST) programs and Sequential Function Chart (SFC) step actions.
- The required syntax to invoke action block depends on scope.

7.1.8 Unqualified Invocation

```
ActionBlockName();
```

Valid When

- The Action Block is defined in the same parent POU as the calling code.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Example Structure

Programs
↳ MotorControl
↳ Action Block: StartMotor
↳ ActionBlock: JogMotor

Example Usage

Program MotorControl

```
IF CmdStart THEN
    StartMotor();
END_IF;
```

This form is concise and recommended for local Action Blocks.

7.1.9 Qualified Invocation (Fully Qualified Name)

```
ActionBlockName@ParentPOUName();
```

Required When

- The Action Block is defined in a **different POU**.
- Multiple Action Blocks share the same name.
- Explicit scoping is desired for clarity or safety.

Example Structure

Programs
↳ MotorControl
↳ Action Block: StartMotor
↳ Main (ST logic)

Invocation from Main

```
IF CmdStart THEN
    StartMotor@MotorControl();
END_IF;
```

Attempting to call StartMotor(); in this context results in a compiler error.

7.1.10 Name Disambiguation Example

When multiple POUs define Action Blocks with the same name:

Program PumpControl > Action Block: Start
Program MotorControl > Action Block: Start

Correct Usage:

Start@PumpControl();
Start@MotorControl();

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TIP

- As a best practice, a qualified invocation approach is preferred in large or safety-critical procedures.
- Use Action blocks for actions and procedures, **not** state management.

7.1.10.0.1.1 Debugging and Maintenance

7.1.11 Online Monitoring

- Action Blocks support online monitoring.
- Variable values update in real time.
- Execution flow remains traceable through calling logic.

7.1.12 Maintenance Benefits

- Changes to shared logic require editing only one Action Block.
- Reduces regression risk.
- Improves long term project maintainability.

Action Blocks are especially helpful in multi program or multi machine projects where behavior must remain consistent.

7.1.12.0.1.1 Action Blocks have an Unexpected Variable Selector Prompt in FFLD Editors

Issue:

- Drag-and-drop of an child Action Block from the Library into a parent FFLD editor may cause the Variable Selector dialog box to open as if the Action were a Function Block. (Figure 7-15)
  - This issue does not occur when either:
    - Other editors (e.g., FBD or ST) are used.
    - The Action Block is not the child of the FFLD program.

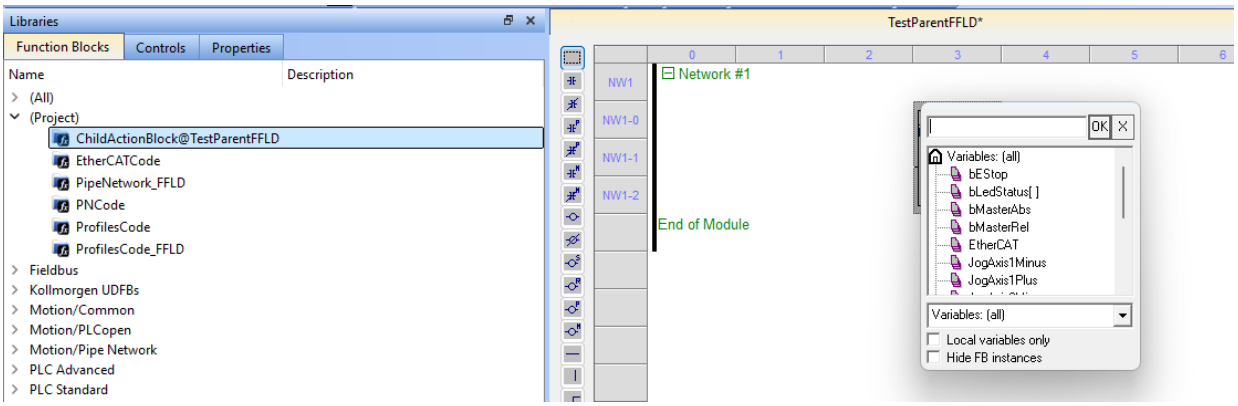


Figure 7-15: Action Blocks - FFLD Workaround

Workaround:

- On the keyboard, press <Esc> to close the dialog box and continue editing.
- The Child Action Block now appears correctly in the editor.

7.1.12.1 User-Defined Function Blocks (POU)

The list of programs is completed with User-Defined Function Blocks (UDFBs).

- UDFBs are described using FBD, FFLD, IL Language SFC, ST Language.
  - They can be used as other function blocks in the programs of the application.
  - Input and output parameters plus private variables of a UDFB are declared in the variable editor as local variables of the UDFB.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- There is no restriction using any operation in a UDFB.
  - A UDFB can call standard functions and function blocks.
- A UDFB can call another UDFB.
  - The called UDFB must be declared **before** the calling one in the program list.

Each time a UDFB is instantiated, its private variables are duplicated for the declared instance.

- The code of the UDFB is duplicated on each call in parent programs.
- This leads to higher performances at run-time, but consumes code space.
- It is recommended to package small algorithms in UDFBs.
  - Large parts of code must be managed in programs.

7.1.13 Programming Languages

In the IEC 11631 standard, syntax and semantics of the programming languages have been defined, leaving no room for variance. Once you have learned them, you can use a wide variety of systems based on this standard.

The languages consist of these textual and graphical language versions:

Graphical

- "Free Form Ladder Diagram (FFLD)" (➜ p. 316)
- "Function Block Diagram (FBD)" (➜ p. 316)
- "Sequential Function Chart (SFC)" (➜ p. 316)

Textual

- "Instruction List (IL Language)" (➜ p. 316)
- "Structured Text (ST Language)" (➜ p. 317)

All the languages are interlinked and provide a common suite.

The choice of programming language depends on:

- The programmer's background.
- The problem to solve.
- The level of describing the problem.
- The structure of the control system.
- The interface to other people / departments.

7.1.13.1 Free Form Ladder Diagram (FFLD)

Free Form Ladder Diagram (FFLD) is based on the graphical presentation of Relay Ladder Logic.

7.1.13.2 Function Block Diagram (FBD)

Function Block Diagram (FBD) is very common to the process industry. It expresses the behavior of functions, function blocks, and programs as a set of interconnected graphical blocks, as in electronic circuit diagrams. It looks at a system in terms of the flow of signals between processing elements.

7.1.13.3 Instruction List (IL Language)

Instruction List (IL) is the European counterpart of FFLD. As textual language and looks like Assembler.

7.1.13.4 Sequential Function Chart (SFC)

Sequential Function Chart (SFC) graphically describes the sequential behavior of a control program.

It is derived from Petri Nets.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

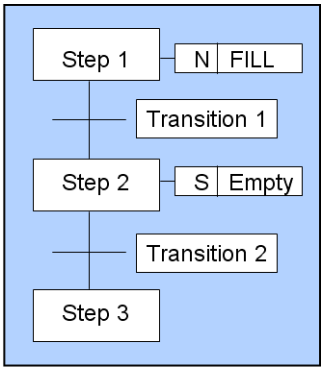


Figure 7-16: Sequential Function Chart

SFC organizes the internal structure of a program and helps to deconstruct a control problem into manageable parts, while maintaining the overview.

SFC consists of steps, linked with Action Blocks and Transitions.

- Each step represents a particular state of the systems being controlled.
- A transition is associated with a condition, which, when true, causes the step before the transition to be deactivated, and the next step to be activated.
- Steps are linked to action blocks, performing a specific control action.
- Each element can be programmed in any of the IEC languages, including SFC itself.

7.1.13.4.1 Alternative and Parallel Sequences

You can use alternative sequences and even parallel sequences, like those commonly required in batch applications.

Example: One sequence is used for the primary process and the second for monitoring the overall operating constraints.

Figure 7-17 shows where parallel sequences are also possible:

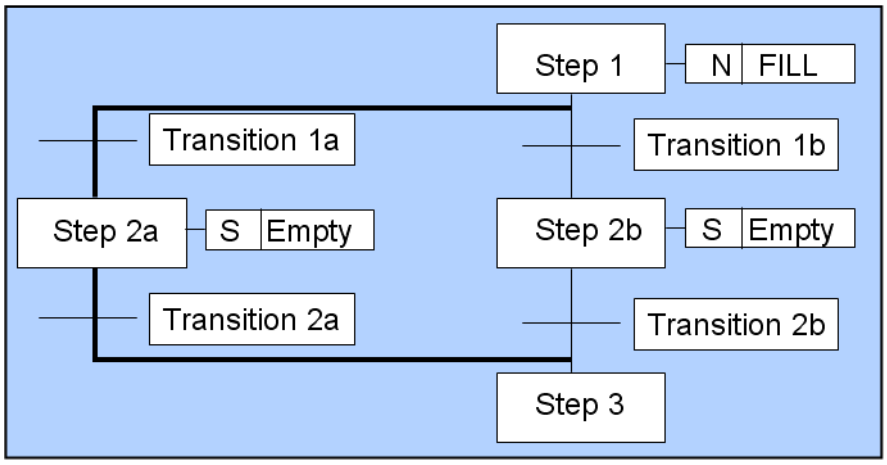


Figure 7-17: Example of a Parallel Sequence in SFC

From Step 1, it either goes to Step 2a or Step 2b, depending on which of the transition conditions is met. Both conditions need to exclude each other.

7.1.13.5 Structured Text (ST Language)

Structured Text (ST) is a powerful, high-level language with its roots in ADA, Pascal, and “C”.

- It contains all the essential elements of a modern programming language, including selection branches (IF-THEN-ELSE and CASE OF) and iteration loops (FOR, WHILE, and REPEAT).
- These elements can also be nested.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- It can be used for the definition of complex function blocks, which can be used within any of the other languages.

7.1.14 Alias Definitions

The compiler supports the definition of aliases.

- An alias is a unique identifier used in programs to replace another text.
  - Definitions are typically used to replace a constant expression and facilitate the maintenance of programs.
  - See usage in (➔ # 523, Use the Defines List) for more information.

There are three levels of definitions:

- Common to all the projects present on your machine
- Global to all programs within your project
- Local to one program

Local definitions are edited together with the corresponding program.

- Definitions are entered in a text editor.
  - Each definition must be entered on one line of text following this syntax:

```
#define Identifier Equivalence (* comments *)
```

Examples

```
#define OFF   FALSE          (* redefinition of FALSE constant *)
#define PI    3.14            (* numerical constant *)
#define ALARM (bLevel > 100)  (* complex expression *)
```

- You can use a definition within the contents of another definition.
  - The definition used in the other one must be declared first.

Example

```
#define PI     3.14
#define TWOPI  (PI * 2.0)
```

- A definition can be empty.

Example

```
#define CONDITION
```

The defined word can be used for directing the Conditional Compiling directives.

TIP

You can enter `#define` lines directly in the source code of programs in IL or ST languages.

The use of definitions can disturb the program monitoring and make error reports more complex. It is recommended to restrict the use of definitions to simple expressions that do not risk creating a misunderstanding when reading or debugging a program.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Word	Description
DATE	• Date stamp of compiling expressed as a string constant expression. • Format is: Year/Month/Day-Hours-Minutes.
MACHINE	Name of the machine where compiling is run, expressed as a string constant expression.
USER	Name of the user where compiling is run, expressed as a string constant expression.

7.1.14.1 Example

This ST program sets variables declared as STRING(255):

```
strDate := __DATE__;
strMachine := __MACHINE__;
strUser := __USER__;

// Result:
// strDate is '2007/11/25-10:45'
// strMachine is 'LaptopJX'
// strUser is 'John'
```

7.1.15 Exception Handling

The compiler is used to write your own exception programs for handling particular system events.

These are the exceptions:

- "Division By Zero Exceptions" (➔ p. 319).
- "Shutdown Exceptions" (➔ p. 320) (after the last cycle).
- "Startup Exceptions" (➔ p. 320) (before the first cycle).

NOTE

Exception handling is supported for both Normal and Optimized PLC code generation.

- In the sub-program that handles the exception you can perform any safety or trace operation.
- Then, select between these possibilities:
  - Return without any special call.
    - In this case the standard handling is performed:
      - A system error message is generated.
      - The result of the division is replaced by a maximum value and the application continues.
  - Call the FatalStop function.
    - The runtime then stops immediately in Fatal Error mode.
  - Call the CycleStop function.
    - The runtime finishes the current program and then turns in "cycle setting" mode.

Handlers can also be used in DEBUG mode for tracking the bad operation.

- Put a breakpoint in the handler.
  - When stopped, the call stack shows the location of the division in the source code of the program.

7.1.15.1 Division By Zero Exceptions

You can write your own exception program for handling the **Division by zero** exception.

In this procedure, you must follow for setting an exception handler:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. Create a new sub-program without any parameter that will handle the exception.
2. In the editor of global defines, insert this line:

```
#OnDivZero SubProgramName
```

7.1.15.2 Shutdown Exceptions

You can write your own exception program to be executed after the last application cycle when the runtime system is cleanly stopped.

1. Create a new main program that handles the exception.
   It cannot be an SFC program.
2. In the editor of global defines, insert this line:

```
#OnShutdown ProgramName
```

⚠WARNING

You cannot put breakpoints in the Shutdown program.

7.1.15.3 Startup Exceptions

You can write your own exception program to be executed before the first application cycle is executed.

1. Create a new main program that will handle the exception.
   It cannot be an SFC program.
2. In the editor of global defines, insert this line:

```
#OnStartup ProgramName
```

⚠WARNING

The program is executed before all other programs within the first cycle.
This implies that the cycle timing can be longer during the first cycle.
You cannot put breakpoints in the Startup program.

7.2 Motion Concepts

7.2.1 Introducing Motion

7.2.1.1 Motion Control Main Functions

To ensure accurate positioning and movement, motion control consists of the two following main parts:

- Setpoint generation
- Regulation

7.2.1.1.1 Setpoint Generation

- This consists of generating a trajectory defined by **position versus time**.
- It is purely logical and does not relate to the physical world.

7.2.1.1.2 Regulation

Even using the very best drives, you cannot maintain accurate positioning without a feedback loop.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The regulation consists of following the generated position settings using classical feed-forward or feedback control-loops (by means of PID).
  - See "How do I implement feedback?" (➔ p. 793) for more information.
- Regulation is the part which takes care of the physical world of making moving motors.

These two functions can be located on the same hardware (as in a "stand-alone" servo drive) or on two separate hardware devices, linked together by a fieldbus.

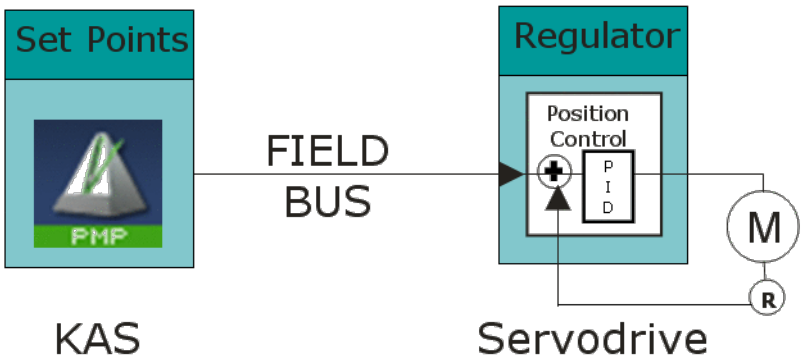


Figure 7-18: Regulation with Remote Drive

7.2.1.2 Single and Multi-Axis Motion

In **Single-Axis**, one setpoint generator is linked to one axis.

Multi-Axis motion consists of synchronizing several axes linked to a common motion source.

This source can be external (e.g., a physical motor (called master) or an internal profile generator (called virtual master)) (Figure 7-19).

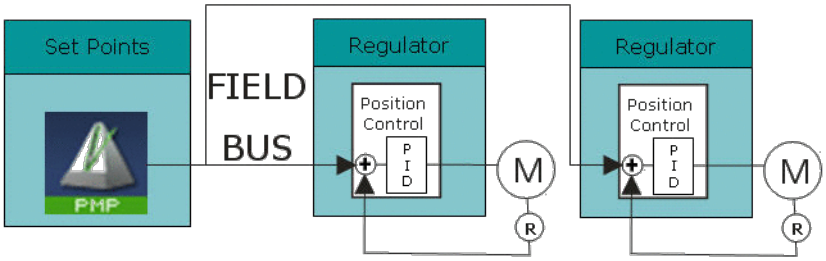


Figure 7-19: Multi-Axis Driven by a Virtual Master

7.2.1.3 Hardware Organization of Motion Functions

A complete motion control chain is made of two main parts that can be subdivided into several more basic functionalities. Depending on your hardware system configuration, each of these elementary functions can theoretically be embedded in different hardware modules.

This image is one of the possible configurations:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

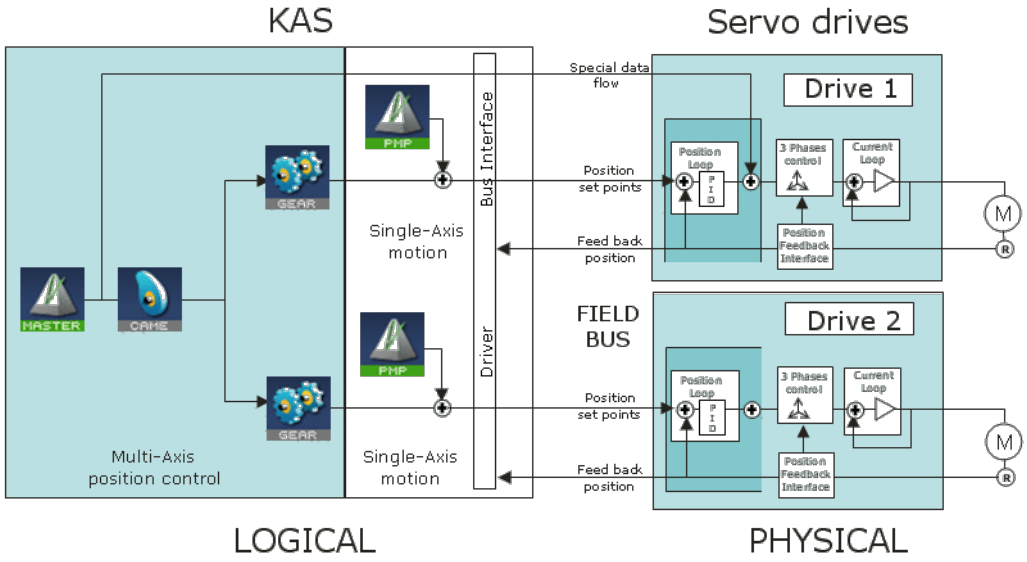


Figure 7-20: Hardware Organization of Motion Functions

The scope of Kollmorgen Automation Suite is to:

- Manage all the logical parts of the motion control.
- Ignore the physical aspects (which are handled by the hardware).

To make the link between the logical and physical worlds, KAS includes some components that acts as interface.

We no longer consider regulation and the physical world. Only "Setpoint Generation" (→ p. 320) is taken into account.

7.2.1.4 Motion Profile

In motion control, a common need is to move a system from one steady position to another (point-to-point motion).

Following the fastest possible motion within an allowed maximum value for speed, acceleration, and jerk, results in a third-order motion profile as shown in this image:

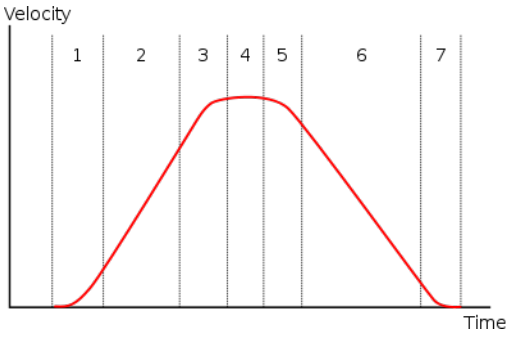


Figure 7-21: Third-order motion profile

The motion profile consists of a maximum of seven phases defined by:

- Acceleration increase with maximum positive jerk.
- Constant maximum acceleration (zero jerk).
- Acceleration decrease, approaching the desired maximum velocity with maximum negative jerk.
- Constant maximum speed (zero jerk, zero acceleration).
- Deceleration increase, approaching the desired deceleration with maximum negative jerk.
- Constant maximum deceleration (zero jerk).
- Deceleration decrease, approaching the desired position at zero velocity with maximum positive jerk.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

If the initial and final positions are sufficiently close together, the maximum acceleration or maximum velocity may never be reached.

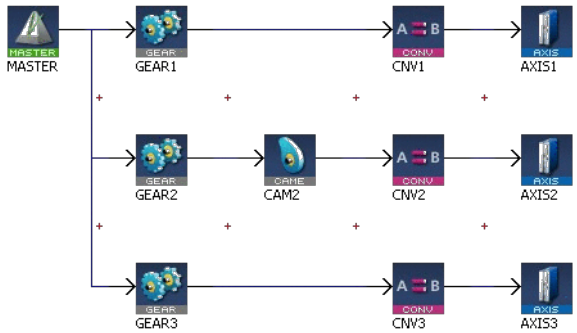
7.2.1.5 Executing Motion

Axis block and Input block are the types of Pipe Blocks are used to command motion in a Pipe Network.

- **Axis block** starts motion directly on one axis.
- **Input blocks** start motion that affect all axes connected downstream in a Pipe Network.
  - Input blocks can be one of three types:
    - Motion / Pipe Network - Master - Trapezoidal motion
    - Motion / Pipe Network - PMP - Parabolic Motion
    - Motion / Pipe Network - Sampler - Externally generated motion from another axis or external encoder.

In this example:

- Executing MLAxisAbs, MLAxisMoveVel, and MLAxisRel functions can be used to cause motion on a particular axis.
- MLMstRun, MLMstAbs and MLMstRel functions cause motion on Axis1, Axis2, and Axis3.



See Also

- "Error Management" (➔ p. 857)
- "Restarting Motion with Pipe Network" (➔ p. 884)

7.2.1.6 Motion State Machine

The Motion State Machine is driven by the IEC 61131-3 application with the help of dedicated function blocks.

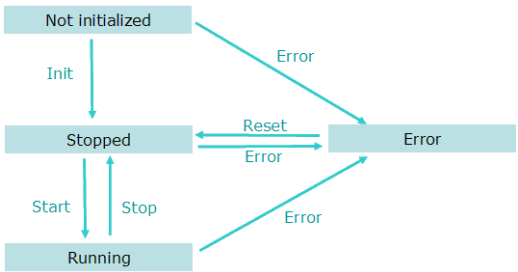


Figure 7-22: Motion State Machine

Each arrow represents a transition from one state to another.

7.2.1.7 KAS Touch Probe and Fast Input - Overview

These are links to KAS documentation helpful in configuring and using the Touch Probe feature, also known as "fast inputs" or "capture engine".

7.2.1.7.1 Configuration

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.1.7.2.1 Drive Configuration

- **AKD:** AKD Support With MC_TouchProbe
- **AKD2G:** AKD2G Support With MC_TouchProbe
- **EtherCAT Configuration for Drives:** The correct input PDOs must enabled for touch probes to work correctly. New KAS projects enable the PDOs needed for most touch probe applications. Some applications need special configurations. Those configurations can be changed here:
"PDO Selection/Mapping tab" (→ p. 554)

7.2.1.7.3.2 Pipe Network Configuration

- MLAxisCfgFastIn
- MLTrigSetEdge

7.2.1.7.4.3 PLCopen Configuration

- MC_TouchProbe
- MC_Reference
- MC_MachRegist

7.2.1.7.5 List of Function Blocks and Pipe Blocks

7.2.1.7.6.1 PLCopen Function Blocks

I/O

- MC_TouchProbe

Reference

- MC_Reference

Registration

- MC_MachRegist
- MC_MarkRegist
- MC_StopRegist

7.2.1.7.7.2 Pipe Network Blocks

Motion / Pipe Network - Axis

- MLAxisCfgFastIn
- MLAxisIsTrigged
- MLAxisRstFastIn

Motion / Pipe Network - Trigger

- MLTrigClearFlag
- MLTrigInit
- MLTrigIsTrigged
- MLTrigReadDelay
- MLTrigReadPos
- MLTrigReadTime
- MLTrigSetEdge
- MLTrigWriteDelay

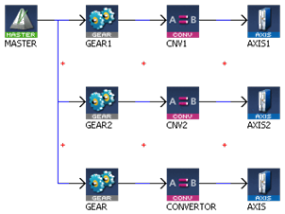
7.2.2 Pipe Network or PLCopen

Using KAS these are the ways to generate motion functions and motion profiles:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Pipe Network" (➔ p. 325)
- "PLCopen" (➔ p. 325)

7.2.2.1 Pipe Network



The Pipe Network is used to create a high-performance motion algorithm which is tightly integrated to the PLC program with motion library function blocks.

- For high-performance, complex, or synchronized multi-axis applications, the pipe concept in KAS provides a simple conversion of mechanical applications into a graphical representation of application elements and the process flow.
- This format makes it easy to understand, program, and update the motion profiles and positional relationships.
- The KAS application begins with the creation of a Pipe Network structure linking Master objects (source) to Axes objects (destination).
  - This includes the definition of specific transformer motion profiles.
  - This structure is controlled from the PLC application using dedicated function blocks in the Motion Library.
- To be able to use pipes correctly, it is necessary to consider the "Pipe Network Concept" (➔ p. 326).

7.2.2.2 PLCopen



- Standard function blocks can be used and directly incorporated into the PLC application.
- Programming of motion is done using standard MC function blocks that can be incorporated in single-axis or multi-axis applications.
- See the PLCopen Web site.

7.2.2.3 Pipe Network and PLCopen Motion Engine Differences

This table outlines some of the main feature differences between the Pipe Network and the PLCopen motion engines and their associated function blocks.

Topic	Pipe Network	PLCopen
Axis setup method	Includes in the Pipe Network Axis block properties. See Motion / Pipe Network - Axis.	Part of Axis definition screen in the Project tree.
Does a Function block requires instantiation?	No except for MLAxisStop.	Most require it.
Function block format	Begins with **ML** (e.g., MLAxisRel)	Begins with **MC_** (e.g., MC_MoveRelative)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Topic	Pipe Network	PLCopen
Function block standard input format	Requires additional function blocks to define motion parameters (e.g., speed, accel, decel, etc.)	Includes standard set of inputs to define motion (e.g., speed, accel, decel, etc.)
How the Axis name is setup?	Automatically done as part of Pipe Network Axis block properties. See Motion / Pipe Network - Axis.	1. Create an instance of a Axis_Ref variable structure in the dictionary. 2. Assign an axis number to it in a PLC program.
Is there additional motion editor?	Yes (Pipe Network editor)	No
Method to start execution	Most are level triggered.	Most are edge triggered.
Motion buffering	Execution of multiple motion commands in a row is handled by the programmer.	Function blocks have built in buffering modes.
Motion execution status for function block executing motion	Use the MLMotionStatus function block.	Each function block includes a standard set of outputs for motion status.
Motion jerk reduction	Primarily available by adding cams to the Pipe Network.	Function blocks have jerk reduction input.

7.2.3 Pipe Network Concept

To introduce the Pipe Network concept, we can use a mechanical analogy.

In this figure, the mechanical system is composed of three-axes and driven by one motor.

- All axes are connected to the motor through shafts, gears and cams.
- When the motor is in motion, all axes are moving synchronously.
- The speed relation between the Master and the Axis is achieved by using a mechanical Gear.
- A mechanical cam is used to get linear motion from a rotating wheel.

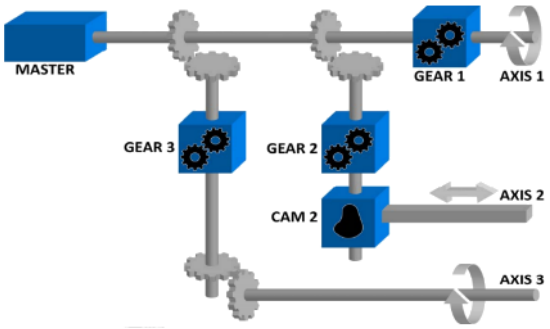


Figure 7-23: Mechanical System

The Pipe Network in this figure corresponds to the mechanical system described above.
The pipe concept is a one-to-one translation of a mechanical system into the logical world.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

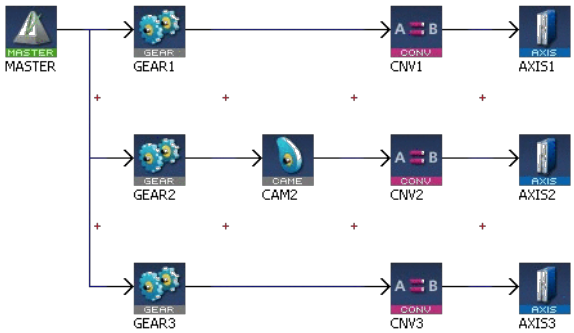


Figure 7-24: Pipe Network Structure

In Pipe Network, the analogy is:

- The main motor of the mechanical machine becomes a Virtual Master Pipe Block.
- The gear boxes becomes Gear Blocks.
- The mechanical cam becomes a Cam Block.
- The axes becomes Axis Blocks.

The Pipe Network concept allows motion engineers to define, in a very natural way, the physical relationships between the different axes of their machine.

This modular approach provides a solution for almost any multi-axis requirements. It remains open for new, additional functions that can be required in the future.

7.2.3.1 Pipe Network

To control the machine application with multi-axes that are dynamically interconnected, you can design several pipes with the KAS-IDE to create the global Pipe Network.

- See Figure: "**Pipe Network Structure**".
- Relationships between the axes are developed and connected graphically, allowing you to visualize how the machine functions.
- Each horizontal flow is considered as a separate pipe.
    - Figure 7-25 shows three pipes.

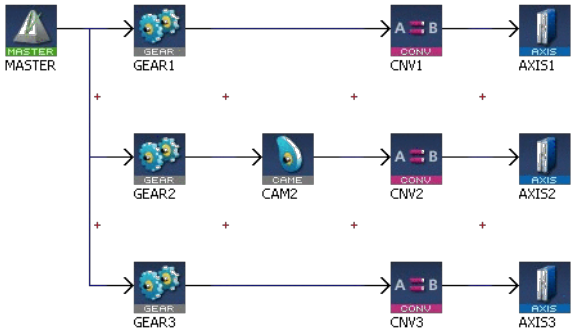


Figure 7-25: Three pipes

- The Pipe Network can be edited at any time.

NOTE

You do not have to finalize the Pipe Network before writing a PLC program.
You must compile the project to have the latest Pipe Network information available in the PLC program editor.

- Program code does not have to be written when setting up the foundation of a program because the parameters are entered in set-up screens.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

IMPORTANT

Pipe Network code is generated automatically by the compiler.
Do not try to modify it.

- In the programs, define activation or deactivation statements to install or remove pipes and Pipe Blocks.
  - This allows the dynamic adjustment of the machine behavior depending on the result.
- The Pipe Network contains a full library of single-axis motion commands for sections of an application where an axis operates independently.

7.2.3.2 Pipe Structure

A pipe is a set of Pipe Blocks linked together (where position flows from one Pipe Block to the next). (Figure 7-26)

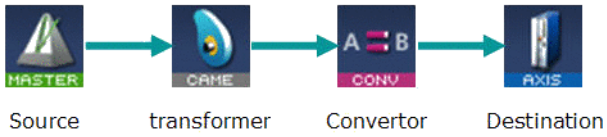


Figure 7-26: Typical Pipe Structure

The general structure of a pipe is:

1. Start with an input Pipe Block (source).
2. Optionally followed by transformer Pipe Blocks.
3. Followed by an output Pipe Block (convertor).
4. Finish with the destination Pipe Block.

TIP

Jerk may ultimately cause a jerk in motor motion when a cam block is applied to the upstream pipe network positions.
To avoid jerk in the pipe network, the potential position offset between the cam's first point and the input to the cam block must be taken care of in the application program by setting a cam offset or another method.

Figure 7-27 shows an option for velocity instead of position using a differential Pipe Block:

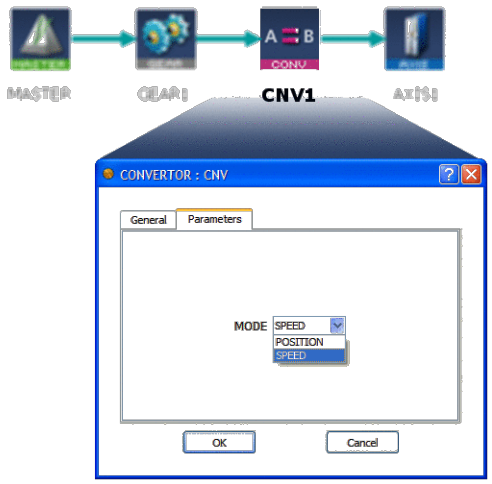


Figure 7-27: Velocity option

See (→ # 187, Add Motion) about the different kinds of Pipe Blocks.

7.2.3.3 Pipe Block

Pipes are built using logical entities called Pipe Blocks.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- A Pipe Block:
  - Is an object whose purpose is to modify a flow of values with strict time constraints.
  - Normally has both input and output flows of values.

This table has the different Pipe Blocks, based on their functions:

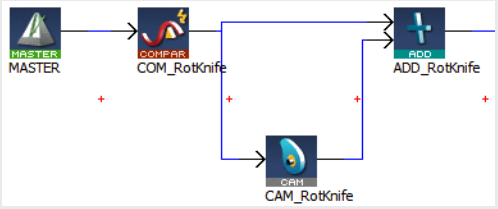
Function	Description
Destination	Models a physical axis of the machine.
Input (source)	Works as generator of values: • Sample external source objects. • Create a discrete flow of values as an input to the pipe.
Output (convertor)	Block that can end a pipe: • Convert the incoming values from user units to correct system units for the destination objects.
Transformer	• Apply a specific algorithm to the input value to produce their output. • Transformations can be linear or complex (e.g., cam). • Can create events depending on the incoming values.

7.2.3.3.1 Pipe Network - List of Pipe Blocks

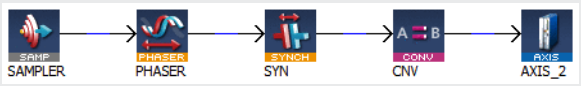
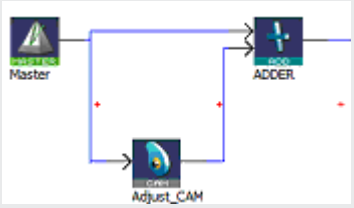
This table provides a description of each Pipe Block.

Pipe Block	Function	Link to Pipe Block Function Overviews	Description
Adder	Mathematical	Motion / Pipe Network - Adder.	• Adds two data flows. • Use an Adder block with a Cam block to dynamically change the distance moved during each period (or modulo) of motion.
Axis	Destination	Motion / Pipe Network - Axis	Models the link from the Pipe Network to a physical axis.
Cam	Modification	Motion / Pipe Network - CAM Profile	• Use a Cam Pipe Block to optimize the motion profile. • Applies a cam table (also called Cam Profile) to the input data flow.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Pipe Block	Function	Link to Pipe Block Function Overviews	Description
Comparator	Event-driven	Motion / Pipe Network - Comparator	• Monitor the input data flow and detects the crossing of a particular value. • By tracking the position at one point of the Pipe Network, you can use a Comparator Pipe Block to synchronize when code is executed in a PLC application program. Click to view an example Figure 7-28 shows the changing of the offset move by changing the amplitude (or offset) of the Cam Pipe Block.  **Figure 7-28:** Example: Offset move In a PLC application program, the: • MLCompWriteRef function block is used to arm the comparator block. • MLCompCheck function block is used to check the position. • By using condition statements in a user program, specific actions (e.g., changing the move distance of the offset) can then be taken. Another example shows the use of a Comparator Pipe Block to determine if a high-speed input is within the acceptable position range.
Convertor	Output	Motion / Pipe Network - Convertor	Converts input data flow to a position and forwards it to an axis.
Delay	Event-driven	Motion / Pipe Network - Delay	• Delay the data flow during some cycles. • Use a Delay Pipe Block to delay the flow of position through a Pipe Network. **Example** • Place it before a Trigger block in a pipe which is not connected to a drive. • There is a delay of five servo update cycles between the dynamic position in the Pipe Network and the triggering of a high-speed digital input.
Derivator	Mathematical	Motion / Pipe Network - Derivator	Applies a derivation on the input data flow.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Pipe Block	Function	Link to Pipe Block Function Overviews	Description
Gear	Modification	Motion / Pipe Network - Gear	• Applies a gearing ration on the input data flow. • Use a Gear Pipe Block to perform electronic gearing. • This block allows gear ratios and the slope of the gear change to be initially set, then changed from in the application program.
Integrator	Mathematical	Motion / Pipe Network - Integrator	Integrates the input data flow.
Master	Input	Motion / Pipe Network - Master	Virtual master generating values (position) at each cycle.  Use a Master Pipe Block to create a virtual master to link two or more axes. • The Profile generator in the Master block is trapezoidal. • If a parabolic type profile is required, use a PMP Pipe Block. • If the master is an external encoder or another axis, use the Sampler Pipe Block.
Phaser	Modification	Motion / Pipe Network - Phaser	Applies a phase offset to the input. Use a Phaser Pipe Block to: • Perform a dynamic phase adjustment inside the Pipe Network. • Phase-advance or phase-retard a position as required to synchronize different motion elements on a machine.
Sampler	Input	Motion / Pipe Network - Sampler	Samples external value (encoder, resolver, PLC variable etc.)  Use a Sampler Pipe Block to: • Read an external encoder as an input signal into the Pipe Network. • Directly read the actual position of another axis.  Cam Profiles are created using the "Cam Profile Editor" (→ p. 695).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Pipe Block	Function	Link to Pipe Block Function Overviews	Description
Synchronizer	Event-driven	Motion / Pipe Network - Synchronizer	Starts and stops a sub-pipe in a controlled way. • Use a Synchronizer Pipe Block to synchronize two axes. • This Pipe Block is useful in applications where it is necessary to start the motion of a second axis and sync to the first.
Trigger	Event-driven	Motion / Pipe Network - Trigger	Computes the local pipe value from the timestamp of a Fast Input event.  • Use a Trigger Pipe Block to read the position when a high-speed input is triggered on the machine. • The trigger block allows you to catch the position at a particular location in the Pipe Network, as required by the application.

7.2.3.3.2 Change the Information Flow from Position to Velocity

Use the **Convertor** Pipe Block to change the Pipe Network flow of information from position to velocity.

This block is normally set up to a Receive position, so it must be changed to receive the expected input signal type shown in this image:

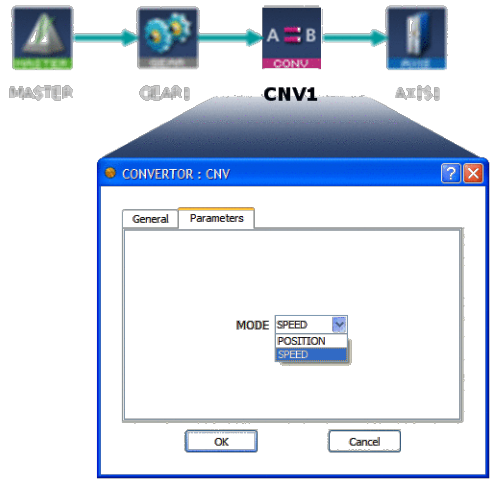


Figure 7-29: Change the Mode of Converter Block to SPEED (not POSITION mode).

7.2.3.4 Axis Pipe Block

The Axis pipe block is the link between the logical and physical worlds, managing position data for a physical or simulated axis.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

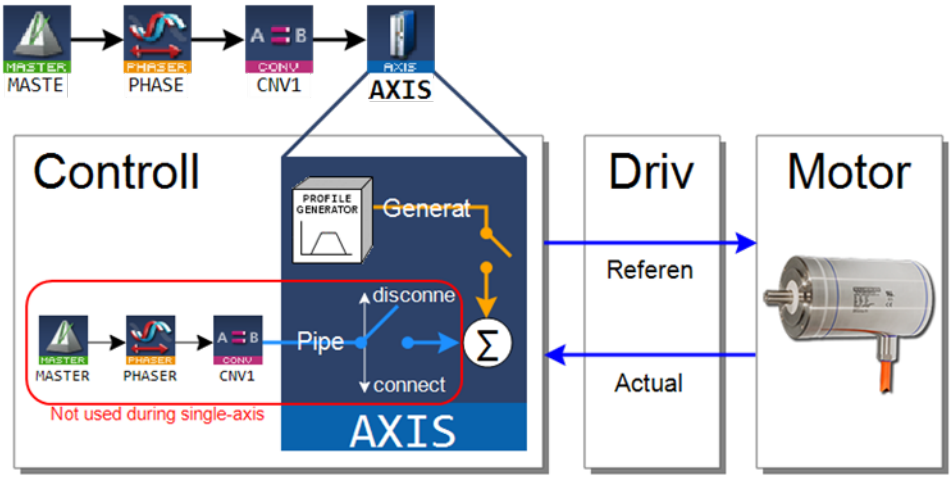


Figure 7-30: Axis Pipe Block Positions

7.2.3.4.1 Associated Data on Positions

This data are illustrated here:

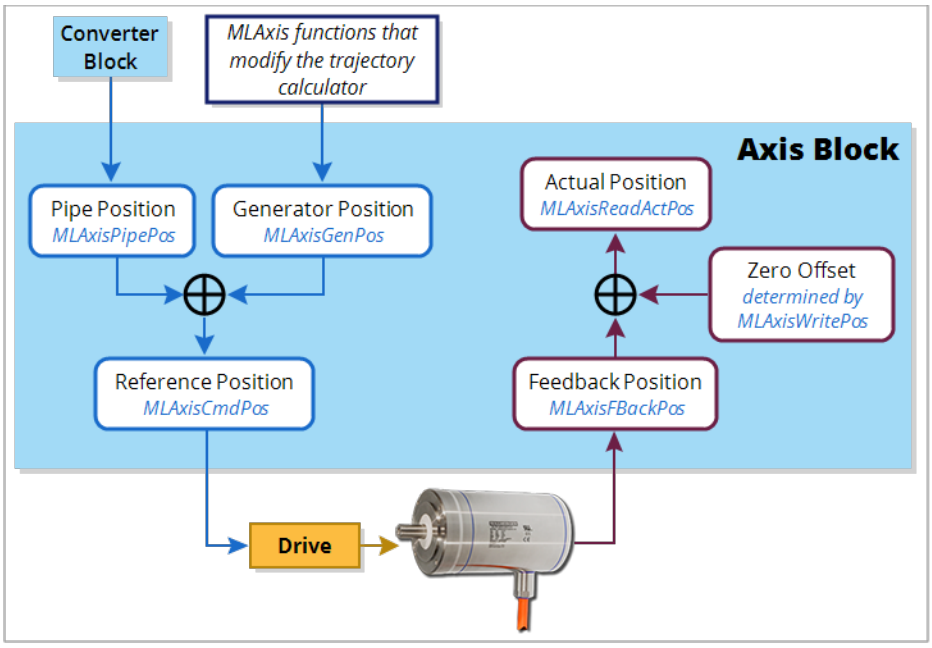


Figure 7-31: Axis Block Positions Diagram

NOTE

All positions are in user units with modulo applied if active, unless specified.

Position / Offset	Description
Actual Position	This is the actual position of the underlying axis as reported by the drive. • It is the sum of the feedback value (Position actual value) returned from the communication link to the drive and any zero-offset due to an MLWritePos function block. • See either MLAxisWritePipPos or MLAxisWritePos. `ActualPos := FeedbackPos + ZeroOffset`

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Position / Offset	Description
Feedback Position	• This is the current position the drive reports for an axis, scaled to user units. • It does not take into account the value of the Zero Offset or axis modulo.
Generator Position	This is the summation of all previous commands (i.e., calls to functions which perform motion) to the Axis internal motion generator. See either such as MLAxisAbs, MLAxisMoveVel, or MLAxisRel. • It is modified by MLAxisWritePos to insure no jumps in the **Reference Position** command. • It accumulates changes in pipe position due to activate and deactivation of the pipe the Axis block is associated with.
Pipe Position	• This is the output of the convertor block is written into the **Pipe Position** value whenever the Convertor block is connected to the axis and the pipe is active. • See Motion / Pipe Network - Convertor.
Reference Position	• This is the commanded axis position sent to the drive. • It is the summation of **Pipe Position** and **Generator Position**. • ReferencePosition = Pipe Position + Generator Position
Zero Offset	This adjusts the coordinate system so the Actual Position reports correct values after homing or using MLAxisWritePos.

7.2.3.4.2 Axis Block Initialization

A call to the MLAxisInit function block is required to implement motion for the axis.

- All positions and offsets are set to zero.
- The Axis Block motion generator is initialized with the proper ranges.
- Values are aligned: `ReferencePosition = Pipe Position + Generator Position`.

7.2.3.4.3 Axis Connection to a Pipe

A call to the MLCNVConnect function block is required to get motion generated in the pipe to the Axis.

- Pipe Offset calculation is: `Pipe Offset = Pipe Position - Reference Position`.
- Values are aligned: `ReferencePosition = Pipe Position + Generator Position`.

7.2.3.4.4 Change Axis Position Functions

Function	Description
MLAxisAbs	Performs a move to an absolute position.
MLAxisAdd	A selected Axis performs a move for a specified distance relative to the endpoint of the previous move.
MLAxisMoveVel	Jog at the specified speed.
MLAxisReAlign	Realigns the actual position with the reference position by moving the axis by the specified delta position.
MLAxisRel	A selected Axis performs a move for a specified distance relative to the current position.
MLAxisStop	Stop with the specified deceleration. This does **not** remove the input source, but does stop the drive from continuing to move.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Function	Description
MLAxisWritePipPos	Forces the pipe position internal value. **IMPORTANT** This function works only when no pipe is connected.
MLAxisWritePos	Sets a new value to an axis' current location.
MLCNVConnect	Connects a converter Pipe Block to the specified axis.
MLCNVDisconnect	Disconnect a converter Pipe Block from its associated axis.
Motion / Pipe Network - Pipe	Starts calculating Pipe data. If the convertor block is connected to the axis it will reconnect the convertor and start updating the pipe position.
Motion / Pipe Network - Pipe	Stops sending Pipe data to the Axis Block Pipe Position and disconnects the convertor output from the axis.

7.2.3.4.5 Homing

Homing is the process of moving the motor to a known physical reference point on the machine.

- Homing can be performed with the **MLFB_Home*** Kollmorgen or by adding custom application code that uses only the Motion / Pipe Network - Axis function blocks.
  - The Pipe Network is typically not used in homing.
- Custom homing is done with MLAxisRel and MLAxisAbs to make motion and MLAxisWritePos to set a position offset.

7.2.3.4.6 Monitor an Axis

These are the function blocks to monitor the performance and status of an axis.

Function	Description
MLAxisCmdPos	Returns the reference position of the axis.
MLAxisGenIsRdy	Verifies if an axis is ready.
MLAxisReadActPos	Returns the actual position of the axis.
MLAxisReadGenStatus	Returns the status of the internal generator of the axis.
MLAxisStatus	Returns the status of the axis.

7.2.3.4.7 Multi-Axis Operation

For multi-axis applications, automatic operation requires both:

- Motion synchronization between two or more axes.
- The Pipe Network is required to achieve the synchronization.

To start up the Pipe Network, these two function calls must be executed in an application program:

```
PipeNetwork(MLPN_ACTIVATE):
PipeNetwork(MLPN_CONNECT);
```

Multi-axis synchronized motion is accomplished using a motion block associated with one of these input Pipe Blocks:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Master: MLMstRun, MLMstRel, and MLMstAbs
- PMP: MLPmpAbs, MLPmpRel
- Sampler: MLSmpConnect, MLSmpConECAT, MLSmpConPNAxis, MLSmpConPLCAxis

7.2.3.4.8 Realign Positions

A call to the MLAxisReAlign function block is used to realign the axis after an error occurs

- Motion must come to a stop first.
- The MLAxisReAlign is executed.
  You must set the movement of this block to MLAxisReadActPos - MLAxisCmdPos.
- The target position must be reached before any additional motion can occur.
  It can be checked by using the MLAxisReAlgnRdy function block.

7.2.3.4.9 Set Zero Axis

A call to the MLAxisWritePos function block is used to set a position offset at the Axis when the Pipe Network is not yet connected.

- Pipe Position and Pipe Offset are set to zero.
- Generator Position is set to equal to Zero Position.
  Zero Position is defined in MLAxisWritePos function block.
- Then `ReferencePosition = Pipe Position + Generator Position.`

7.2.3.4.10 Single-Axis Operation

This includes motion done on an individual axis block: jogging, absolute move, or incremental moves.

- If these are single-axis based, then motion is executed with the MLAxisMoveVel, MLAxisAbs, and MLAxisRel FBs.
  - These motions are typically done during machine setup or adjustment and are often referred to as manual mode.
  - For these operations, the Pipe Network does not need to be connected to the axis.

7.2.3.4.11 Units

User Units (abbreviated UU) are dimensions used in the application code to define axis position or distance for an axis.

The **User Units per Revolution** field is used to set how many of these custom units are associated with one revolution of the motor.

Examples

User Units	User Units per Revolution Value	Notes
Degrees	360.0	
Gradians	400.0	
Inches	15.7079632679489662	This equals pi*5. This could represent the circumference of a 5 inch diameter drum.
Radians	3.14159265358979324	

KAS standard units for velocity and acceleration are UU/sec and UU/sec^2.

7.2.3.4.12 View Axis Block Position Functions

Function	Description
MLAxisCmdPos	Returns the reference position of the axis.
MLAxisFBackPos	Returns the feedback position of the axis.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Function	Description
MLAxisGenPos	Returns the generator position of the axis.
MLAxisPipePos	Returns the pipe position of the axis.
MLAxisReadActPos	Returns the actual position of the axis.

7.2.3.5 Pipe Block Lifetime

7.2.3.5.1 Activation

- The pipe is activated when the output of the Convertor Pipe Block is connected to its related Axis.
- All characteristics are reset to the declaration values and the history of the block begins.

7.2.3.5.2 Usage

- As long as the pipe remains active, its values are cyclically calculated.
- Functions can be performed and events can be created.

7.2.3.5.3 Deactivation

- The pipe is deactivated when the deactivate function is applied to the pipe.
- All internal current values are lost and the block no longer exists.

7.2.3.6 Pipe Network Phase Execution

7.2.3.6.1 Absolute Phase of Execution

The absolute phase of execution of a pipe is the elapsed time between any fixed reference and the next computation for the specified pipe.

7.2.3.6.2 Relative Phase of Execution

The relative phase of execution between two pipes is the elapsed time between the computation of the first pipe and the second one.

- The relative phase of execution between two pipes of the same Pipe Network is 0 (zero).
- The phase of execution between two Pipe Network cannot be specified by the user and depend on the pipe activation time of the application execution.

7.2.3.7 Use Motion Function Block for Pipe Network

Use motion library function blocks in your PLC application program to interface to the Pipe Network (see "Design Motion with Pipe Network" (➔ p. 187)).

ML function blocks are used to:

- Create and initialize the Pipe Network.
- Perform motion at a single-axis or multi-axis level.
- Read information from points in the Pipe Network.

  Example:

  - Read a high-speed input position from a Trigger Pipe Block.
  - Read Command or Reference position from an Axis Pipe Block.
  - Determine when a position has been reached in a Comparator Pipe Block.
- Modify how the blocks work in the Pipe Network.

  Example:

  - Change the phase offset of Phaser Pipe Block.
  - Change the amplitude or offset of a CAM Profile.
  - Change the speed of a Master Pipe Block.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.3.7.1 Buffer Mode

With the Pipe Network engine, when a motion function block is executed while another one is presently executing, there is an immediate change.

That means the previous function block is aborted and the new one immediately becomes the active move and begins executing.

7.2.3.7.2 Motion Init

During initialization, and using the MLMotionInit function, the IEC 61131-3 application can create the different motion objects it needs (pipes, blocks, axes):

- Pipe Create
- Profile Create

NOTE

When the state machine leaves the Init state, the creation of new motion objects is no longer allowed. This is to avoid memory allocation problems while running the application.

7.2.3.7.3 Motion Start

- The Start method (MLMotionStart function) initializes the motion engine, motion bus driver, and initializes the EtherCAT network to operational mode.
  - MLMotionStopbus driver deactivates the execution of the motion engine.
- The function blocks MLMotionStart, MLMotionStop, and MLMotionRstErr can be used by the IEC 61131-3 applications to navigate between states (i.e., Not initialized, Running, Stopped and Error).

7.2.3.7.4 Using the Q Output of ML Function Blocks for the Pipe Network

There is a Q output on most ML function blocks.

- The operation of the Q output is different for different ML function blocks.
- The Q output can be useful in PLC application programs.

Examples:

- MLAxisRel.Q is set when function block successfully executes.
- MLAxisStop.Q is set when motion is stopped (zero speed).
- MLPrfWriteIOffset.Q is set if cam offset has been changed to the new value.

See Also

- "What is the Difference between Q and OK?" (→ p. 339)

7.2.3.8 Pipe Network - General Rules

This section outlines rules for using ML function in the Pipe Network context.

7.2.3.8.1 Languages

Functions that interact with the Pipe Network start with ML (e.g., MLAxisRel, MLPrfWriteOScale, or MLMstRel).

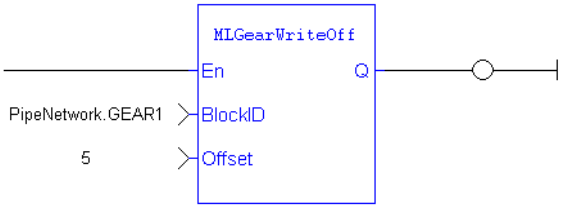
These functions can be used in all the Programming Languages.

7.2.3.8.2 BlockID Inputs

The BlockID input is a DINT ID.

It is the second input to a Pipe Network function when using the FFLD Language:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



The BlockID input is the first one if programming in the ST Language:

```
MLGearRatSlp( BlockID (*DINT*) );
```

This input identifies the block in the Pipe Network the function interacts with.

If using the graphical "Pipe Network Editor" (→ p. 733), the used variable starts with **PipeNetwork.xxx**.

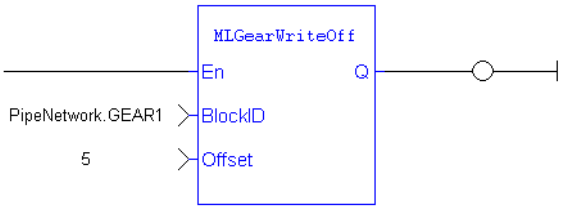
The exception is if it is acting on a CAM profile, then the input is named ProfileID and the variable starts with **Profiles.xxx**.

TIP

- When selecting a Pipe Block as the BlockID for an ML function, choose a Pipe Block with the same type which is in the name of the ML function.
- Example: MLMstxxx functions expect a Master block to be chosen for the BlockID.
  - MLAxisxxx functions need an Axis block to be chosen for the AxisID input,
  - MLPrfxxx functions need a Profile entered for ProfileID, etc.

7.2.3.8.3 Output Status

Most Pipe Network functions have a default Boolean output labeled either **Q** or **OK**:



7.2.3.8.4 What is the Difference between Q and OK?

- **OK** returns True when the function successfully executes.
- **Q** output is initially set to 0 (zero) until the first time the block is successfully executed in a running program.
  - After this execution, the Q output is set to 1.
  - It remains to 1 until the function does NOT execute successfully.
  - Alternately, after an unsuccessful execution the Q output is set to 0 (zero).
  - It remains to 0 (zero) until a successful execution resumes.

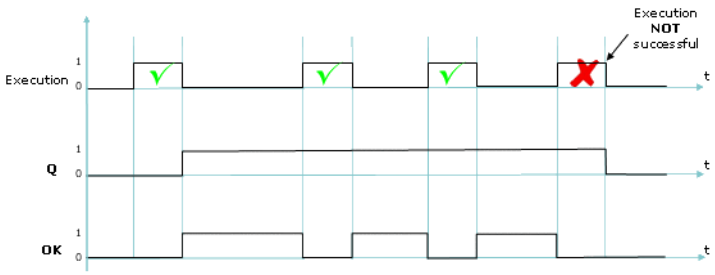


Figure 7-32: Execution Not Successful

7.2.3.8.5 When Q is Set to TRUE

Verify the description for each individual function block to be sure how its Q output is behaving.

- Some functions change the Q output from low to high immediately after it starts executing.
- Others (including most functions that command motion) wait to change the output until the function has completely finished executing.

7.2.3.8.6 Input Parameters

- The En input parameter, which is used to execute the function, is not edge-triggered.
  - If a function is seen in the PLC code and its En input is positive, the function executes.
- Example: A MLAxisRel command continuously executes relative moves in a program if it is called each program cycle; thus it acts as a Run/Jog command if continually commanded.

7.2.3.8.7 Missing Input Parameters

All inputs to a function must be entered for code to compile.

7.2.3.8.8 Position versus Distance

- **Position** is a value defined in a coordinate system.
- **DeltaPosition** is a relative measure related to technical units.
  - It is the difference between two positions.

7.2.3.8.9 Default Block Parameters

The parameters set when initializing a Pipe Network block are used as defaults when calling functions.

- These parameters can be modified in a program by using specific functions to set these values.
  - If a value is never set in a program the parameter entered during initialization is used.
- Example: When making a Master relative move (MLMstRel) you input the DeltaPosition but not the velocity or acceleration.
  You can set the velocity for the move by using the MLMstWriteSpeed function before calling the relative move.
  If the speed is not set in the program, the default parameter entered during the initialization (i.e., in the properties dialog box of the Pipe Blocks) is used.

7.2.3.9 Use the Generated Pipe Network Function

When the "Pipe Network Editor" (➔ p. 733) is used:

- The KAS-IDE automatically creates a subprogram named **PNCode** and an instance of that subprogram named **Pipe Network**.
- These are found in the Dictionary under **Global variables**. (Figure 7-33)

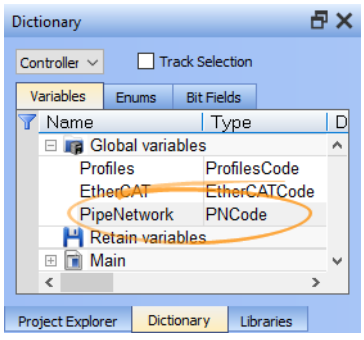


Figure 7-33: PipeNetwork and PNCode

The Pipe Network instance contains the IDs for all the pipe blocks created in the Pipe Network Editor.

It can execute certain actions when called as a function.

These functions are:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Pipe Network(MLPN_CREATE_OBJECTS) – Creates all Pipe Network related objects. This needs to be called first, before any other access to the Pipe Network instance. The Pipe Network template applications make this call in the **Main** program.
- Pipe Network(MLPN_ACTIVATE) – Activates all pipes defined in the Pipe Network Editor by calling Motion / Pipe Network - Pipe on each pipe.
- Pipe Network(MLPN_DEACTIVATE) – Deactivates all pipes defined in the Pipe Network Editor by calling Motion / Pipe Network - Pipe on each pipe.
- Pipe Network(MLPN_CONNECT) – Connects all converter blocks defined in the Pipe Network Editor to their associated axis blocks by calling MLCNVConnect on each converter block. This is typically done after all homing is completed.
- Pipe Network(MLPN_DISCONNECT) – Disconnects all converter blocks defined in the Pipe Network Editor from their associated axis blocks by calling MLCNVDisconnect on each converter block. This is typically done when axes need to be re-homed.
- Pipe Network(MLPN_POWER_ON) – Powers up (Enables) all axes defined in the Pipe Network Editor, enabling any associated servo or stepper drives mapped to the axes. This is done by calling MLAxisPower on each axis block.
- Pipe Network(MLPN_POWER_OFF) – Powers down (Disables) all axes defined in the Pipe Network Editor, disabling any associated servo or stepper drives mapped to the axes. This is done by calling MLAxisPower on each axis block.

The IDs for the blocks used in the Pipe Network Editor are needed when writing code that accesses those function blocks. For example, when you need to read the actual position for an axis, you will need to use the pipe block ID in MLAxisReadActPos:

```
// Structured Text Example
// Syntax = Function Name (PN Instance Name. Pipe Block ID)
MLAxisReadActPos(PipeNetwork.AXIS1);
```

The IDs for the pipe blocks can be found in the dictionary under the **PNCode** subprogram but are accessed through the Pipe Network instance. (Figure 7-34)

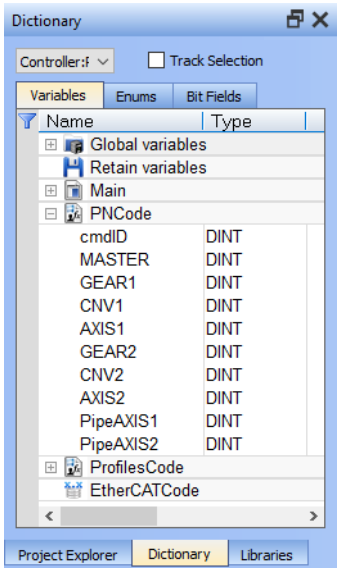


Figure 7-34: PNCode Dictionary

7.2.4 PLCopen

Kollmorgen Automation Suite supports the International PLCopen motion standard.

- The PLCopen international standard was created to obtain uniformity of motion function blocks and motion startup between machine control products.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Users familiar with PLCopen are automatically familiar with PLCopen inside the KAS-IDE.
- The PLCopen function blocks for Motion Control, is based on the IEC 61131-3 "Function Blocks (POU)" (➔ p. 307) concept with these factors in consideration:
  - Completeness
  - Consistency - conforming to the IEC 61131-3 standard
  - Efficiency - in the number of function blocks, for efficiency in design (and understanding)
  - Flexibility - future extensions / range of applications
  - Simplicity - ease of use for the application program builder and installation & maintenance
  - Universality - hardware independent

KAS supports PLCopen motion in these program formats: FBD, FFLD, IL, SFC, and ST.

- PLCopen blocks in KAS start with MC (e.g., MC_MoveAbsolute).
  - MC blocks are an alternative to using the ML Motion function blocks (e.g., MLAxisAbs) and associated Pipe Networks in many applications.
- Using MC Motion function blocks does **not** require a separate motion editor.

7.2.4.1 PLCopen Function Blocks

This function block (FB) library is designed for controlling one or more servo axes using the IEC 61131-3 PLCopen standard. See "Function Blocks (POU)" (➔ p. 307).

To offer flexibility, ease of use, and reusability, the library consists of command-oriented function blocks with a reference to the axis (e.g., the abstract data type "AXIS_REF Structure" (➔ p. 371)).

The PLCopen Library contains function blocks for:

- **Control**: Function blocks to define and initialize motion, control power, and reset errors.
- **I/O**: Function blocks to control interaction with Digital I/O and Touch Probe and trigger "PLCopen Registration" (➔ p. 875) functionality.
- **Info**: Function blocks to provide information on motion, position, status, and the ability to read and write other drive parameters.
- **PLCopen Motion**: Function blocks to execute different types of motion.
- **Profile**: Function blocks for master/slave motion.
- **Reference**: Function blocks to reset position.
- **Registration**: Function blocks to perform registration.

Function Block	Description
MC_ AddSuperAxis	Adds a Superimposed Axis to the Axis's list of assigned superimposed axes.
MC_CamIn	Performs a slave axis move which follows the master axis based on the Cam Table specified by CamTableID.
MC_CamOut	Disengages the slave axis from an MC_CamIn move or the master axis immediately in a cam block.
MC_ CamTblSelect	Defined to read and initialize the specified profile. Selects the Cam tables by setting the pointers to the relevant tables.
MC_GearIn	Performs a slave axis move which follows the master axis using the ratio specified by RatioNumerator and RatioDenominator. Once engaged, the slave moves RatioNumerator units for every RatioDenominator units the master moves.
MC_GearOut	Disengages the slave axis from a MC_GearIn or MC_GearInPos move or the master axis.
MC_MachRegist	Enables Mark-to-Machine registration.
MC_MarkRegist	Enables Mark-to-Mark registration.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Function Block	Description
MC_ MoveAbsolute	Performs a single-axis move to a specified endpoint position.
MC_ MoveAdditive	Performs a single-axis move for a specified distance from the endpoint of the previous move.
MC_ MoveRelative	Performs a single-axis move of a specified distance relative to the actual position at the time of the start of execution.
MC_ MoveSuperimp	Performs a single-axis move superimposed upon the currently executing move.
MC_ MoveVelocity	Performs a single-axis, non-ending move at a specified velocity.
MC_Power	Requests to either enable the drive and close the loop, or disable the drive and open the loop.
MC_ ReadActPos	Reads the actual position of the axis.
MC_ ReadAxisErr	Returns the error status of the specified axis.
MC_ ReadBoolPar	Returns the value of the specified Boolean axis parameter. Returns the value of a drive parameter. • The returned value has to be converted to REAL if required. • If not possible, the vendor must provide a supplier-dependent FB for it.
MC_ReadParam	Returns the value of the specified axis parameter. Returns the value of a drive parameter. • The returned value has to be converted to REAL if required. • If not possible, the vendor must provide a supplier-dependent FB for it.
MC_ReadStatus	Returns the state of the specified axis.
MC_ RemSuperAxis	Remove an axis from the axis's list of assigned, superimposed axes.
MC_ResetError	Resets the errors of the specified axis.
MC_Stop	Aborts the active move, removes the next move from the queue, performs a controlled stop, and switches the axis to Stopping state.
MC_StopRegist	Turns off registration for the specified axis and disarms the specified fast input.
MC_ WriteBoolPar	Writes the specified axis Boolean parameter.
MC_WriteParam	Writes the specified axis parameter.

7.2.4.1.1 PLCopen Function Block ErrorIDs

These are the errors returned in the ErrorID output of the function blocks.

ErrorID	Description
0	No error.
1	Queue full.
2	Abort mode required.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ErrorID	Description
3	Invalid axis.
4	One of these possible scenarios: • Invalid master axis. • Master axis and slave axis are the same. • Master axis is currently slaved to the specified slave axis. • Master axis and slave axis do not have the same update rate.
5	Invalid parameter number.
6	Invalid move.
7	Invalid override.
8	Buffer mode required.
9	Invalid parameter data.
10	Move cannot be executed because either: • An axis error exists. • The axis is in the stopping state. • The axis is disabled.
11	Invalid buffer mode.
12	Move was aborted due to an E-stop.
13	Move was aborted due to a controlled stop.
14	Invalid start mode or sync mode.
15	Invalid cam profile.
16	Invalid slave count.
17	Input value is out of range.
18	Cannot access time stamp or latched position data.
19	Data not available. Example: If a MC_ReadParam of FollowingError (1006) is programmed on a simulator axis where no following error is available, an error 19 (data not available) is returned.
20	Motion engine is not running.
21	Invalid ratio of velocity to acceleration, or acceleration to jerk. See "Limitations on Acceleration and Jerk" (➔ p. 375).
22	Too many profiles – the number of selected profiles is limited to 256.
23	Internal Error.
24	Object already exists.
25	Block needs to be called between MLMotionInit and MLMotionStart.
26	The axis is owned by another group.
27	The axis is already present in the group.
28	No axes are mapped to the group.
29	Invalid name. A name cannot be an empty string.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ErrorID	Description
30	Name is already used.
31	The number of axes is too small.
32	The number of axes is too large.
33	Invalid update rate.
34	Invalid axis group.
35	Invalid kinematic parameter (i.e., position, speed, velocity, acceleration, deceleration, or jerk).
36	Invalid floating point number.
37	Invalid object index.
38	Invalid number of positions in array.
39	Position is not a valid number.
40	Invalid coordinate system.
41	Axis is not initialized.
42	Array is not the correct size.
43	A move cannot be made while the group is in a stopping state.
44	The axes group is not in the Disabled or Standby state.
45	The axes group is in the Disabled state.
46	The axis is not in this axes group.
47	Duplicate points specified.
48	Invalid radius specified.
49	Co-linear points were specified for the border points of a circle.
50	Cannot construct a circle with specified parameters.
51	Invalid circle mode.
52	Invalid path choice for circular move.
53	Invalid transition mode.
54	The axis group does not have exactly two axes. **Example**: Transitions are only allowed on groups with two axes.
55	Invalid engine type for axis.
56	An axis or its associated drive is in an error state.
57	Cannot execute function because both queues are not empty.
58	Lines are parallel.
59	Not enough room for transitions.
60	Abort mode not allowed for Circular Absolute moves.
61	Invalid transition parameters.
62	Transition not allowed with Abort Move.
63	The axes group is not in the Standby state.
64	The maximum number of axes has been created.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ErrorID	Description
65	The axis is not powered on.
66	Error in data reported from drive.
67	The axes group is not in the ErrorStop state.
68	The axes group is still in Error State.
69	The axes group is not in Standby or ErrorStop state.
70	The fast input is in use.
71	Set position is in progress.
72	Invalid data point count.
73	Position Axis and Trigger Axis cannot be different for position mode trigger.
74	Invalid Fast Input trigger mode.
75	Cannot simultaneously perform motion from different motion engines.
76	Cannot call this function block twice in the same cycle with the same value for execute input. This error may arise from these scenarios: • The user application code was written to call the function block more than once in the same cycle. • The PLC program is overloaded causing the application to think the function block is called more than once in the same cycle.
77	An axis is in a controlled stop error state.
78	Invalid kinematic transform type specified.
79	Invalid kinematic transform parameter specified.
80	Axes group has no ACS to MCS transform specified.
81	The maximum number of transforms has been created.
82	Invalid transition between coordinate systems.
83	Invalid enabled physical axes count.
84	Operation is not valid when the axis is enabled.
86	The requested transition mode is not supported.
87	The remaining distance of the active move is shorter than the transition corner distance.
88	Aborting a move during an S-Curve Corner transition is not allowed.
100	The Cam_Data array is not large enough to hold the specified number of elements.
101	Invalid master or slave scale. The scale cannot be less than 0 (zero).
102	Element master or slave position is outside the range defined by Cam_Props.
103	A segment type was specified that is not supported by the value of the Options argument.
104	Master position of an element is too close to the master position of a previous element.
106	Invalid profile ID.
107	CamDataCount argument exceeds maximum array size of 20,000 data points.
108	Profile is currently is use.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ErrorID	Description
109	Attempting to build a profile already containing elements.
110	Unable to allocate cam table.
111	Too few cam elements specified.
112	Cam elements are not sorted.
113	Average segment velocity outside of range defined by start and end element velocities.
200	First element's MasterIn value not equal to 0 (zero).
201	Last element's MasterIn value does not equal value of InputScale.
202	Cannot modify the first element in the cam element table. SlaveOut value is outside the output range specified by Cam_Props.
203	Cannot modify the last element in the cam element table. SlaveOut value is outside the output range specified by Cam_Props.
204	Unknown cam error.

7.2.4.1.2 Axes Group Parameters

7.2.4.1.3.1 Boolean Parameters

This is a list of supported Boolean parameters.

These parameters are read and written by the MC_GrpReadBoolPar and MC_GrpWriteBoolPar function blocks.

Parameter	ID	Name	R/W	Description
MC_GRP_PARAM_IGNORE_AXIS_ESTOP	1000	Ignore Axis EStop	Read / Write	• Controls whether the axes group continues performing motion if one of the member axes is experiencing an Estop. • FALSE indicates the axes group should enter an Estop state if a member axis enters the Estop state. • TRUE indicates the axes group should continue performing motion.
MC_GRP_PARAM_AXIS_ESTOP_ACTIVE	1001	Axis EStop Active	Read-only	• This Read-only parameter is TRUE whenever an axis in the group is experiencing an axis Estop Error. • When there are no axis Estop Errors present on the axes in a group, this parameter is FALSE.

7.2.4.1.4.2 Non-Boolean Parameters

This is a list of supported non-Boolean parameters.

These parameters are read and written by the MC_GrpReadParam and MC_GrpWriteParam function blocks.

Parameter	ID	Name	R/W	Description
MC_GRP_PARAM_CIRCLE_TOLERANCE	2000	Command Velocity	Read / Write	• This parameter controls how large an error is acceptable when constructing circles using circle mode MC_CIRC_MODE_CENTER. • Zero defaults to the 1 part in 100,000 tolerance. • See "Precision Requirements for Circular Move Input Parameters" (→ p. 830).

7.2.4.1.5 Axis Parameters

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

See:

- "Boolean Parameters" (➔ p. 349)
- "Non-Boolean Parameters" (➔ p. 355)

7.2.4.1.6.1 Update Rate Types

7.2.5 Controller

The update rate depends on KAS application program update rate.
See "Define the PLC Cycle" (➔ p. 253).

7.2.6 EtherCAT Cyclic

The update rate depends on the EtherCAT.
See these topics about update rates:

- "EtherCAT Master Settings tab" (➔ p. 539)
- "Define the PLC Cycle" (➔ p. 253)

7.2.7 EtherCAT Non-Cyclic

The update rate depends on the reading of the parameter through EtherCAT.

7.2.8 Static

- If read-only, the values do not change after the axis is created.
- If read / write, the values change only when the user sets the value.

7.2.8.0.1.1 User Units to Feedback Units Ratio

Parameters "Non-Boolean Parameters" (➔ p. 355) and "Non-Boolean Parameters" (➔ p. 355) are set during the MC_CreatePLCAxis function block execution.

- These two parameters work together to form the User Units to Feedback Units Ratio (UU/FBU Ratio).
- The drive interface units are fixed by the drive and define the drive units per revolution.
  - This is used to command the drive per the ratio.

7.2.9 Example 1

Where the drive interface units are set to 1048576 units per revolution, the value of the **UU-per-revolution** ratio is calculated as:

```
UU per revolution = MC_AXIS_PARAM_UU_FB_RATIO_NUM * 1048576 / MC_AXIS_
PARAM_UU_FB_RATIO_DEN
```

7.2.10 Example 2

A Drive with a Ratio of:	Generates Revolution of Drive Motor	Generates UU-to-FBU Ration
360 UU / 1048576 FBU	360 UU	0.000343323

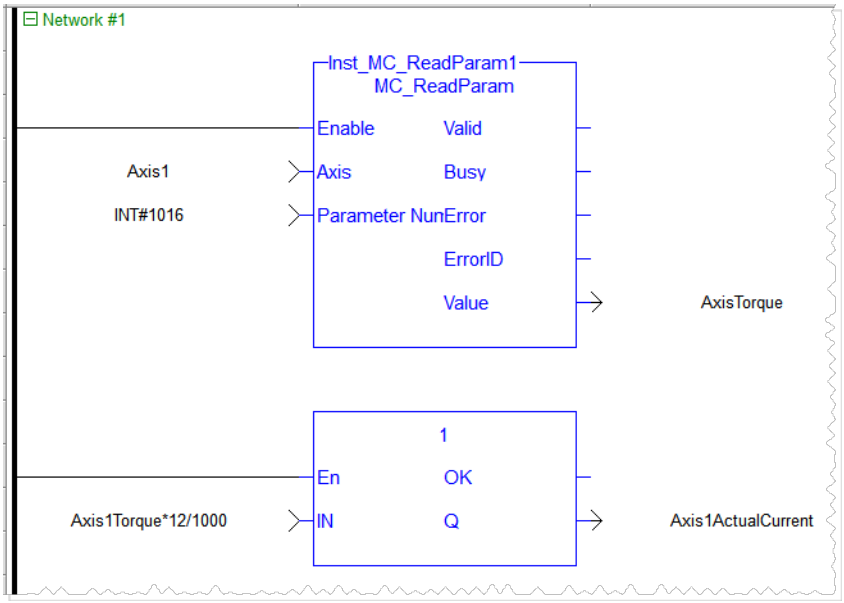
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

A Drive with a Ratio of:	Generates Revolution of Drive Motor	Generates UU-to-FBU Ration
11379 UU / 1898996404 FBU	6.283185 UU	5.99211e-6

7.2.11 Example 3 - Reading the Current to Units of Amps

After reading the current using MC_ReadParam, this equation converts current to amps for a 12 amp peak drive:

```
MC_AXIS_PARAM_TORQUE_ACTUAL * Drive Continuous Current Rating / 1000
```



7.2.11.0.1.1 Boolean Parameters

This table is a list of Boolean supported parameters.

These parameters are read and written by the function blocks MC_ReadBoolPar and MC_WriteBoolPar.

"1011" (→ p. 350)	"1013" (→ p. 350)	"1025" (→ p. 350)	"1026" (→ p. 350)
"1027" (→ p. 351)	"1033" (→ p. 352)	"1038" (→ p. 354)	"1039" (→ p. 354)

Parameter Details

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1011	MC_AXIS_PARAM_IN_POSITION	Axis In-Position	Read-only	"Controller" (→ p. 348)	• TRUE if the axis: • Has no active or next move queued. • Command delta is 0 (zero). • Actual position is within the in-position bandwidth of the command position. • FALSE otherwise, Boolean.
1013	MC_AXIS_PARAM_DRIVE_WARNING	Drive Warning	Read-only	"EtherCAT Cyclic" (→ p. 348)	Drive Warning Status. See Note 1.
1025	MC_AXIS_PARAM_REGIST_GOOD	Good Registration Mark Occurred	Read-only	"EtherCAT Non-Cyclic" (→ p. 348)	• TRUE indicates a good registration mark was encountered. • This Boolean is automatically reset after it has been read. • See Note 2.
1026	MC_AXIS_PARAM_REGIST_BAD	Bad Registration Mark Occurred	Read-only	"EtherCAT Non-Cyclic" (→ p. 348)	• TRUE indicates a bad registration mark was encountered. • This Boolean is automatically reset after it has been read. • See Note 2.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1027	MC_AXIS_PARAM_FI_OCCURRED	Fast Input Occurred	Read-only	"EtherCAT Non-Cyclic" (➔ p. 348)	• This parameter is deprecated. • The recommended replacement is the **Done** output of the MC_TouchProbe function block to verify if the fast input has occurred. **Deprecated Behavior** • TRUE if a fast input occurred on either Capture Engine 0 or Engine 1. • This Boolean is automatically reset after it has been read. • See Note 3.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description

1033	MC_AXIS_PARAM_APPLY_SUPERIMPOSED_DISTANCE	Apply Superimposed Distance	Read / Write	"Controller" (➔ p. 348)	The default value for this parameter is FALSE.
					• If FALSE, MC_MoveAbsolute does not take into account any added superimposed distance due to MC_MoveSuperimp.
					• The final reported position of the axis, which does include superimposed distance, may not be equal to the **Position** input to MC_MoveAbsolute.
					Click for an Example of FALSE
					`Param 1033 = False`
					`MC_MoveSuperimp move 1 +50`
					`MC_MoveSuperimp move 2 -35`
					`MC_MoveAbsolute 125 (superimposed distance = 50-35 = 15)`
					`End Position = 140`
					• If TRUE, MC_MoveAbsolute compensates for all added superimposed distance from MC_

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description

					MoveSuperimp calls at the time of the initial MC_ MoveAbsolute call.
					• The final reported position of the axis is equal to the **Position** input to MC_ MoveAbsolute.
					• If MC_ MoveSuperimp is called during MC_ MoveAbsolute, then the final position includes the superimposed distance.
					Click for an Example of TRUE
					`Param 1033 = True`
					`MC_ MoveSuperimp move 1 +50`
					`MC_ MoveSuperimp move 2 -35`
					`MC_ MoveAbsolute 125`
					`End Position = 125`
					Note
					• When MC_ MoveAbsolute begins execution, the distance to move is calculated at that point in time.
					• If a superimposed move is started

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
					while MC_MoveAbsolute is in progress, the final position of the axis when MC_MoveAbsolute completes is not equal to the Position input, regardless of the value of this parameter.
1038	MC_AXIS_PARAM_IGNORE_DRIVE_LIM_STATUS	Ignore Drive Limit Active	Read / Write	"EtherCAT Non-Cyclic" (→ p. 348)	• The default value is FALSE. • When TRUE, the Pipe Network motion engine ignores the DS-402 Status word bit #11 (Internal Limit Active). • See "Overtravel Conditions" (→ p. 860).
1039	MC_AXIS_PARAM_DRIVE_OVERRIDE_ON_LIMIT_ACTIVE	Drive Override on Limit Active	Read / Write	"Static" (→ p. 348)	• Set to TRUE if the drive ignores controller demand position when DS-402 Status word bit #11 (Internal Limit Active) is on. • Set to FALSE if the Pipe Network is responsible for stopping an axis when DS-402 Status word bit #11 is on. • The default value for axes connected to AKD-family drives is TRUE. • For all other axes, the default value is FALSE.

Notes

See "Tuning Controller Performance for EtherCAT Communication Latency" (→ p. 885).

1. This is a configuration parameter.
2. There is some delay in acquiring fast input information from drives and calculating the registration marks. While the information is evaluated cyclically, there may be a few cycles between when the fast input occurs and the system records the registration marks.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

3. There is some delay in acquiring fast input information from drives and calculating the fast input position. While the information is evaluated cyclically, there may be a few cycles between when the fast input occurs and the system records the fast input data.
4. There is some delay in acquiring the actual position values. While the information is evaluated cyclically, there may be a few cycles between when the actual position is updated in the drive and the system records the fast input data.

See Also

- "Axis Parameters" (→ p. 347)
- "Non-Boolean Parameters" (→ p. 355).

7.2.11.0.2.2 Non-Boolean Parameters

This table is a list of supported non-Boolean parameters.

These parameters are read and written by the function blocks MC_ReadParam and MC_WriteParam.

"1" (→ p. 355)	"10" (→ p. 355)	"11" (→ p. 356)	"1000" (→ p. 356)	"1001" (→ p. 356)	"1002" (→ p. 356)	"1003" (→ p. 356)	"1004" (→ p. 356)	"1005" (→ p. 356)
"1006" (→ p. 356)	"1007" (→ p. 356)	"1008" (→ p. 357)	"1009" (→ p. 357)	"1010" (→ p. 357)	"1012" (→ p. 357)	"1014" (→ p. 357)	"1015" (→ p. 357)	"1016" (→ p. 358)
"1017" (→ p. 358)	"1018" (→ p. 358)	"1019" (→ p. 358)	"1020" (→ p. 358)	"1021" (→ p. 358)	"1022" (→ p. 358)	"1023" (→ p. 359)	"1024" (→ p. 359)	"1028" (→ p. 359)
"1029" (→ p. 359)	"1030" (→ p. 359)	"1031" (→ p. 359)	"1032" (→ p. 359)	"1034" (→ p. 360)	"1035" (→ p. 360)	"1036" (→ p. 360)	"1037" (→ p. 360)	"1500" (→ p. 361)
"1501" (→ p. 361)	"1502" (→ p. 361)							

Parameter Details

ID	Parameter	Name	R/W	Update Rate Type	Description
1	MC_AXIS_PARAM_CMD_POS	Command Position	Read-only	"Controller" (→ p. 348)	• Axis command position - includes any command deltas from superimposed axes. • User units. • If a set position is in process due to homing or registration, there may be a 1 cycle delay.
10	MC_AXIS_PARAM_ACT_VEL	Actual Velocity	Read-only	"EtherCAT Cyclic" (→ p. 348)	• Axis actual velocity • User units / sec

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
11	MC_AXIS_PARAM_CMD_VEL	Command Velocity	Read-only	"Controller" (→ p. 348)	• Axis command velocity - includes any command deltas from superimposed axes. • User units / sec
1000	MC_AXIS_PARAM_PHASE_SHIFT	Phase Shift	Read-only	"Controller" (→ p. 348)	• The amount of phase shift applied by MC_Phasing. • Slave axis' user units
1001	MC_AXIS_PARAM_SUPERIMPOSED_DISTANCE	Superimposed Distance	Read-only	"Controller" (→ p. 348)	• The cumulative distance traveled via MC_MoveSuperimp moves. • User units
1002	MC_AXIS_PARAM_MASTER_OFFSET	Master Offset	Read / Write	"Controller" (→ p. 348)	• Write: The amount to increment the master offset for an active master/slave move. • User units. • Read: The amount of master offset applied. • User units.
1003	MC_AXIS_PARAM_SLAVE_OFFSET	Slave Offset	Read / Write	"Controller" (→ p. 348)	• Write: The amount to increment the slave offset for an active master/slave move. • User units. • Read: The amount of slave offset applied. • User units.
1004	MC_AXIS_PARAM_MOVE_TYPE_ACTIVE	Active Move Type	Read-only	"Controller" (→ p. 348)	The active move type. See Move Types.
1005	MC_AXIS_PARAM_MOVE_TYPE_NEXT	Next Move Type	Read-only	"Controller" (→ p. 348)	The queued (next) move type. See Move Types.
1006	MC_AXIS_PARAM_POSITION_ERROR	Position Error	Read-only	"EtherCAT Cyclic" (→ p. 348)	Position error in user units.
1007	MC_AXIS_PARAM_FEEDBACK_LAST	Raw Feedback	Read-only	"EtherCAT Cyclic" (→ p. 348)	• Raw Feedback position in user units. • Keeps track of the actual position based on the physical feedback device connected to the drive. • Not offset as a result of calling MC_SetPos.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1008	MC_AXIS_PARAM_ROLLOVER_POSITION	Rollover	Read / Write	"Controller" (→ p. 348)	• The axis rollover position is in User units. • If the axis is a servo axis: • This parameter can only be written when there are no moves in the queue. [1] • When this parameter is changed, the new position is scaled from the old position. In particular: `{equation}NewPosition = OldPosition * (New Rollover) / (Old Rollover) {/equation}.`
1009	MC_AXIS_PARAM_VELCOMP_FACTOR	Velocity Compensation Factor	Read / Write	"Controller" (→ p. 348)	• The factor used to multiply the velocity compensation value to account for the number of updates of delay in transmission of the feedback value from the drive to the control. • See "Velocity Compensation" (→ p. 378). • See Note 1 and Note 4.
1010	MC_AXIS_PARAM_VELCOMP_FILTER	Velocity Compensation Filter	Read / Write	"Controller" (→ p. 348)	• The number of updates to apply a change in velocity compensation. • See "Velocity Compensation" (→ p. 378). • See Note 1 and Note 4.
1012	MC_AXIS_PARAM_IN_POSITION_BAND	Axis In-Position Bandwidth	Read / Write	"Controller" (→ p. 348)	The bandwidth about the command position to determine the state of the in-position flag. See Note 2.
1014	MC_AXIS_PARAM_DRIVE_STATUS	Drive Status	Read-only	"EtherCAT Cyclic" (→ p. 348)	• Drive Status Word • Similar to MLAxisStatus. • Some status bits are set only at program startup.
1015	MC_AXIS_PARAM_UU_FB_RATIO_NUM	User Unit to feedback unit ratio numerator	Read-only	"Static" (→ p. 348)	• The numerator of the user-units-to-feedback-units ratio (UU/FBU Ratio). • See "User Units to Feedback Units Ratio" (→ p. 348). • See Note 1.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1016	MC_AXIS_PARAM_TORQUE_ACTUAL	Actual Torque	Read-only	"EtherCAT Cyclic" (→ p. 348)	• The actual torque being delivered by the drive, expressed in thousandths of max torque. • See "Example 3 - Reading the Current to Units of Amps" (→ p. 349).
1017	MC_AXIS_PARAM_BUS_ADDRESS	Drive Address	Read-only	"Static" (→ p. 348)	• Drive address value used in EtherCAT fieldbus functions as drive address. • Before using in fieldbus functions, this value needs to be converted to an integer by using a convert any to DINT function. • See Note 1.
1018	MC_AXIS_PARAM_SENSOR_DELAY	Sensor Delay	Read / Write	"Controller" (→ p. 348)	Compensation for Physical sensor delay for MC_TouchProbe, in microseconds. See Note 1.
1019	MC_AXIS_PARAM_INTERP_CMD_POS	Interpolated Command Position	Read-only	"Controller" (→ p. 348)	• Command position solely from this axis's interpolator (in User units). • This value does not include any command deltas from other axes that are currently superimposed upon it.
1020	MC_AXIS_PARAM_INTERP_CMD_VEL	Interpolated Command Velocity	Read-only	"Controller" (→ p. 348)	• Command velocity solely from this axis's interpolator (in user units). • This value does not include any command deltas from other axes that are currently superimposed upon it.
1021	MC_AXIS_PARAM_REGIST_COMP	Registration Compensation	Read-only	"Controller" (→ p. 348)	• The latest calculated registration compensation value. • This value is updated each time a good registration mark is encountered. • This value is in User units.
1022	MC_AXIS_PARAM_REGIST_DIST	Distance Between the Last Two Good Registration Marks	Read-only	"EtherCAT Non-Cyclic" (→ p. 348)	• The distance between the last two good registration marks. • This value is in User units. • See Note 2.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1023	MC_AXIS_PARAM_REGIST_GOOD_CNT	Number of Consecutive Good Registration Marks	Read / Write	"EtherCAT Non-Cyclic" (→ p. 348)	• Number of consecutive good registration marks. • This value is incremented each time a good registration mark is encountered. • It is automatically zeroed when a bad registration mark is encountered. • The ability to write this parameter is provided to allow the application to zero this value. • See Note 2.
1024	MC_AXIS_PARAM_REGIST_BAD_CNT	Number of Consecutive Bad Registration Marks	Read / Write	"EtherCAT Non-Cyclic" (→ p. 348)	• Number of consecutive bad registration marks. • This value is incremented each time a bad registration mark is encountered. • It is automatically zeroed when a good registration mark is encountered. • The ability to write this parameter is provided to allow the application to zero this value. • See Note 2.
1028	MC_AXIS_PARAM_UU_FB_RATIO_DEN	User units to feedback unit ratio denominator	Read-only	"Static" (→ p. 348)	• The denominator of the User units to feedback-units ratio (UU/FBU Ratio). • See "User Units to Feedback Units Ratio" (→ p. 348).
1029	MC_AXIS_PARAM_CM_ACT_CMD_POS	Coordinated Motion Applied Command Position	Read-only	"Controller" (→ p. 348)	Amount of motion actually applied to the PLCopen axis drive by Coordinated Motion commands.
1030	MC_AXIS_PARAM_CM_CMD_POS	Coordinated Motion Command Position	Read-only	"Controller" (→ p. 348)	Amount of motion requested of a PLCopen axis by the Coordinated Motion commands.
1031	MC_AXIS_PARAM_INGEAR_BANDWIDTH	In Gear bandwidth	Read / Write	"Controller" (→ p. 348)	• The bandwidth about the target slave velocity where the slave axis locks onto the master axis and the **InGear** output turns on for the MC_GearIn function block. • User units / sec • Default value = 0.1. • See Note 1.
1032	MC_AXIS_PARAM_DRIVE_AXIS_NUMBER	Drive Axis Number	Read-only	"Static" (→ p. 348)	One-based number that specifies the axis on the drive.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1034	MC_AXIS_PARAM_UU_FB_RATIO	User units to feedback unit ratio	Read-only	"Controller" (→ p. 348)	• The User units to feedback units ratio for the axis. • See "User Units to Feedback Units Ratio" (→ p. 348).
1035	MC_AXIS_PARAM_BUS_INTERFACE_TYPE	Bus Interface Type	Read-only	"Static" (→ p. 348)	The parameter value is: • DRIVE_BUS_INTERFACE_ECAT when the axis is associated with an EtherCAT device. • DRIVE_BUS_INTERFACE_SIMULATOR when the axis is simulated.
1036	MC_AXIS_PARAM_SIM_CLOSING	Simulator Distance Closing Fraction	Write-only	"Controller" (→ p. 348)	This setting affects how responsive a simulated axis to changing command positions. • The valid range of this parameter is (0.0, 1.5]. • The default value is 0.7. Values are broken down into these ranges: • (0.0, 1.0): Simulate following error. • A value of 0.1 generates a large following error. • A value of 0.9 generates a small following error. • 1.0: No following error. • (1.0,1.5]: Simulate leading error. **TIP** A value of 1.0 is recommended when using an axis as a virtual master to command motion through camming and/or gearing.
1037	MC_AXIS_PARAM_UU_REV_RATIO	User units revolution ratio	Read / Write	"Controller" (→ p. 348)	• The user units per revolution ratio for the axis. • See "Axis Data Parameters" (→ p. 202). **⚠WARNING** Do NOT change the user units per revolution ratio when the axis is enabled, geared, or cammed to another axis. Undefined behavior may result.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1500	MC_AXIS_PARAM_BUS_POSITION_DEMAND	Position demand value	Read-only	"EtherCAT Cyclic" (→ p. 348)	Drive Position Demand Value 32-bits. **NOTE** There may be a 1 cycle delay.
1501	MC_AXIS_PARAM_BUS_STATUS_WORD	Drive Status Word	Read-only	"EtherCAT Cyclic" (→ p. 348)	Drive Axis Status Word 16-bits. • DS-402 compatible drives: • This is the same as the CANopen Status Word. • Stepper Drives: • This is the same as the **STM Inputs** (CoE object 0x6010). • See the AKT2G-SM-Lx Object Description. • Simulated Drives: N/A **NOTE** There may be a 1 cycle delay.
1502	MC_AXIS_PARAM_BUS_CONTROL_WORD	Drive Control Word	Read-only	"EtherCAT Cyclic" (→ p. 348)	Drive Axis Control Word 16-bits. • DS-402 compatible drives: • This is the same as the CANopen Control Word. • Stepper Drives: • This is the same as the **STM Outputs** (CoE object 0x7010). • See the AKT2G-SM-Lx Object Description. • Simulated Drives: N/A **NOTE** There may be a 1 cycle delay.

Notes

See "Tuning Controller Performance for EtherCAT Communication Latency" (→ p. 885).

1. This is a configuration parameter.
2. There is some delay in acquiring fast input information from drives and calculating the registration marks. While the information is evaluated cyclically, there may be a few cycles between when the fast input occurs and the system records the registration marks.
3. There is some delay in acquiring fast input information from drives and calculating the fast input position. While the information is evaluated cyclically, there may be a few cycles between when the fast input occurs and the system records the fast input data.
4. There is some delay in acquiring the actual position values. While the information is evaluated cyclically, there may be a few cycles between when the actual position is updated in the drive and the system records the fast input data.

These parameters are read and written by the function blocks MC_ReadParam and MC_WriteParam.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

"1" (→ p. 355)	"10" (→ p. 355)	"11" (→ p. 356)	"1000" (→ p. 356)	"1001" (→ p. 356)	"1002" (→ p. 356)	"1003" (→ p. 356)	"1004" (→ p. 356)	"1005" (→ p. 356)
"1006" (→ p. 356)	"1007" (→ p. 356)	"1008" (→ p. 357)	"1009" (→ p. 357)	"1010" (→ p. 357)	"1012" (→ p. 357)	"1014" (→ p. 357)	"1015" (→ p. 357)	"1016" (→ p. 358)
"1017" (→ p. 358)	"1018" (→ p. 358)	"1019" (→ p. 358)	"1020" (→ p. 358)	"1021" (→ p. 358)	"1022" (→ p. 358)	"1023" (→ p. 359)	"1024" (→ p. 359)	"1028" (→ p. 359)
"1029" (→ p. 359)	"1030" (→ p. 359)	"1031" (→ p. 359)	"1032" (→ p. 359)	"1034" (→ p. 360)	"1035" (→ p. 360)	"1036" (→ p. 360)	"1037" (→ p. 360)	"1500" (→ p. 361)
"1501" (→ p. 361)	"1502" (→ p. 361)							

Parameter Details

ID	Parameter	Name	R/W	Update Rate Type	Description
1	MC_AXIS_PARAM_CMD_POS	Command Position	Read-only	"Controller" (→ p. 348)	• Axis command position - includes any command deltas from superimposed axes. • User units. • If a set position is in process due to homing or registration, there may be a 1 cycle delay.
10	MC_AXIS_PARAM_ACT_VEL	Actual Velocity	Read-only	"EtherCAT Cyclic" (→ p. 348)	• Axis actual velocity • User units / sec
11	MC_AXIS_PARAM_CMD_VEL	Command Velocity	Read-only	"Controller" (→ p. 348)	• Axis command velocity - includes any command deltas from superimposed axes. • User units / sec
1000	MC_AXIS_PARAM_PHASE_SHIFT	Phase Shift	Read-only	"Controller" (→ p. 348)	• The amount of phase shift applied by MC_Phasing. • Slave axis' user units
1001	MC_AXIS_PARAM_SUPERIMPOSED_DISTANCE	Superimposed Distance	Read-only	"Controller" (→ p. 348)	• The cumulative distance traveled via MC_MoveSuperimp moves. • User units

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1002	MC_AXIS_PARAM_MASTER_OFFSET	Master Offset	Read / Write	"Controller" (→ p. 348)	• Write: The amount to increment the master offset for an active master/slave move. • User units. • Read: The amount of master offset applied. • User units.
1003	MC_AXIS_PARAM_SLAVE_OFFSET	Slave Offset	Read / Write	"Controller" (→ p. 348)	• Write: The amount to increment the slave offset for an active master/slave move. • User units. • Read: The amount of slave offset applied. • User units.
1004	MC_AXIS_PARAM_MOVE_TYPE_ACTIVE	Active Move Type	Read-only	"Controller" (→ p. 348)	The active move type. See Move Types.
1005	MC_AXIS_PARAM_MOVE_TYPE_NEXT	Next Move Type	Read-only	"Controller" (→ p. 348)	The queued (next) move type. See Move Types.
1006	MC_AXIS_PARAM_POSITION_ERROR	Position Error	Read-only	"EtherCAT Cyclic" (→ p. 348)	Position error in user units.
1007	MC_AXIS_PARAM_FEEDBACK_LAST	Raw Feedback	Read-only	"EtherCAT Cyclic" (→ p. 348)	• Raw Feedback position in user units. • Keeps track of the actual position based on the physical feedback device connected to the drive. • Not offset as a result of calling MC_SetPos.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1008	MC_AXIS_PARAM_ROLLOVER_POSITION	Rollover	Read / Write	"Controller" (→ p. 348)	• The axis rollover position is in User units. • If the axis is a servo axis: • This parameter can only be written when there are no moves in the queue. [1] • When this parameter is changed, the new position is scaled from the old position. In particular: `{equation}NewPosition = OldPosition * (New Rollover) / (Old Rollover) {/equation}.`
1009	MC_AXIS_PARAM_VELCOMP_FACTOR	Velocity Compensation Factor	Read / Write	"Controller" (→ p. 348)	• The factor used to multiply the velocity compensation value to account for the number of updates of delay in transmission of the feedback value from the drive to the control. • See "Velocity Compensation" (→ p. 378). • See Note 1 and Note 4.
1010	MC_AXIS_PARAM_VELCOMP_FILTER	Velocity Compensation Filter	Read / Write	"Controller" (→ p. 348)	• The number of updates to apply a change in velocity compensation. • See "Velocity Compensation" (→ p. 378). • See Note 1 and Note 4.
1012	MC_AXIS_PARAM_IN_POSITION_BAND	Axis In-Position Bandwidth	Read / Write	"Controller" (→ p. 348)	The bandwidth about the command position to determine the state of the in-position flag. See Note 2.
1014	MC_AXIS_PARAM_DRIVE_STATUS	Drive Status	Read-only	"EtherCAT Cyclic" (→ p. 348)	• Drive Status Word • Similar to MLAxisStatus. • Some status bits are set only at program startup.
1015	MC_AXIS_PARAM_UU_FB_RATIO_NUM	User Unit to feedback unit ratio numerator	Read-only	"Static" (→ p. 348)	• The numerator of the user-units-to-feedback-units ratio (UU/FBU Ratio). • See "User Units to Feedback Units Ratio" (→ p. 348). • See Note 1.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1016	MC_AXIS_PARAM_TORQUE_ACTUAL	Actual Torque	Read-only	"EtherCAT Cyclic" (→ p. 348)	• The actual torque being delivered by the drive, expressed in thousandths of max torque. • See "Example 3 - Reading the Current to Units of Amps" (→ p. 349).
1017	MC_AXIS_PARAM_BUS_ADDRESS	Drive Address	Read-only	"Static" (→ p. 348)	• Drive address value used in EtherCAT fieldbus functions as drive address. • Before using in fieldbus functions, this value needs to be converted to an integer by using a convert any to DINT function. • See Note 1.
1018	MC_AXIS_PARAM_SENSOR_DELAY	Sensor Delay	Read / Write	"Controller" (→ p. 348)	Compensation for Physical sensor delay for MC_TouchProbe, in microseconds. See Note 1.
1019	MC_AXIS_PARAM_INTERP_CMD_POS	Interpolated Command Position	Read-only	"Controller" (→ p. 348)	• Command position solely from this axis's interpolator (in User units). • This value does not include any command deltas from other axes that are currently superimposed upon it.
1020	MC_AXIS_PARAM_INTERP_CMD_VEL	Interpolated Command Velocity	Read-only	"Controller" (→ p. 348)	• Command velocity solely from this axis's interpolator (in user units). • This value does not include any command deltas from other axes that are currently superimposed upon it.
1021	MC_AXIS_PARAM_REGIST_COMP	Registration Compensation	Read-only	"Controller" (→ p. 348)	• The latest calculated registration compensation value. • This value is updated each time a good registration mark is encountered. • This value is in User units.
1022	MC_AXIS_PARAM_REGIST_DIST	Distance Between the Last Two Good Registration Marks	Read-only	"EtherCAT Non-Cyclic" (→ p. 348)	• The distance between the last two good registration marks. • This value is in User units. • See Note 2.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1023	MC_AXIS_PARAM_REGIST_GOOD_CNT	Number of Consecutive Good Registration Marks	Read / Write	"EtherCAT Non-Cyclic" (→ p. 348)	• Number of consecutive good registration marks. • This value is incremented each time a good registration mark is encountered. • It is automatically zeroed when a bad registration mark is encountered. • The ability to write this parameter is provided to allow the application to zero this value. • See Note 2.
1024	MC_AXIS_PARAM_REGIST_BAD_CNT	Number of Consecutive Bad Registration Marks	Read / Write	"EtherCAT Non-Cyclic" (→ p. 348)	• Number of consecutive bad registration marks. • This value is incremented each time a bad registration mark is encountered. • It is automatically zeroed when a good registration mark is encountered. • The ability to write this parameter is provided to allow the application to zero this value. • See Note 2.
1028	MC_AXIS_PARAM_UU_FB_RATIO_DEN	User units to feedback unit ratio denominator	Read-only	"Static" (→ p. 348)	• The denominator of the User units to feedback-units ratio (UU/FBU Ratio). • See "User Units to Feedback Units Ratio" (→ p. 348).
1029	MC_AXIS_PARAM_CM_ACT_CMD_POS	Coordinated Motion Applied Command Position	Read-only	"Controller" (→ p. 348)	Amount of motion actually applied to the PLCopen axis drive by Coordinated Motion commands.
1030	MC_AXIS_PARAM_CM_CMD_POS	Coordinated Motion Command Position	Read-only	"Controller" (→ p. 348)	Amount of motion requested of a PLCopen axis by the Coordinated Motion commands.
1031	MC_AXIS_PARAM_INGEAR_BANDWIDTH	In Gear bandwidth	Read / Write	"Controller" (→ p. 348)	• The bandwidth about the target slave velocity where the slave axis locks onto the master axis and the **InGear** output turns on for the MC_GearIn function block. • User units / sec • Default value = 0.1. • See Note 1.
1032	MC_AXIS_PARAM_DRIVE_AXIS_NUMBER	Drive Axis Number	Read-only	"Static" (→ p. 348)	One-based number that specifies the axis on the drive.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1034	MC_AXIS_PARAM_UU_FB_RATIO	User units to feedback unit ratio	Read-only	"Controller" (→ p. 348)	• The User units to feedback units ratio for the axis. • See "User Units to Feedback Units Ratio" (→ p. 348).
1035	MC_AXIS_PARAM_BUS_INTERFACE_TYPE	Bus Interface Type	Read-only	"Static" (→ p. 348)	The parameter value is: • DRIVE_BUS_INTERFACE_ECAT when the axis is associated with an EtherCAT device. • DRIVE_BUS_INTERFACE_SIMULATOR when the axis is simulated.
1036	MC_AXIS_PARAM_SIM_CLOSING	Simulator Distance Closing Fraction	Write-only	"Controller" (→ p. 348)	This setting affects how responsive a simulated axis to changing command positions. • The valid range of this parameter is (0.0, 1.5]. • The default value is 0.7. Values are broken down into these ranges: • (0.0, 1.0): Simulate following error. • A value of 0.1 generates a large following error. • A value of 0.9 generates a small following error. • 1.0: No following error. • (1.0,1.5]: Simulate leading error. **TIP** A value of 1.0 is recommended when using an axis as a virtual master to command motion through camming and/or gearing.
1037	MC_AXIS_PARAM_UU_REV_RATIO	User units revolution ratio	Read / Write	"Controller" (→ p. 348)	• The user units per revolution ratio for the axis. • See "Axis Data Parameters" (→ p. 202). **⚠WARNING** Do NOT change the user units per revolution ratio when the axis is enabled, geared, or cammed to another axis. Undefined behavior may result.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ID	Parameter	Name	R/W	Update Rate Type	Description
1500	MC_AXIS_PARAM_BUS_POSITION_DEMAND	Position demand value	Read-only	"EtherCAT Cyclic" (→ p. 348)	Drive Position Demand Value 32-bits. **NOTE** There may be a 1 cycle delay.
1501	MC_AXIS_PARAM_BUS_STATUS_WORD	Drive Status Word	Read-only	"EtherCAT Cyclic" (→ p. 348)	Drive Axis Status Word 16-bits. • DS-402 compatible drives: • This is the same as the CANopen Status Word. • Stepper Drives: • This is the same as the **STM Inputs** (CoE object 0x6010). • See the AKT2G-SM-Lx Object Description. • Simulated Drives: N/A **NOTE** There may be a 1 cycle delay.
1502	MC_AXIS_PARAM_BUS_CONTROL_WORD	Drive Control Word	Read-only	"EtherCAT Cyclic" (→ p. 348)	Drive Axis Control Word 16-bits. • DS-402 compatible drives: • This is the same as the CANopen Control Word. • Stepper Drives: • This is the same as the **STM Outputs** (CoE object 0x7010). • See the AKT2G-SM-Lx Object Description. • Simulated Drives: N/A **NOTE** There may be a 1 cycle delay.

Notes

See "Tuning Controller Performance for EtherCAT Communication Latency" (→ p. 885).

1. This is a configuration parameter.
2. There is some delay in acquiring fast input information from drives and calculating the registration marks. While the information is evaluated cyclically, there may be a few cycles between when the fast input occurs and the system records the registration marks.
3. There is some delay in acquiring fast input information from drives and calculating the fast input position. While the information is evaluated cyclically, there may be a few cycles between when the fast input occurs and the system records the fast input data.
4. There is some delay in acquiring the actual position values. While the information is evaluated cyclically, there may be a few cycles between when the actual position is updated in the drive and the system records the fast input data.

See Also

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Axis Parameters" (➜ p. 347)
- "Boolean Parameters" (➜ p. 349).

7.2.11.0.3 Axis Positions Data

7.2.11.0.4.1 Actual Position

This is the position of the axis read from the drive interface, which is read from the feedback device.

7.2.11.0.5.2 CommandPosition

This is the command position (without modulo and in the axis' user units) sent to the drive interface to command an axis.

This position is tied to the Status output of the MC_Power function block:

- Status = 1: The command position is a combination of the Normal, Superimposed, and Phasing commands.
- Status = 0: The command position tracks the Actual Position.

7.2.12 CoordinatedMotionCmdPos

This is the command position generated by the Coordinated Motion Interpolator when interpolating coordinated motion moves.

7.2.13 Group Actual Position

This is the set of positions of an axes group read from the axis interface:

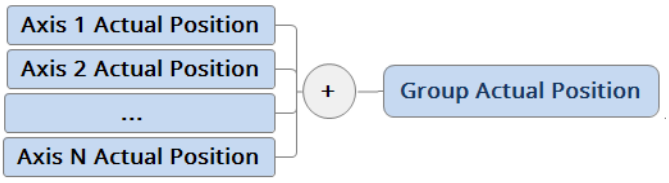
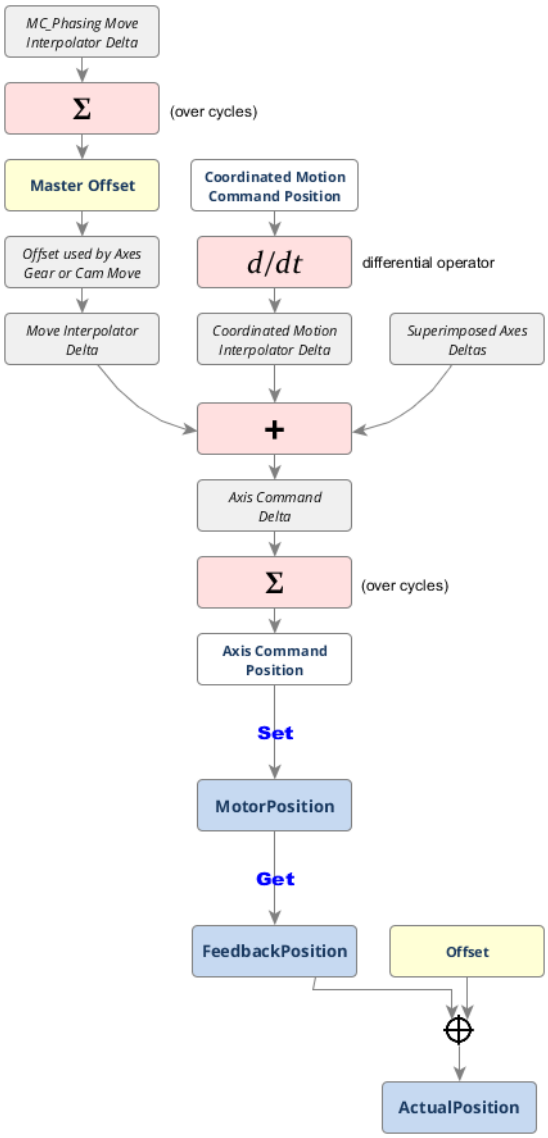


Figure 7-35: How **Actual Position** is Calculated



7.2.14 NormalCmdPos

This is the command position generated by the Normal Interpolator when interpolating a single axis move or a slave move.

7.2.15 PhaseCmd Pos

This is the command position generated by the Phasing Interpolator when interpolating a MC_Phasing master phase shift.

It is incorporated in the generation of the NormalCmdPos when interpolating a slave move.

7.2.16 SuperimposedCmdPos

This is the command position generated by the Superimposed Interpolator when interpolating a MC_Superimp move

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

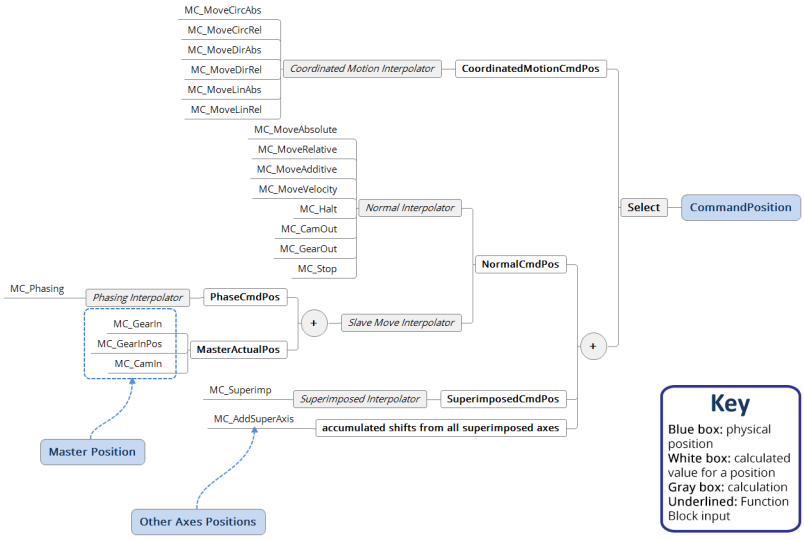


Figure 7-36: Calculated PLCopen Axis Position Data

7.2.16.0.1 AXIS_REF Structure

The PLCopen specification indicates a "Structure" (➔ p. 288) used for identifying the axis at a function block input.

AXIS_REF contains these members:

Member	Type	Description
AXIS_NUM	INT	The axis number.
AXIS_ENGINE	UINT	Reserved. Do not modify this variable or rely on its value.

See "Change a PLCopen Axis" (➔ p. 196) about Axis Number.

You have to create and initialize this data structure in your application.

7.2.16.0.2 Axis Velocity Data

Velocity data are related to PLCopen Axis.

7.2.16.0.3.1 Actual Velocity

This is the rate of change of the actual position of the axis read from the drive interface which is read from the feedback device:

Feedback device > Drive interface > ActualPosition(n) – ActualPosition(n-1)

7.2.16.0.4.2 Command Velocity

This is the rate of change in command position.

- This velocity is tied to the Status output of the MC_Power function block.
  - Since command velocity is the rate change of command position:
    - Status = 1: The rate of change of the command position sent to the drive interface to command the axis.
    - Status = 0: The command velocity tracks the Actual Velocity.

7.2.16.0.5 Buffer Modes

Some of the function blocks (FBs) have an input called BufferMode.

- With this input, the FB can either work in a Non-buffered mode (default behavior) or in a buffered mode.
  - The difference between those modes is when they start their action:
    - A non-buffered mode command acts immediately, even if this interrupts another motion.
    - A buffered mode command waits until the current FB sets its **Done** output (or **InPosition** or **InVelocity**).

This table lists the buffer modes that can be specified at the BufferMode input of the function blocks.

Value	Buffer Mode	Short Name	Description
0	BM_ ABORTING	Abort	A move that specifies Abort: • Aborts the active move. • Removes the next move from the queue. • Immediately becomes the active move and begins executing.
1	BM_ BUFFERED	Buffer	One of these events can happen with a move that specifies Buffer: • **Case 1**: If there is no active move, this move immediately becomes the active move and begins executing. • **Case 2**: If there is an active move but no next move queued, this move is queued as the next move. • It begins executing when the active move has completed and decelerated to zero velocity. • **Case 3**: If there is an active move and a queued next move, this move does not execute but returns the error **queue full** at the ErrorID output.
2	BM_ BLENDING_ PREVIOUS	Blend to Previous	A move specifying Blend-to-Active behaves the same as Buffer in **Cases 1 and 3**. • In **Case 2**, this move is queued as the next move. • The active move stays at its programmed velocity to its endpoint. • When the active move reaches its endpoint, this move becomes active and begins to accelerate or decelerate to its programmed velocity.
3	BM_ BLENDING_ NEXT	Blend to Next	A move specifying Blend-to-Next behaves the same as Buffer in **Cases 1 and 3**. • In **Case 2**, this move is queued as the next move. • When the expected time is reached, the active move begins to accelerate or decelerate so it reaches this move's programmed velocity at the time the active move reaches its endpoint.
4	BM_ BLENDING_ LOW	Blend to Low	A move specifying Blend-to-Low behaves like Blend-to-Active if the active move's velocity is lower than this move's velocity. • It behaves like Blend-to-Next if this move's velocity is lower than the active move's velocity.
5	BM_ BLENDING_ HIGH	Blend to High	A move specifying Blend-to-High behaves like Blend-to-Active if the active move's velocity is higher than this move's velocity. • It behaves like Blend-to-Next if this move's velocity is higher than the active move's velocity.

IMPORTANT

The **MC_BUFFER_MODE_BLENDING_NEXT** and **MC_BUFFER_MODE_BLENDING_HIGH** buffer modes may increase the acceleration to meet the target position at the specified velocity.
The acceleration is automatically re-calculated to reach the target position at the specified velocity if the

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

specified acceleration is too small to reach the target velocity within the specified distance.

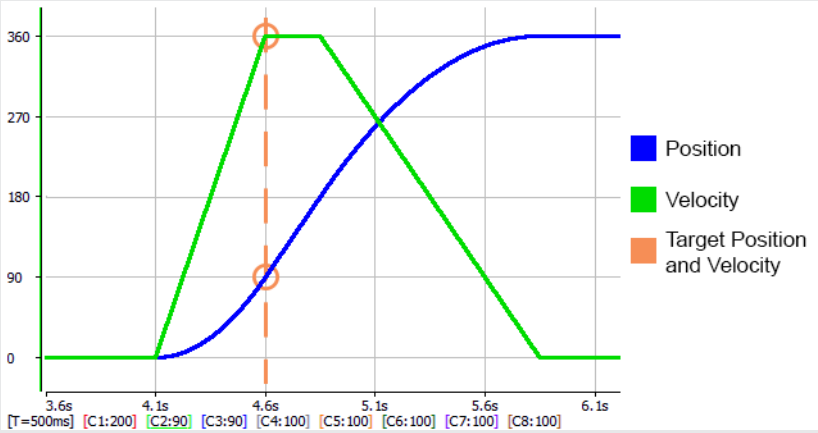
Example: These absolute position moves are commanded:
Move #1:

Position = 90, Velocity = 180, Accel = 360

Move #2: (with Blending High or Blending Next)

Position = 360, Velocity = 360, Accel = 360

To reach velocity = 360 within a distance of 90, the acceleration must be 720.
The function block automatically calculates and applies an acceleration of 720 to reach the target position at the specified velocity:



7.2.16.0.6 Function Block Acceleration and Jerk Parameters

Acceleration is the first derivative of velocity, or the rate of change of velocity.

- Deceleration is a term to describe acceleration when the axis decreases its speed.
- Function block inputs acceleration and deceleration specify how quickly an axis may change its velocity.
  - The acceleration input is used when an axis increases its speed.
  - The deceleration input is used when an axis decreases its speed.
- Jerk is the second derivative of velocity, or the rate change of acceleration.
  - The jerk rate therefore specifies how quickly an axis may change its acceleration.
  - Jerk controls how abrupt the axis begins and ends the acceleration and deceleration curves.

See "Motion Profile" (→ p. 322).

IMPORTANT

In these sections, acceleration can refer to either acceleration or deceleration when discussing function block inputs.

- "Rules" (→ p. 373)
- "Methods" (→ p. 374)
  - "Example 1" (→ p. 374)
  - "Example 2" (→ p. 374)
  - "Example 3" (→ p. 374)
- "Limitations on Acceleration and Jerk" (→ p. 375)

7.2.16.0.7.1 Rules

The amount of time an axis takes to change its velocity from one velocity to another is a function of both acceleration and jerk.

The larger the values of acceleration and jerk, the more quickly an axis attains its programmed velocity.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.17 Generalizations

These are generalizations that can be made about acceleration, jerk, and their relationships to each other:

- The higher the acceleration rate, the faster the axis obtains programmed velocity.
- The higher the jerk rate, the more responsive the axis is to changes in command.
- Excessive jerk typically, and more noticeably, contributes to harsh acceleration rather than excessive acceleration.
- Too low of a jerk value contributes to slow axis responsiveness to changing commands.
  - Lower jerks tend to soften the beginning and end of acceleration.
  - Higher jerks sharpen the beginning and end of acceleration.
- Typically:
  - Jerk > Acceleration
  - Acceleration > Velocity

7.2.17.0.1.1 Methods

There are several methodologies to determine proper acceleration and jerk values.

- These methodologies allow calculation of parameters given different desired profiles.
  - Once parameters are calculated, they can be modified to obtain the designated results.
- Acceleration and Jerk values are subject to the limits of ratios.

7.2.18 Example 1

1/3,1/3,1/3 time, given velocity and time.

```
Acceleration = (3 * Velocity)/(2 * time)
Jerk = 3* Acceleration / time
```

- This allows you to calculate an appropriate acceleration and jerk for an axis acceleration / deceleration profile to:
  - Jerk or ramp acceleration up for 1/3 of the time.
  - Accelerate 1/3 of the time.
  - Ramp acceleration down 1/3 of the time.
- Time is the desired amount of time to reach desired velocity.
  - This is the time to change velocity, not the time to complete the move.

7.2.19 Example 2

1/3,1/3,1/3 velocity, given velocity and time.

```
Acceleration = (5 * Velocity) / (3 * time)
Jerk = (3 * Acceleration ^2) / (2 * velocity)
```

- This allows you to calculate an appropriate acceleration and jerk, if you would like an axis acceleration / deceleration profile to:
  - Jerk or ramp acceleration up for 1/3 of the velocity change.
  - Accelerate 1/3 of the velocity change.
  - Ramp acceleration down 1/3 of the velocity change.
- Where velocity is the desired velocity change, and time is the desired amount of time to reach the desired velocity change.
  - This is the time to change velocity, not the time to complete the move.

7.2.20 Example 3

Calculate jerk, given velocity, acceleration, and time.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
Jerk = (2 * Acceleration) / ( time – ( velocity / (2 * acceleration)))
```

If you know the maximum acceleration of the axis and want to calculate a jerk given the velocity and time, use this equation.

This is the time to change velocity, not the time to complete the move.

7.2.20.0.1.1 Limitations on Acceleration and Jerk

The ratios of acceleration to jerk and velocity to jerk are limited on most function blocks.

- The ratio of velocity to acceleration must be less than **20**.
  - A value of 20 suggests a time to accelerate to velocity of approximately 20 seconds, assuming infinite jerk.
  - As jerk is decreased, this acceleration time is increased.
- The ratio of acceleration to jerk must be less than **2**.
  - A value of 2 suggests the time to jerk to the acceleration rate is approximately 2 seconds.

7.2.20.0.2 Move Types

MoveID	Description	Related FB
0	No move	
1	Distance move	• MC_MoveRelative • MC_MoveSuperimp • MC_Phasing
2	Position move	MC_MoveAbsolute
3	Velocity move	MC_MoveVelocity
4	Halt move	MC_Halt
5	Gear-in move	MC_GearIn
6	Gear-out move	MC_GearOut
7	Reference move	MC_Reference
8	Stop move	MC_Stop
9	Gear-in pos. move	MC_GearInPos
10	Cam profile move	MC_CamIn
11	Cam-out move	MC_CamOut

7.2.20.0.3 Profile Generator

Each servo axis has these Profile Generators which has its own queue:

- Normal - handles all single-axis and master-slave moves.
- Superimposed - handles MC_MoveSuperimp moves exclusively.
- Phasing - handles MC_Phasing phase shifts exclusively.

The Profile Generators allow these types of moves to execute simultaneously.

7.2.20.0.4 Queuing

A queuing mechanism is provided for all PLCopen motion function blocks including single-axis and master/slave moves.

- This mechanism allows the application to queue a next move while the active move is executing.
- The Buffer Modes define the transition from the active move to the next move.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.20.0.5 S-curve and Trapezoidal Acceleration / Deceleration

7.2.20.0.6.1 S-curve

If the Jerk input of a motion function block is non-zero, S-curve acceleration / deceleration is used.

- The Acceleration input specifies the maximum acceleration reached when the velocity increases in magnitude.
- The Deceleration input specifies the maximum deceleration reached when the velocity decreases in magnitude.
- The Jerk input specifies the constant rate of change of acceleration and deceleration.

7.2.20.0.7.2 Small Jerk Acceleration

Figure 7-37 is a velocity plot of the acceleration of a move when Jerk is a small value.

The smaller the Jerk value, the more gradual the rate of change of acceleration and deceleration when transitioning from one velocity to another.

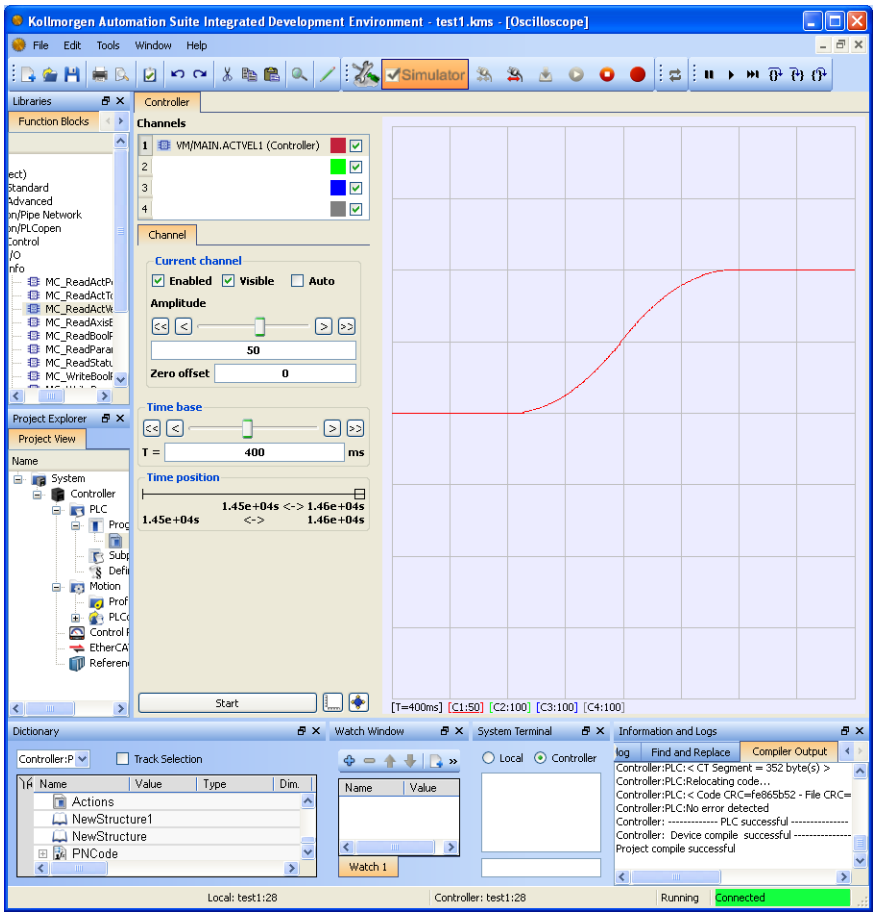


Figure 7-37: Small Jerk Acceleration

7.2.20.0.8.3 Large Jerk Acceleration

Figure 7-38 is a velocity plot of the acceleration of a move when Jerk is a large value.

The larger the Jerk value, the more abrupt the rate of change of acceleration and deceleration when transitioning from one velocity to another.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

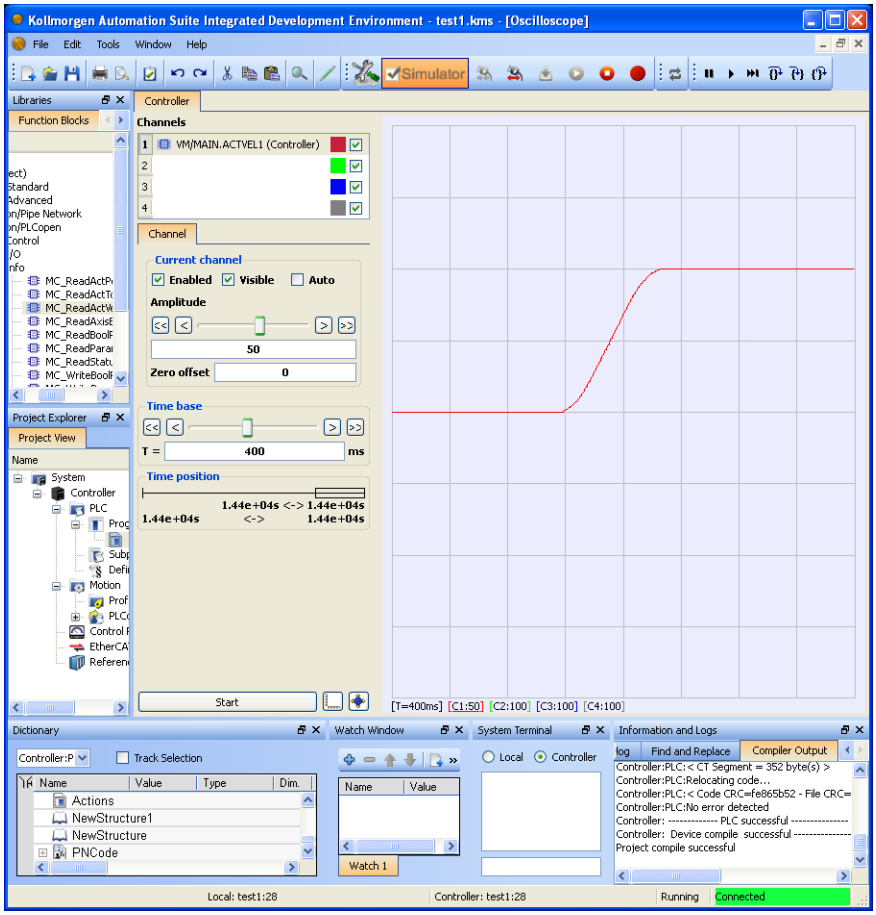


Figure 7-38: Large Jerk Acceleration

7.2.20.0.9.4 Trapezoidal

If the Jerk input of a motion function block is zero, trapezoidal acceleration and deceleration is used.

- The Acceleration input specifies the linear acceleration rate.
- The Deceleration input specifies the linear deceleration rate.

Figure 7-39 is a velocity plot of the acceleration of a move when trapezoidal acceleration is used (Jerk = 0 (zero)).

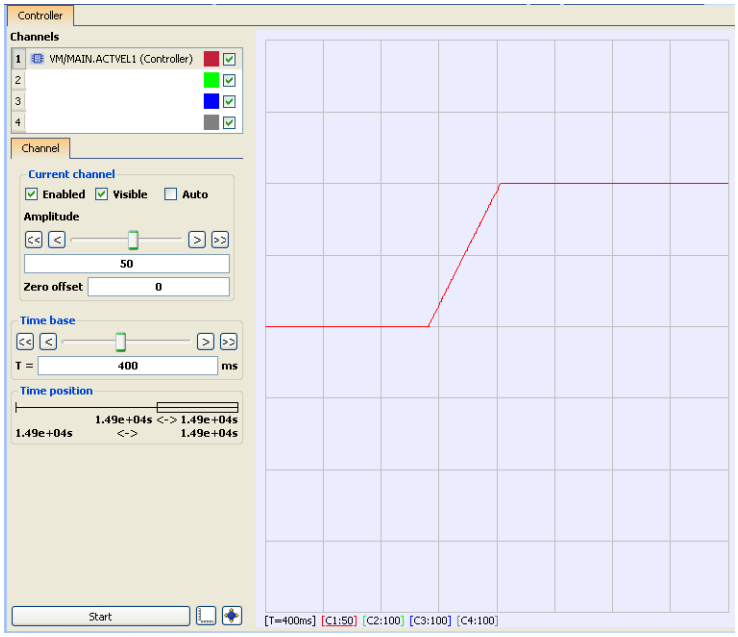


Figure 7-39: Trapezoidal Acceleration

7.2.20.0.10 Velocity Compensation

Velocity compensation is a value that is added to the master actual position to predict where the master will be when the new slave command position is sent to the slave.

- The compensation is used to get rid of the delay in reading the master's feedback.
- This involves prediction about where the master will be, so there is likely some small error in the prediction.

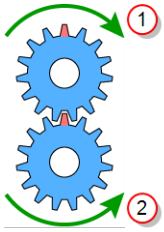


Figure 7-40: Ideal 1:-1 Gearing

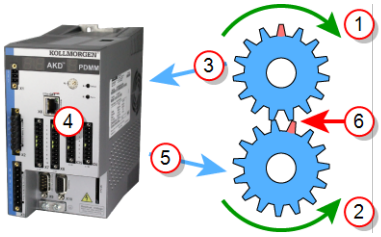


Figure 7-41: Real-world 1:-1 Gearing

1. Master
2. Slave
3. Feedback
4. 1+ cycle delay
5. Slave command position
6. Out of sync

Figure 1 and Figure 2 represent a 1:-1 gearing.

- Figure 7-40 is ideal gearing where the master and slave are always perfectly synchronized.
  - When the red tooth of the master gear is at the top, the red tooth of the slave gear is also at the top.
- Figure 7-41 is a real-world gearing example.
  - There is a time delay in reading the master feedback position, communicating that value to the controller, and the controller communicating the new command position to the slave axis.
  - This delay causes the slave to lag behind in its position.
  - This is demonstrated by the slave's read tooth being slightly out of position.

To compensate for the delay, we want to predict where the master axis will be when the new slave position applied to the slave. This is the purpose of the velocity compensation.

- Velocity compensation is controlled by two axis parameters:
  - "Velocity Compensation Factor" (➔ p. 379)
  - "Velocity Compensation Filter" (➔ p. 379)
- Both parameters are properties of master axes.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- When an axis is synchronized to a master via camming or gearing, the PLCopen motion engine uses these two properties of the master axis to predict where the master axis will be in the next motion cycle.
- The PLCopen motion engine then uses this predicted master position to calculate the new slave position.

7.2.20.0.11.1 Velocity Compensation Factor

The **Velocity Compensation Factor** (Axis Parameter 1009) is the number of motion cycles to compensate for the feedback latency.

- It should be set equal to the number of motion cycles between when the feedback device acquires the position and when the motion controller processes the position feedback.
- A value of 0 indicates no compensation is applied.
- The larger the value of Velocity Compensation Factor, the more prediction takes place and the more prediction error there will be.
- For physical axes, the value for Velocity Compensation Factor is normally 1 or 2.
  - Some feedback devices or drives may require larger values.
- For virtual masters, the value is usually be 0.
  - Virtual masters do not need compensation since their position is always known by the controller.

7.2.20.0.12.2 Velocity Compensation Filter

The **Velocity Compensation Filter** (Axis Parameter 1010) is a way to filter noisy feedback systems and/or masters that may have jitter due to servo action.

- Numerically, the Velocity Compensation Filter is the number of motion cycles over which a previously calculated compensation is applied.
  - This helps smooth noisy systems.
- The default value for Velocity Compensation Filter is 1.
  - No filtering takes place.
  - All corrections are applied in a single cycle.
- Larger values smooth the noise, but can also cause the prediction of the master position to be further off by what is characterized as additional following error or phase delay.
- Balancing between the level of noise and the amount of phase delay is application specific and is best determined by experimentation.
- Good values typically lie in the range of 1 to 5.

See Also

"Axis Parameters" (➔ p. 347)

7.2.20.1 Function Blocks - General Rules

The general rules for PLCopen are:

- "Active Output" (➔ p. 379)
- "Behavior of Busy Output" (➔ p. 380)
- "Behavior of CommandAborted Output" (➔ p. 380)
- "Behavior of Done Output" (➔ p. 380)
- "Error Handling Behavior" (➔ p. 380)
- "Input Parameters" (➔ p. 381)
- "Inputs Exceed Application Limits" (➔ p. 383)
- "Missing Input Parameters" (➔ p. 383)
- "Output Exclusivity" (➔ p. 383)
- "Output Status" (➔ p. 383)
- "Sign Rules" (➔ p. 383)

7.2.20.1.1 Active Output

The **Active** output is set at the moment the function block takes control of the motion of the respective axis.

7.2.20.1.2.1 Coordinated Motion

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Use this table to determine the state of the Busy and Active outputs when transitioning from one move to the next.

FB Output	First Move	Second Move
Busy	False	True
Active	False	False
Done	True*	False

*This output is False if the move is interrupted by a MC_GrpHalt, MC_GrpStop or if the second move's buffer mode is **Aborting**.

7.2.20.1.3 Behavior of Busy Output

The **Busy** output indicates that the function block is still working, with new output values to be expected.

- **Busy** is SET at the rising edge of **Execute** and RESET when one of the outputs (e.g., **Done**, **Aborted**, or **Error**) is set.
- It is recommended that this function block is kept in the active loop of the application program for at least as long as **Busy** is TRUE, because the outputs can still change.
- For one axis, several function blocks can be busy, but only one can be active at a time.
    - Exceptions are **MC_SuperImposed** and **MC_Phasing**, where more than one function block related to one axis can be active.

7.2.20.1.4 Behavior of CommandAborted Output

- **CommandAborted** is set when a commanded motion is interrupted by another motion command.
- The reset-behavior of **CommandAborted** is similar to **Done**.
    - When **CommandAborted** occurs, the other output signals (e.g., **InVelocity**) are reset.

7.2.20.1.5 Behavior of Done Output

The **Done** output (as well as **InGear**, **InSync**) is set when the commanded action has been completed successfully.

- With multiple function blocks working on the same axis in a sequence, this applies:
    - When one movement on an axis is interrupted with another movement on the same axis without having reached the final goal, **Done** of the first function block is not set.

When a motion command is executed, these are the possible outcomes:

- It completes successfully.
    - At that time, the **Done** output goes high.
- It is aborted prior to completing by a subsequent motion command.
    - At that time, the **CommandAborted** output goes high.
- It encounters an error prior to completing or an invalid input is specified.
    - At that time, the **Error** output goes high.

These outputs stay in this state until that motion function block is executed again.

At that time, the **Done**, **CommandAborted** and **Error** outputs goes low and the **Busy** output goes high, provided all the inputs are valid.

7.2.20.1.6 Error Handling Behavior

Two outputs deal with errors that can occur while executing a function block.

These outputs are defined as:

- **Error**: The rising edge of **Error** informs you that an error occurred during the execution of the function block.
- **ErrorID**: Error number.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- **Done**, **InVelocity**, **InGear**, and **InSync** mean successful completion.
  - These signals are logically exclusive to **Error**.
- Instance errors do not always result in an axis error.
  - Some bring the axis to **StandStill**.

7.2.20.1.7 Input Parameters

Unless otherwise specified in the function block's description, the input parameters are read with the rising edge of the **Execute** input.

The input parameters are:

The PLCopen function / function block input parameters are:

- "PLCopen Function Blocks with Execute" (→ p. 381)
- "PLCopen Function Blocks with Enable" (→ p. 382)
- "PLCopen Functions with Input Parameter En" (→ p. 382)

7.2.20.1.8.1 PLCopen Function Blocks with Execute

- These FBs initiate execution on the rising edge of **Execute**.
- Execution continues until completion independent of the **Execute** value (or values) following initiation.
- Once execution is complete, the function block can be rearmed or reengaged by introducing another rising edge of **Execute**.

NOTE

Dual-Core limitation: Consecutive calls to the function block with repeating **Execute** values in the same cycle return an error code "76" (→ p. 346).

Function Block	Description
MC_AbortTrigger MC_SetPosition MC_SyncSlaves MC_TouchProbe	A positive transition of this input causes this function block to execute.
MC_CamIn MC_CamOut MC_GearIn MC_GearInPos MC_GearOut MC_Halt MC_MoveAbsolute MC_MoveAdditive MC_MoveRelative MC_MoveSuperimp MC_MoveVelocity	A positive transition of this input requests to queue the move.
MC_CamTblSelect	A positive transition of this input reads and initializes the specified profile.
MC_Phasing	A positive transition of this input requests to queue the phase shift move.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Function Block	Description
MC_Reference	A positive transition of this input requests to queue the reference move and arm the reference trigger events.
MC_Stop	A positive transition of this input initiates a stop move. While this input is held high, no other move can be queued for this axis.
MC_ WriteBoolPar MC_WriteParam	A positive transition of this input writes the specified parameter.

7.2.20.1.9.2 PLCopen Function Blocks with Enable

These FBs continuously execute every PLC cycle, as long as the Enable remains high.

Function Block	Description
MC_Power	• If this input is high and the drive is currently disabled, this function block requests to close the servo loop and enable the drive. • If this input is low and the drive is currently enabled, this function block requests to open the servo loop and disable the drive.
MC_ ReadActPos	
MC_ ReadActVel	When this input is high, the axis's actual velocity is returned.
MC_ ReadAxisErr	When this input is high, the axis's error status is returned.
MC_ ReadBoolPar MC_ ReadParam	When this input is high, the specified parameter is read.
MC_ ReadStatus	When this input is high, the function block outputs is updated.
MC_ SetOverride	When this input is high, the override factors is written.

7.2.20.1.10.3 PLCopen Functions with Input Parameter En

This is similar to ENABLE on function blocks.
Functions are expected to complete in one cycle.
See "Differences between Functions and Function Blocks" (➔ p. 306).

Function	Description
MC_ AddSuperAxis	When this input is high, the specified axis is added to the superimposed axis list of the receiving axis.
MC_ CreatePLCAxis	When this input is high, a PLCopen axis is created.
MC_EStop	When this input is high, an E-stop is generated for the specified axis.
MC_InitAxis	When this input is high, the specified axis is initialized.
MC_ RemSuperAxis	When this input is high, the specified axis is removed from the superimposed axis list for the specified receiving axis.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Function	Description
MC_ResetError	When this input is high, the specified axis's errors is reset.

7.2.20.1.11 Inputs Exceed Application Limits

If a function block is commanded with parameters which result in a violation of application limits, the instance of the function block generates an error.

7.2.20.1.12 Missing Input Parameters

If any input parameter of a function block is missing (**open**), the compiler generates an error.

7.2.20.1.13 Output Exclusivity

If the **Execute** input of a function block is set TRUE, either the **Busy**, **Done**, **Error**, or **CommandAborted** outputs must also be set TRUE.

- These outputs are mutually exclusive, meaning that a function block may have only one of them set TRUE at any time.
- The output of the MC_GearOut function block is an exception to this exclusivity rule.
- The MC_GearOut is done when the slave axis is disengaged from the master axis.
- When completed, the MC_GearOut remains busy and active until it is aborted by a different motion function block.
- The MC_GearOut function block represents an exception to the exclusivity rule as the **Done** and **Active** outputs can be TRUE at the same time.

7.2.20.1.14 Output Status

The **Done**, **Error**, **ErrorID** and **CommandAborted** outputs are reset with the next rising edge of **Execute**.

If a function block receives a new **Execute** before it finishes (as a series of commands on the same instance), the function block does not return any feedback, like **Done** or **CommandAborted**, for the previous action.

7.2.20.1.15 Sign Rules

- Velocity, Acceleration, Deceleration and Jerk are **always positive** values.
- Position and Distance can be **positive or negative**.

7.2.20.2 PLCopen State Machine

This diagram normatively defines according to PLCopen the behavior of the axis at a high-level when multiple motion control function blocks are simultaneously activated.

- This combination of motion profiles is useful in building a more complicated profile or in handling exceptions within a program.
  - In real implementations there can be additional states defined at a lower level.
- The basic rule is that motion commands are always taken sequentially, even if the PLC has the capability of real parallel processing.
  - These commands act on the state diagram of the axis.
- The axis is always in one of the defined states.
  - A change of state is reflected immediately when issuing the corresponding motion command.
  - The response time of immediately is system dependent.

7.2.20.2.1 Defined States

The defined states are:

- Continuous Motion
- Discrete Motion
- Error Stop
- Homing
- Stand Still

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Stopping
- Synchronized Motion

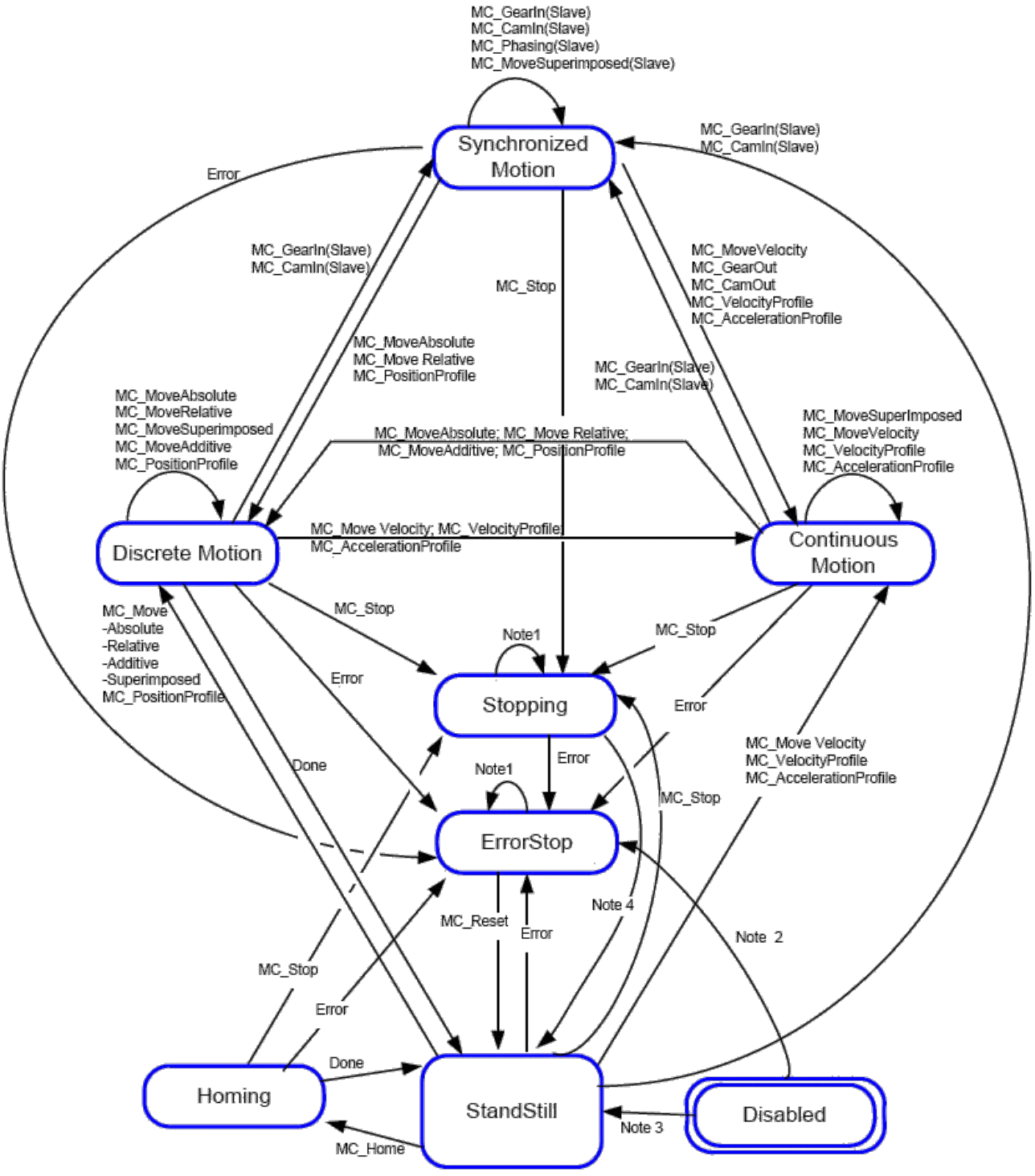


Figure 7-42: Motion State Machine (PLCopen)

Notes

- In the **ErrorStop** or **Stopping** state, all function blocks can be called, although they are not executed. The exceptions are MC_Reset and Error which generate the transition to **StandStill** or **ErrorStop** respectively.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- MC_Power FB is called with Enable=TRUE and there is an error in the Axis.
- MC_Power FB is called with Enable=TRUE and there is **no** error in the Axis.
- MC_Stop.Done **and not** MC_Stop Execute.

7.2.20.2.2.1 State Machine for Multi-axes Motion Control

The diagram is focused on a single-axis.

- The multi-axes function blocks (e.g., MC_CamIn, MC_GearIn, or MC_Phasing) can be looked at, from a state diagram point of view, as multiple single-axes all in specific states.
- Example: The CAM-master can be in the state **Continuous Motion**.
- The corresponding slave is in the state **Synchronized Motion**.
- Connecting a slave axis to a master axis has no influence on the master axis.

A normal procedure starts in **StandStill**.

- In this state the power can be switched on per axis (via the MC_Power command).
  - Access the **Homing** state (via the issue of the command Home per axis).
    This returns to **StandStill** after normal completion.
  - From here, transfer an axis to either **Discrete Motion** or **Continuous Motion**.
  - Via the **Stopping** state you can return to **StandStill**.
  - **ErrorStop** is a state to which the axis transfers in case of error.
  - Via a (manual) MC_ResetError command, return to **StandStill**.
    - From this state, the machine can be moved to an operational state again.

The States define the functionality of the function blocks.

- Function Blocks not listed in the State Diagram do not affect the state of the axis.
  - This means that, whenever they are called, the state does not change.
    They are:
    - MC_CamTblSelect
    - MC_ReadActPos
    - MC_ReadAxisErr
    - MC_ReadBoolPar
    - MC_ReadParam
    - MC_ReadStatus
    - MC_WriteBoolPar
    - MC_WriteParam

7.2.20.2.3 Disabled State

The **Disabled** state describes the initial state of the axis.

Disable means power off without error.

- In this state, the movement of the axis is not influenced by the FBs.
  - The axis feedback is operational.
  - If the MC_Power FB is called with Enable=TRUE while being in **Disabled**, this leads to either:
    - **Standstill** if there is no error inside the axis.
    - **ErrorStop** if an error exists.
  - Calling **MC_Power** with Enable=FALSE in any state, the axis goes to the state **Disabled**, either directly or via any other state.
  - If a motion generating FB controls an axis while the **MC_Power** FB with Enable=FALSE is called, the motion generating FB is aborted (CommandAborted).

7.2.20.2.4 ErrorStop State

The **ErrorStop** state is the axis goes to a stop, if possible.

- No further FBs are accepted until a reset has been done from the **ErrorStop** state.
- The transition Error refers to errors from the axis and axis control, and not from the function block instances.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- These axis' errors are reflected in the output of the function blocks FB instances errors.
- Issuing MC_Home in any other state than **StandStill** goes to **ErrorStop**, even if MC_Home is issued from the **Homing** state itself.
- **ErrorStop** is valid as highest priority and applicable in case of an error.
  - The axis can have either power enabled or disabled, and can be changed via MC_Power.
  - However, as long as the error is pending, the state remains **ErrorStop**.

7.2.20.2.5 From StandStill to Stopping

StandStill is power on without an error.

- Calling the FB MC_Stop in state **StandStill** changes the state to **Stopping** and back to **Standstill** when Execute = FALSE.
- The **Stopping** state is kept as long as the input Execute = TRUE.
- The **Done** output is set when the stop ramp is finished.

7.2.20.3 Program a Multi-Core Controller

A multi-core controller operates slightly differently to take advantage of the extra performance provided by a multi-core CPU.

Consequently, there are some differences in programming a multi-core controller from programming a single-core controller.

This page lists those differences.

7.2.20.3.1 Cached Data

A multi-core controller gains most of its performance by running PLC code and motion code at the same time.

To avoid race conditions, function blocks that read motion data reports the data that the motion code used or calculated during the last cycle.

7.2.20.3.2.1 Function Blocks - Return Cached Data

List of function blocks that return cached data

- MC_AddSuperAxis
- MC_ClearFaults
- MC_EStop
- MC_GrpReadActAcc
- MC_GrpReadActPos
- MC_GrpReadActVel
- MC_GrpReadBoolPar
- MC_GrpReadCmdPos
- MC_GrpReadCmdVel
- MC_GrpReadError
- MC_GrpReadParam
- MC_GrpReadStatus
- MC_GrpWriteBoolPar
- MC_GrpWriteParam
- MC_MoveContVel
- MC_Power
- MC_ReadActPos
- MC_ReadActVel
- MC_ReadAxisErr
- MC_ReadBoolPar
- MC_ReadParam
- MC_ReadStatus
- MC_RemSuperAxis
- MC_ResetError

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- MC_SetPos
- MC_SetPosition - Deprecated
- MC_StopRegist
- MC_WriteParam
- MLAxisAddTq
- MLAxisCmdPos
- MLAxisFBackPos
- MLAxisGenEN
- MLAxisGenIsRdy
- MLAxisGenPos
- MLAxisPipePos
- MLAxisReadActPos
- MLAxisReadGenStatus
- MLAxisReadTq
- MLAxisReadVel
- MLAxisReAlgnRdy
- MLAxisStatus
- MLAxisStop
- MLAxisTimeStamp
- MLBlkIsReady
- MLBlkReadOutVal
- MLCNVConECAT
- MLCNVConnect
- MLCNVConnectEx
- MLCNVDisconnect
- MLMotionStatus
- MLPhaReadActPhase
- MLPmpStatus
- MLSyncWriteDeltaS
- MLTrigClearFlag
- MLTrigIsTrigged
- MLTrigReadPos
- MLTrigReadTime
- MLTrigSetEdge
- Motion / Pipe Network - Comparator
- Motion / Pipe Network - Integrator

See Also

- "Calling Conflicting Pipe Network Functions in the Same Cycle" (➔ p. 389)
- "Calling Function Blocks Multiple Times Same Cycle" (➔ p. 387)

7.2.20.3.3 Calling Function Blocks Multiple Times Same Cycle

A given function block can usually be called multiple times without causing problems when using a single-core controller.

Example

It is possible to call **MC_MoveVelocity** multiple times to start velocity moves on multiple axes.

Set the **Execute** input to **False** between every call to start a new velocity move.

- When doing this, the ability to monitor the status of the velocity moves is lost because the axis parameter is changed for each call.
- If there is no concern with monitoring the status of the velocity moves, call **MC_MoveVelocity** any number of times in a given cycle.

Function blocks that start a motion-related action need to avoid race conditions when using a multi-core controller.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The function blocks have to save argument data so that calculations and error detection can be determined during the next controller cycle.

There are two consequences to this:

- With the possible exception of the **multicoreFbExecuteDuplicateValueInCycle** error "76" (➔ p. 346), the **Error** and **ErrorID** outputs report no error on the first cycle.
  - The **Error** and **ErrorID** outputs reports their status in the next VM cycle.
- The function block cannot be called with the same Execute state twice in a single VM cycle.
  - If a function block is called with the same Execute state twice in a single VM cycle, a **multicoreFbExecuteDuplicateValueInCycle** error "76" (➔ p. 346) is returned immediately.

7.2.20.3.4.1 Function Blocks - Start Motion-related Actions

List of function blocks that start motion-related actions

- MC_AbortTrigger
- MC_AddAxisToGrp
- MC_AxisSetDefaults
- MC_CamIn
- MC_CamOut
- MC_CamResumePos
- MC_CamStartPos
- MC_GearIn
- MC_GearInPos
- MC_GearOut
- MC_GrpDisable
- MC_GrpEnable
- MC_GrpHalt
- MC_GrpReset
- MC_GrpSetPos
- MC_GrpStop
- MC_Halt
- MC_MachRegist
- MC_MarkRegist
- MC_MoveAbsolute
- MC_MoveAdditive
- MC_MoveCircAbs
- MC_MoveCircRel
- MC_MoveContVel
- MC_MoveDirAbs
- MC_MoveDirRel
- MC_MoveLinAbs
- MC_MoveLinRel
- MC_MoveRelative
- MC_MoveSuperimp
- MC_MoveVelocity
- MC_Phasing
- MC_Reference
- MC_RemAxisFromGrp
- MC_SetPos
- MC_Stop
- MC_SyncSlaves
- MC_TouchProbe
- MC_UngroupAllAxes

See Also

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Calling Conflicting Pipe Network Functions in the Same Cycle" (➔ p. 389)
- "Program a Multi-Core Controller" (➔ p. 386)

7.2.20.3.5 Calling Conflicting Pipe Network Functions in the Same Cycle

With the Pipe Network motion engine, it is possible to execute conflicting actions in the same cycle.

Example

In this example, an axis is commanded to stop followed immediately by a command to realign the axis.

Here, the realign command overwrites the command to stop the axis.

```
MLAxisMoveVel( PipeNetwork.AXIS1, 0 ); // Stop the axis
MLAxisReAlign( PipeNetwork.AXIS1, LREAL#360.0, LREAL#360.0, LREAL#360.0,
LREAL#0.0);
```

- While this example is easy to see, it is possible that conflicting actions happen in different programs, subprograms, or user-defined function blocks (UDFBs).
- When conflicting actions occur in different parts of the application, it can be confusing why some action does not occur while debugging.
- To help users identify conflicting actions, when the controller runtime detects conflicting actions, a warning is generated at runtime.

This example generates this motion warning:

```
Axis pipe block: 'MLAxisReAlign' action overrode previous action
'MLAxisMoveVel'.
```

When motion warnings occur that state ‘action overrode previous action’ occur, it is recommended that the application be updated to avoid producing these warnings.

7.2.20.3.6.1 Pipe Network Functions - Motion-related Actions and Parameters

List of Pipe Network functions that start motion-related actions or set motion-related parameters

- MLAddWriteInput
- MLAddWriteOff1
- MLAddWriteOff2
- MLAddWriteRat1
- MLAddWriteRat2
- MLAxisAbs
- MLAxisAdd
- MLAxisAddTq
- MLAxisCfgFastIn
- MLAxisGenEN
- MLAxisGenWriteAcc
- MLAxisGenWriteDec
- MLAxisGenWriteSpd
- MLAxisMoveVel
- MLAxisPower
- MLAxisRatedTq
- MLAxisReAlign
- MLAxisRel
- MLAxisRstFastIn
- MLAxisStop
- MLAxisWriteModPos
- MLAxisWritePipPos

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- MLAxisWritePos
- MLAxisWriteUUnits
- MLBlkWriteModPos
- MLCNVConnect
- MLCNVDisconnect
- MLCompReset
- MLCompWriteRef
- MLDerWriteInModPos
- MLGearWriteOff
- MLGearWriteOSlp
- MLGearWriteRatio
- MLGearWriteRatSlp
- MLIntWriteOutVal
- MLMstAbs
- MLMstAdd
- MLMstForcePos
- MLMstRel
- MLMstRun
- MLMstWriteAccel
- MLMstWriteDecel
- MLMstWriteInitPos
- MLMstWriteSpeed
- MLPhaWritePhase
- MLPhaWriteSlope
- MLPmpAbs
- MLPmpForcePos
- MLPmpRel
- MLPmpRun
- MLPmpWriteAccel
- MLPmpWriteFstSpd
- MLPmpWriteJerk
- MLPmpWriteLstSpd
- MLPrfWriteIOffset
- MLPrfWriteIScale
- MLPrfWriteOOffset
- MLPrfWriteOScale
- MLSyncStart
- MLSyncStop
- MLSyncWriteDeltaS
- MLTrigClearFlag
- MLTrigSetEdge
- MLTrigWriteDelay

See Also

- "Calling Function Blocks Multiple Times Same Cycle" (➔ p. 387)
- "Program a Multi-Core Controller" (➔ p. 386)

7.2.21 Recipes

This documentation describes how to use Recipes in KAS.

It is intended for:

- Machine builders (OEMs).
- System integrators and distributors.
- Controls Engineers.
- End users and operators.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Recipes enable efficient handling of machine configuration data for different products, improving flexibility, consistency, and traceability.

7.2.21.1 What is a Recipe?

In industrial automation, a recipe is a named collection of parameter values that together define the configuration for a particular manufacturing process or product variant, sometimes referred to as a SKU.

Rather than hard-coding process parameters into the control program itself, recipes allow operators and engineers to separate what the machine does (the program logic) from how it is configured for a given job (the data).

For a PLCopen-based controller, a recipe file is:

- A structured data set that provides motion-related parameters (e.g., process values and kinematic parameters).
- Loaded at runtime and applied by PLCopen motion function blocks.

An application control logic can remain unchanged while a recipe dynamically configures how machine behavior and motion profiles are executed for a specific product or operation.

Recipes are a foundational concept in batch and discrete manufacturing and are formally described in the ISA-88 (IEC 61512) standard for batch process control.

See ISA-88 Series of Standards.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.2 What Problems Do Recipes Solve?

Modern manufacturing lines frequently produce multiple product variants on the same equipment. Without recipe management, engineers face a difficult choice:

- Hard-code all variants into the program - resulting in complex, branching logic that is difficult to maintain and error-prone to modify.
- Manually adjust parameters at the HMI or in the source code - a slow, undocumented, and risky process that depends entirely on operator knowledge.

Recipe management eliminates both of these problems by providing a structured, systematic mechanism for storing and applying process parameters.

Problem	How Recipes Help
Product changeovers are slow.	A single operator action loads a complete parameter set instantly.
Parameter changes are undocumented.	Recipe files are named, versioned, and stored in a traceable format.
Operators can enter invalid values.	Recipe variables carry type information and can enforce validation limits.
Process knowledge is not documented.	Parameters are captured in files that can be backed up, shared, and audited.
Scaling to new variants requires code changes.	A new recipe file is created with no changes to the control program.

7.2.21.2.1 Example Use Cases

Example Industry	Use Case
Food and Beverage	Switching between product formulations (e.g., flavors, concentrations).
Pharmaceuticals	Batch manufacturing with strict parameter traceability requirements.
Labeling	Store label offset, print speed, and dispense timing for different container sizes.
Agricultural Automation	Define dryer temperature, airflow, and target moisture content for different grain types (e.g., corn, soybeans, wheat).
Laser Notching / Cutting	Define laser power, pulse frequency, cutting speed, and focus offset for different parts.

7.2.21.3 Core Concepts

The recipe system uses these terms:

Term	Meaning	Example
"Recipe Definition" (→ p. 392)	A group of recipe variables that belong together.	Processing parameters or bottling parameters.
"Recipe Files" (→ p. 393)	The files that stores definitions, instances, or both.	processing.xml or masterrecipes.toml.
"Recipe Instance" (→ p. 393)	The actual values used for one product, batch, or setup.	FizzNova settings or BubbleRush settings.
"Recipe Variable" (→ p. 394)	A single parameter mapped to a PLC program variable.	MixSpeedRPM, ProductName, FillPressure.

7.2.21.3.1 Recipe Definition

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

A recipe definition groups related variables together that collectively describe one configurable aspect of the machine.

Multiple recipe sets can exist in a project, each mapped to different parts of the program.

7.2.21.3.2 Recipe Files

Recipe files may contain:

- A recipe definition.
- Recipe instances.
- User-defined areas for non-recipe data or audit tracking fields.

Recipe files are designed to be:

- Saved and loaded at runtime.
- Transferred between machines.
- Archived for audit or compliance purposes.
- Edited offline.

7.2.21.3.3 Recipe Instance

A recipe instance is a set of values for all recipe variables associated with a recipe definition.

A recipe instance typically defines values for a single part, formulation, or other product.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.3.4 Recipe Variable

A recipe is composed of recipe variables - individual parameters that are mapped to variables in the PLC program.

Each recipe variable has:

- A name that matches a variable in a PLC program.
- A data type (e.g., BOOL, INT, REAL, STRING).
  - See Data Types.
- A value used by the PLC program.
- Optional: Traits that define minimum/maximum limits and clipping actions.

IMPORTANT

- Elements of arrays and structures can be part a recipe definition.
- To set all values of an array or structure, each element must be set individually.

Trait	Description
HasMax	The recipe variable has a maximum value.
HasMin	The recipe variable has a minimum value.
OnOutOfRange_Clip	When a recipe instance is loaded and the value is out of range, the value is clipped. • The definition of clip is: • If the value is less than the minimum, the minimum value is set. • If the value is greater than the maximum, the maximum value is set. • See numpy.clip.
OnOutOfRange_Error	When a recipe instance is loaded and the value is out of range, an error is generated.
OnOutOfRange_Ignore	When a recipe instance is loaded and the value is out of range, the variable is skipped.

7.2.21.3.5.1 Data Types - Supported

- Bitfields
- BOOL
- BYTE
- DINT
- DWORD
- Enumerations
- INT
- LINT
- LREAL
- LWORD
- REAL
- SINT
- STRING
- TIME
- UDINT
- UINT
- ULINT
- USINT
- WORD

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.4 Quick Start: Basic Recipe Workflow

This is a very basic, overall recipe workflow procedure.

Procedure

1. Create or edit a recipe file in a supported format (e.g., XML or TOML).
2. Verify the recipe variable names match the PLC program variables.
3. Transfer the recipe file to the controller User Data area.
4. In the KAS project application code:
   a. Use the appropriate recipe function block to open, load, apply, save, or close the recipe at runtime.
   b. Monitor Done, Busy, Error, and ErrorID outputs to confirm successful execution or diagnose issues.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.5 Recipe File Formats and Examples

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.5.1 Reserved Recipe Sections

IMPORTANT

- Do not use “Recipe” as a section name prefix.
- This designation is reserved for Kollmorgen’s KAS development team for feature enhancement purposes.
- Examples:
  - **NOT Acceptable**: RecipeOptions
  - **Acceptable**: Options_Recipes, OptionsRecipes, Options-Recipes

7.2.21.5.2 Choosing XML or TOML

KAS supports XML and TOML file formats for recipe files.

- Use XML when the workflow already relies on XML tools, schema validation, or existing automation.
- Use TOML when readability and simple hand editing are more important.

IMPORTANT

- In either format, keep variable names, data types, and instance names consistent with the PLC application.

7.2.21.5.3 Example: Recipe Definition File

NOTE

- The TOML format requires single keys with periods or brackets to be quoted.

XML

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Recipe>
  <RecipeDefinition>
    <Variable name="MyGlobalINT" type="INT"
      traits="HasMin|HasMax|OnOutOfRange_Clip"
      min="-100" max="100" />
    <Variable name="MyProgram.MySTRING" type="STRING" />
    <Variable name="MyProgram.Complex.Var[1]" type="DINT" />
  </RecipeDefinition>
</Recipe>
```

TOML

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TOML

```
[recipe_definition]
[[recipe_definition.variable]]
name = 'MyGlobalINT'
type = 'INT'
traits = ['HasMin', 'HasMax', 'OnOutOfRange_Clip']
min = -100
max = 100

[[recipe_definition.variable]]
name = 'MyProgram.MySTRING'
type = 'STRING'

[[recipe_definition.variable]]
name = 'MyProgram.Complex.Var[1]'
type = 'DINT'
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.5.4 Example: Recipe Definition and Multiple Instance File

NOTE

- In the TOML format, each [[recipe_instance]] entry uses a [recipe_instance.meta] sub-table for instance metadata (e.g., name) and a [recipe_instance.data] sub-table for recipe variable values.
  - This separation ensures that metadata keys can never collide with recipe variable names.
- The [recipe_instance.meta] sub-table is optional; omitting it produces an unnamed instance.
- The TOML format requires single keys with periods or brackets to be quoted.

XML

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Recipe>
  <RecipeDefinition>
    <Variable name="StampProgram.PartCount"
      type="INT" traits="HasMin|HasMax|OnOutOfRange_Clip"
      min="1" max="100" />
    <Variable name="LabelProgram.PartName" type="STRING" />
  </RecipeDefinition>
  <RecipeInstance>
    <!-- A RecipeInstance without a name is
         assigned the name "Instance" -->
    <Variable name="StampProgram.PartCount">0</Variable>
    <Variable name="LabelProgram.PartName">Budget Boff</Variable>
  </RecipeInstance>
  <RecipeInstance name="Cogs">
    <Variable name="StampProgram.PartCount">10</Variable>
    <Variable name="LabelProgram.PartName">Classy Cog</Variable>
  </RecipeInstance>
  <RecipeInstance name="Widgets">
    <Variable name="StampProgram.PartCount">20</Variable>
    <Variable name="LabelProgram.PartName">Wonderful Widget</Variable>
  </RecipeInstance>
</Recipe>
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TOML

TOML

```
[recipe_definition]
[[recipe_definition.variable]]
name = 'StampProgram.PartCount'
type = 'INT'
traits = ['HasMin', 'HasMax', 'OnOutOfRange_Clip']
min = 1
max = 100

[[recipe_definition.variable]]
name = 'LabelProgram.PartName'
type = 'STRING'

[[recipe_instance]]
# A recipe instance without a [recipe_instance.meta]
# sub-table is given the name 'Instance'
[recipe_instance.data]
'StampProgram.PartCount' = 0
'LabelProgram.PartName' = 'Budget Boff'

[[recipe_instance]]
[recipe_instance.meta]
name = 'Cogs'
[recipe_instance.data]
'StampProgram.PartCount' = 10
'LabelProgram.PartName' = 'Classy Cog'

[[recipe_instance]]
[recipe_instance.meta]
name = 'Widgets'
[recipe_instance.data]
'StampProgram.PartCount' = 20
'LabelProgram.PartName' = 'Wonderful Widget'
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.5.5 Example: Recipe Definition and Single Instance File

NOTE

- If a recipe instance is not given a name, the instance is assigned the name "Instance".
- This example does not assign a name.
- The TOML format requires single keys with periods or brackets to be quoted.

XML

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Recipe>
  <RecipeDefinition>
    <Variable type="INT" name="StampProgram.PartCount"
      traits="HasMin|HasMax"
      min="1" max="100" />
    <Variable type="STRING" name="LabelProgram.PartName" />
  </RecipeDefinition>
  <RecipeInstance>
    <Variable name="StampProgram.PartCount">0</Variable>
    <Variable name="LabelProgram.PartName">Budget Boff</Variable>
  </RecipeInstance>
</Recipe>
```

TOML

TOML

```
[recipe_definition]
[[recipe_definition.variable]]
name = 'StampProgram.PartCount'
type = 'INT'
traits = ['HasMin', 'HasMax']
min = 1
max = 100

[[recipe_definition.variable]]
name = 'LabelProgram.PartName'
type = 'STRING'

[[recipe_instance]]
[recipe_instance.data]
'StampProgram.PartCount' = 0
'LabelProgram.PartName' = 'Budget Boff'
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.5.6 Example: Recipe Definition and User Audit Entries

NOTE

- Recipes may include custom tags that can be used for auditing purposes.
- The user is responsible for adding audit-related tags to the recipe file.
- The TOML format requires single keys with periods or brackets to be quoted.

XML

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Recipe>
  <!-- Audit Information -->
  <Version>
    <Revision>4</Revision>
    <ChangeLog>Changed temperature</ChangeLog>
  </Version>

  <CreatedBy>
    <Username>j.smith</Username>
    <EmployeeID>EMP-04471</EmployeeID>
    <Role>Process Engineer</Role>
  </CreatedBy>

  <!-- Paint Process Information -->
  <RecipeDefinition>
    <Variable name="CuringProgram.Temperature"
      type="INT" traits="HasMin|HasMax"
      min="-10" max="250" />
    <Variable name="PaintProgram.ColorName"
      type="STRING" />
  </RecipeDefinition>

  <RecipeInstance name="Panels">
    <Variable name="CuringProgram.Temperature">85</Variable>
    <Variable name="PaintProgram.ColorName">Blue</Variable>
  </RecipeInstance>
</Recipe>
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TOML

TOML

```
#Audit Information
[version]
revision = 4
change_log = 'Changed temperature.'

[author.created_by]
username = 'j.smith'
employee_id = 'EMP-04471'
role = 'Process Engineer'

#Paint Process Information
[recipe_definition]
[[recipe_definition.variable]]
name = 'CuringProgram.Temperature'
type = 'INT'
traits = ['HasMin', 'HasMax']
min = -10
max = 250

[[recipe_definition.variable]]
name = 'PaintProgram.ColorName'
type = 'STRING'

[[recipe_instance]]
[recipe_instance.meta]
name = 'Panels'
[recipe_instance.data]
'CuringProgram.Temperature' = 85
'PaintProgram.ColorName' = 'Blue'
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.5.7 Example: Recipe Instance File

NOTE

- Recipe instance files are only used with RecipeQuickLoad.
- The TOML format requires single keys with periods or brackets to be quoted.

XML

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Recipe>
  <RecipeInstance>
    <Variable name="MyProgram.MyINT">42</Variable>
    <Variable name="MyGlobalREAL">75.5</Variable>
    <Variable name="Test.Message">Testing</Variable>
    <Variable name="Test.Time">TIME#10.5s</Variable>
    <Variable name="MyProgram.Complex.Var[1]">17</Variable>
  </RecipeInstance>
</Recipe>
```

TOML

TOML

```
[[recipe_instance]]
[recipe_instance.data]
'MyProgram.MyINT' = 42
'MyGlobalREAL' = 75.5
'Test.Message' = 'Testing'
'Test.Time' = 'TIME#10.5s'
'MyProgram.Complex.Var[1]' = 17
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.6 Recipe Management

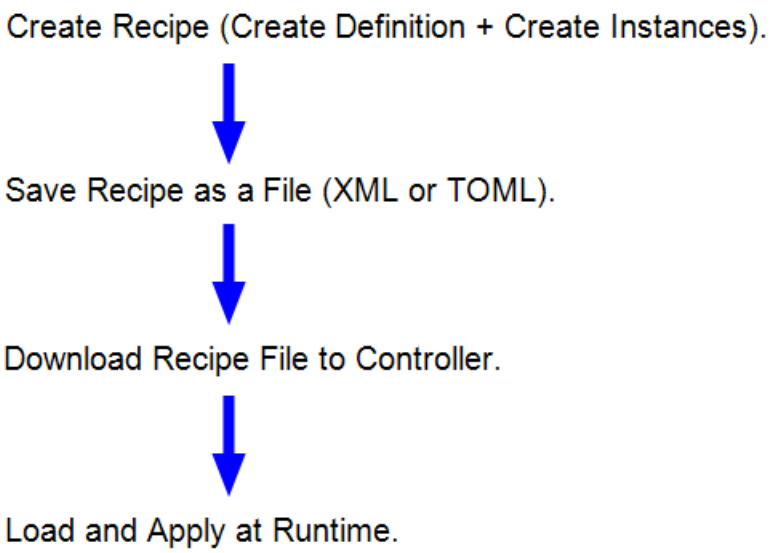
Recipe Management enables the creation, storage, transfer, and application of recipes in a controlled and repeatable way.

Recipe Management typically consists of:

- Engineering Workflow - Design time configuration and setup.
- Operational Workflow - runtime selection and execution.

7.2.21.6.1.1 Recipe Workflow

The typical lifecycle of a recipe is:



7.2.21.6.2 Create and Edit Recipe Files

Recipe files are used to define and store parameter sets outside the application logic.

This allows flexibility without modifying PLC code.

Procedure

NOTE

- Recipe Files can be created using any text editor (e.g., Notepad++, XmlNotepad, XMLBlueprint, TOML Editor for Visual Studio, etc.)
- It is recommended to use an editor that supports the file format you intend to use (E.g., XML or TOML).

1. Create a Recipe Definition:
   a. Define all required variables.
   b. Assign names, data types, and optional validation rules.
2. Add Recipe Instances:
   a. Provide values for each variable.
   b. Assign a name for the instance (recommended for traceability).
3. Save the file in a supported format.

Example Recipe File Structure

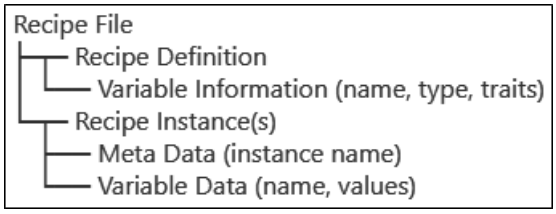


Figure 9-1: Recipe File Structure

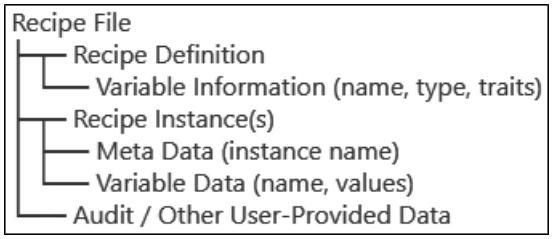


Figure 9-2: Recipe File Structure with Audit Data

- With a recipe set defined, individual recipe files are created in the editor.
- Existing recipe files can be duplicated as a starting point for new variants.
  - This is useful when recipes share most of their parameters and differ only in a few values.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.6.3 Download Recipes to the Controller

Recipe files must be transferred to the controller before they can be used at runtime.

- Recipe files are stored in the controller's persistent file system (Internal Memory or USB flash drive) and are retained across power cycles.

Methods of file transfer:

- "Transfer via Controller Web Server" (→ p. 408).
- "Transfer via HTTP API" (→ p. 411).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.2.21.6.4.1 Transfer via Controller Web Server

The User Data tab provides access to the User Data Storage area on the Controller's internal flash memory. (Figure 9-3)

- This area is intended for user-managed file storage to support PLC application needs.
- The system does not enforce structure or usage – it is entirely up to the user to organize and maintain the content.

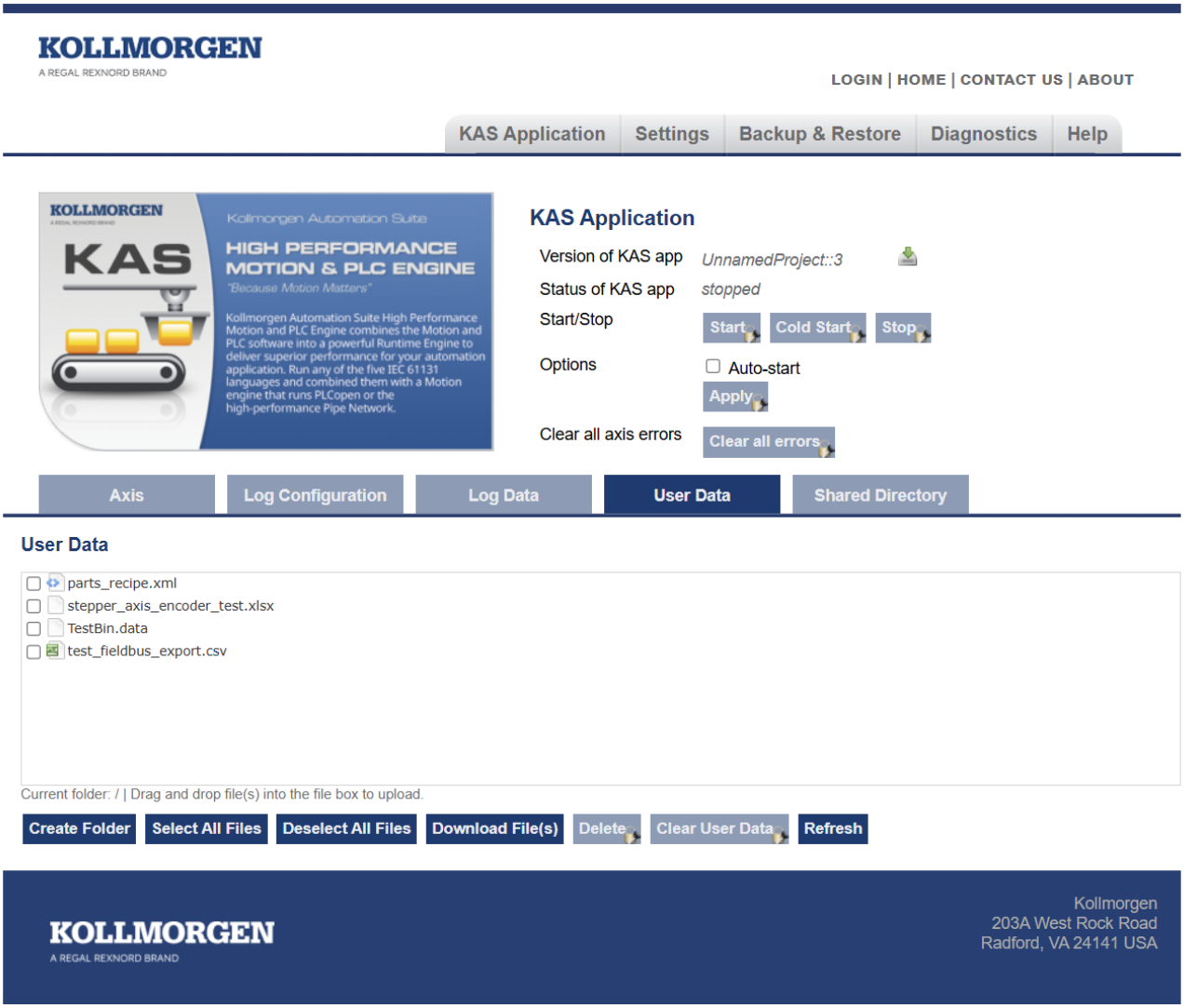


Figure 9-3: User Data tab

The User Data area is flexible, user-controlled storage space where you can maintain folders and files required for application management.

Typical use cases include:

- Storing and updating recipe files.
- Maintaining lookup tables or datasets.
- Exchanging data between external systems and the controller.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

User Data File Explorer

- Display all folders and files currently stored in the User Data area on the controller flash.
- Each folder and file has a check box for selecting an action.

IMPORTANT

- Folders can only be deleted.

7.3 View File Contents from the User Data File Explorer

1. Double-click any file in the list.
   The file contents open in a separate tab. (Figure 9-4)



Figure 9-4: User Data tab - View File Contents

This new tab:

- Provides quick access for inspecting file data without downloading.
- Is useful for verifying configuration or recipe files.

7.4 Upload Folders and Files to the Controller

Folders and files can be uploaded to the PCMM2G controller flash using drag-and-drop.

1. On your computer, open Windows Explorer.
2. Locate and select the applicable folders and/or files.
3. Drag and drop the selected items into the User Data area.

   - Folders and files are stored immediately in the controller's User Data area.
   - Multiple folders and files can be uploaded at once.
   - There is no upload button – drag-and-drop is the only available method.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.5 File Management Actions

All actions apply directly to the folders and files stored on the controller flash:

7.5.1 Create Folder

- Create a new folder in the current directory.

7.5.2 Select All Files

- Selects all files in the list.
- Primarily used to support bulk operations such as:
  - Downloading multiple files.
  - Deleting multiple files.

7.5.3 Deselect All Files

- Deselects all selected files.

7.5.4 Download Files

- Downloads selected files from the controller to your local computer.

IMPORTANT

- Downloading folders is not supported.

7.5.5 Delete

- Deletes the selected folders and files from the controller.

WARNING

- Before deleting folders and files, confirm they are no longer required by the application and create a backup if needed.
- Deleting folders and files cannot be undone.

7.5.6 Clear User Data

- Deletes all folders and files in the User Data area.

WARNING

- Use with caution - this removes all user-managed content.
- This action cannot be undone.

7.5.7 Refresh

- Updates the file explorer to reflect current contents on the controller.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.5.7.0.1.1 Transfer via HTTP API

This is the process to manage the user data area on the controller flash memory from an external device (e.g., an HMI) using HTTP communications.

This method uses cURL (http://curl.haxx.se), a command line tool for performing HTTP communication with a server.

IMPORTANT

- The disk space available for the user data area in the controller flash is very limited.
  - It is strongly recommended to delete the files that are no longer necessary.
- The user files in the controller flash can be deleted using either HTTP commands or from the PLC application.
  - See File Management for available file function blocks.

Procedure

1. **List directories and files** present on the controller flash's user data folder:

```
curl.exe -k https://(IP_Address)/userfiles

Example Result:
{
   "Directories" : [
      {
         "Name" : "diagnostics"
      }
   ],
   "Files" : [
      {
         "Name" : "ScanTimes.csv"
      }
   ]
}
```

2. **Upload files** onto the controller flash's user data folder.
   Two HTTP requests are sent:
   - The first request is to authenticate.
   - The second request is to upload the file.

```
curl.exe -k -X POST -d "username=administrator" -d
"password=administrator" -G https://(IP_Address)/authorize/login -c
cookie.txt

curl.exe -k -F "sendfile=@C:\tmp\recipe.xml" -F "filename=recipe.xml"
https://(IP_Address)/userfiles -b cookie.txt
```

   - The first request is a POST used to provide credentials to the webserver.
     - The cookie received from the server is stored in a temporary **cookie.txt** file.
       - This is used in the second request.
     - The username, password, and IP address must be customized for your system.
   - The second request uploads the file using a HTTP Multipart file upload request.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The `-b cookie.txt` at the end is used to add the authentication cookie received after the first request.
- The IP address, sendfile, and filename must be customized for your system.

NOTE

The first request to provide the credentials to the webserver is not required if the previously logged-in session is still active.

3. **Deleting files** from the controller flash's user data folder:

```
curl.exe -k -X POST -d "username=administrator" -d
"password=administrator" -G https://(IP_Address)/authorize/login -c
cookie.txt

curl.exe -k -X DELETE https://(IP_Address)/userfiles/recipe.txt -b
cookie.txt
```

TIP

- cURL exists as a C library.
- It can be used in compiled software which can be used in the HMI.
- If cURL is not an option, equivalent tools can be used if they can POST and PUT HTTP requests and support cookies.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.5.7.0.2 Runtime-based Recipe Management through Function Blocks

For applications that require programmatic control over recipe operations, a set of function blocks are available for recipe management.

- These function blocks give the PLC program full control over recipe operations at runtime.
- Examples:
  - Automatic changeovers triggered by a HMI selection.
  - Barcode scan.
  - Manufacturing execution system integration.
  - Sequence-driven production.

Important Notes

- Always monitor these outputs from the function blocks:
  - Done
  - Busy
  - Error
- Reset the Execute input between operations.
- Do not trigger the next function until the previous one completes.

Function Block	Description
RecipeApplyInstance	Copy values to from a recipe instance to the program data.
RecipeClose	Close a recipe object.
RecipeOpen	Open a recipe file and create a recipe object.
RecipeQuickLoad	Load a recipe file and immediately apply the items it contains.
RecipeSave	Save a recipe object to a file.
RecipeSaveRetainVars	Save all retain variables, and their values, into a new recipe file.
RecipeStoreInstance	Store the present values of the program data in a recipe instance.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.5.7.1 Recipe Design Example

This example is a beverage production line using a PLCopen-based motion controller.

The system requires specific parameters for different soda brands, which map to physical tasks like variable-speed mixing and high-speed bottling synchronization. This example examines different approaches to designing recipe files and discusses the advantages and disadvantages of each approach.

7.5.7.1.1 Variable Configurations

Instead of maintaining seven separate data structures for minor process steps, parameters are grouped into these core execution phases:

- "Example - Processing Parameters (Thermal and Mixing)" (➔ p. 414)
- "Example - Carbonation Parameters (Pressure Controls)" (➔ p. 414)
- "Example - Motion and Bottling Parameters (Cam Profiles and Packaging)" (➔ p. 414)

7.5.7.1.2 Example - Processing Parameters (Thermal and Mixing)

PLC Variable	Data Type	FizzNova	BubbleRush	PrismPop
sugarKg	REAL	3.0	2.5	1.8
waterL	REAL	25.0	30.0	36.0
mixSpeedRPM	UINT	60	50	70
heatTempC	REAL	1200	1000	900

7.5.7.1.3 Example - Carbonation Parameters (Pressure Controls)

PLC Variable	Data Type	FizzNova	BubbleRush	PrismPop
chillTempC	REAL	2.0	3.0	1.5
co2PressureBar	REAL	3.5	4.0	3.0

7.5.7.1.4 Example - Motion and Bottling Parameters (Cam Profiles and Packaging)

PLC Variable	Data Type	FizzNova	BubbleRush	PrismPop
containerType	STRING	'Can330ml'	'Bottle500ml'	'Bottle2000ml'
fillSpeedBPM	UINT	180	150	170
packCount	UINT	6	12	1

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.5.7.1.5 Architectural Approaches and Tradeoffs

When deploying these parameters to a PLCopen controller, engineers must choose between a distributed or a centralized file structure.

- "Approach A: Multiple Modular Recipe Files" (➔ p. 415)
- "Approach B: Single Master Recipe" (➔ p. 415)

7.5.7.1.6.1 Approach A: Multiple Modular Recipe Files

Each manufacturing stage reads its own independent recipe file (e.g., processing.xml, carbonation.xml, bottling.xml).

How it Operates

The program invokes RecipeApplyInstance independently at each stage of the physical line only when a new batch arrives at that zone.

Advantages

- **Asynchronous Execution**: The bottling line can pack FizzNova while the mixing tanks are already preparing the next batch of BubbleRush.
- **Memory Efficiency**: Lowers peak PLC controller memory overhead by only loading localized variables into active memory registers.

Disadvantages

- **File Sprawl**: Managing, validating, and auditing 3 to 7 separate file updates per SKU drastically increases HMI / SCADA synchronization overhead.

7.5.7.1.7.2 Approach B: Single Master Recipe

A single XML configuration file (e.g., master_recipes.xml) houses all parameter tables merged into one structural dataset per product variant / SKU.

How it Operates

A single call to RecipeApplyInstance downloads all processing, pressure, and motion variables simultaneously into global variables or system data blocks at the start of a production run.

Advantages

- **Simplified Auditing**: A single master file streamlines version control, recipe modification, and regulatory data tracking.
- **Atomic Updates**: Guarantees that all machine zones stay perfectly synchronized on the exact same product variant without risking a data mismatch between stages.

Disadvantages

- **Tracking Complexity**: Because all variables overwrite instantly at the beginning of the line, the PLC application logic must internally track and pass motion and bottling variables down the line along with the physical product.

7.5.7.2 Recipe Troubleshooting

- When a recipe function block fails, check the value of the function block ErrorID output against the table in File, Recipe, and TCP/IP Function Block s.
- The specific error code is used to determine the proper corrective action.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Issue	Possible Cause	Recommended Action
Recipe file does not load.	• Invalid format. • Missing file. • Incorrect file path.	• Check the function block ErrorID output. • Verify the file exists in the expected controller location. • Confirm the selected format matches the file contents.
Recipe instance is not applied.	• The instance name does not match. • The file does not contain the expected instance.	• Check the function block ErrorID output. • Verify the instance name.
Values are not written to program data.	• Variable names do not match PLC program variables. • Variable values are invalid.	• Check the function block ErrorID output. • Compare each recipe variable name against the PLC application variable name. • Verify the variable values are in the specified ranges and contain no typos.

7.6 EtherCAT Motion Bus Concepts

To exchange data between the controller (master) and the devices (slaves), the KAS Runtime relies on the EtherCAT motion bus.

This communication is done in two modes: cyclic and non-cyclic (mailbox).

In **cyclic mode**, a single frame containing the data of all slaves (input and output) travels along all slaves and goes back to the master.

Data is read and/or written on the fly by each slave.

The slave device input and output data definitions are:

- Outputs are written by the master and read by the slave device.
- Inputs are written by the slave device and read by the master.

7.6.1 EtherCATProcess Image

This cyclic frame is called the EtherCAT Process Image.

- It contains the Process Data, which is defined during network initialization.
- The cyclic data is grouped in predefined blocks called Process Data Objects or PDO.

NOTE

PDOs contain real-time cyclic data which is deterministic.
Non-cyclic data is not deterministic and is defined by Service Data Objects (ScDO).

7.6.1.1 References

- EtherCAT Specification V1.0 - see http://www.ethercat.org in the Member Area - Downloads.
- Büttner, H.; Janssen, D.; Rostan, M. (2003), EtherCAT - the Ethernet fieldbus, (PDF), PC Control Magazine 3: 14-19.

7.6.2 EtherCAT Functional Principle

Automation networks are characterized by short data-length per node, typically less than the minimum payload of an Ethernet frame. Using one frame per node per cycle leads to low bandwidth utilization and thus to poor overall network performance.

EtherCAT takes a different approach, called **processing on the fly**.

See "EtherCAT Implementation" (→ p. 423).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- With EtherCAT, the Ethernet packet or frame is no longer received, and then interpreted and copied as process data at every node.
- Instead, the EtherCAT slave devices read the data addressed to them while the telegram passes through the device.
  - Input data is inserted while the telegram passes through.
- The frames are only delayed by a fraction of a microsecond in each node, and many nodes - typically the entire network - can be addressed with just one frame.

7.6.3 EtherCAT Features

EtherCAT is characterized by outstanding performance, very simple wiring, and openness to other protocols. EtherCAT sets new standards where conventional fieldbus systems reach their limits:

- 1000 I/Os in 30 µs.
- Optionally twisted-pair cable or optical fiber.
- Optimum vertical integration (thanks to Ethernet and Internet technologies).

With EtherCAT, the costly Ethernet star topology is replaced with a simple line structure - no expensive infrastructure components are required.

- Optionally, EtherCAT can be wired in the classic way using switches, to integrate other Ethernet devices.
- Where other real-time Ethernet approaches require special connections in the controller, for EtherCAT, very cost-effective standard Ethernet cards suffice.

7.6.3.1 EtherCAT is Versatile

These communication connections are supported (see image):

- Master to Master
- Master to Slave
- Slave to Slave

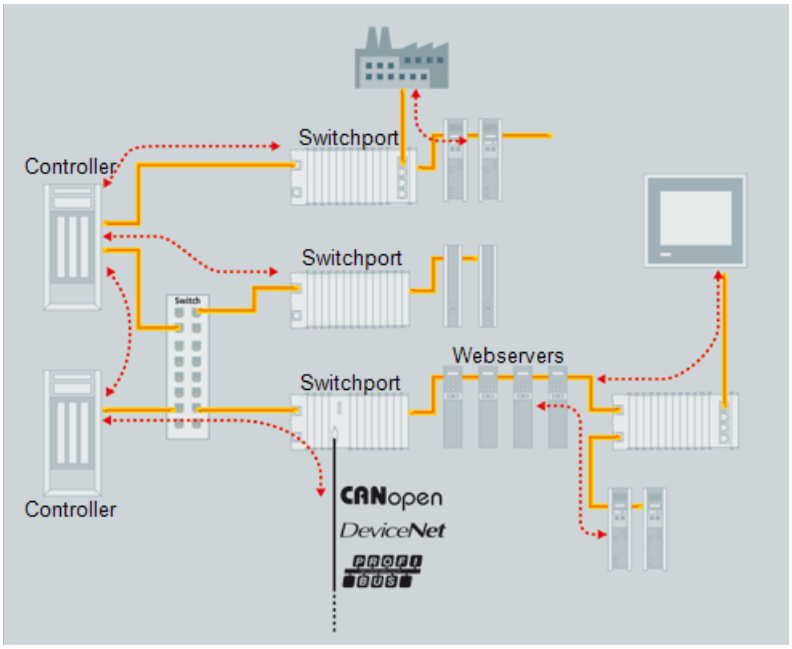


Figure 10-1: Versatile Network Architecture

7.6.3.2 Safety over EtherCAT is Available

- EtherCAT makes Ethernet down to the I/O level technically feasible and economically sensible.
- Outstanding features of this network include:
  - Full Ethernet compatibility.
  - Internet technologies (even in simple devices).
  - Maximum utilization of the large bandwidth offered by Ethernet.
  - Outstanding real-time characteristics at low costs.

7.6.3.3 EtherCAT Device Profiles

Device profiles describe the application parameters and the functional behavior of the devices, including the device class-specific state machines.

- For many device classes, fieldbus technology already offers reliable device profiles, such as for I/O devices or drives.
  - Users are familiar with these profiles and the associated parameters and tools.
- No EtherCAT-specific device profiles have been developed for these device classes.
  - Simple interfaces for existing device profiles are offered.
  - This greatly assists users and device manufacturers alike during the change from existing fieldbuses to EtherCAT.

7.6.3.3.1 CANopen over EtherCAT (CoE)

CANopen device and application profiles are available for a wide range of device classes and applications.

Devices include I/O components, drives, encoders, proportional valves and hydraulic controllers to application profiles for plastic or textile machinery.

- EtherCAT can provide the same communication mechanisms as the familiar CANopen mechanisms: object dictionary, PDO (process data objects) and SDO (service data objects), and even the network management is comparable.
- EtherCAT can be implemented with minimum effort on devices equipped with CANopen.
- Large parts of the CANopen firmware can be re-used.
- Objects can optionally be expanded in order to account for the larger bandwidth offered by EtherCAT.

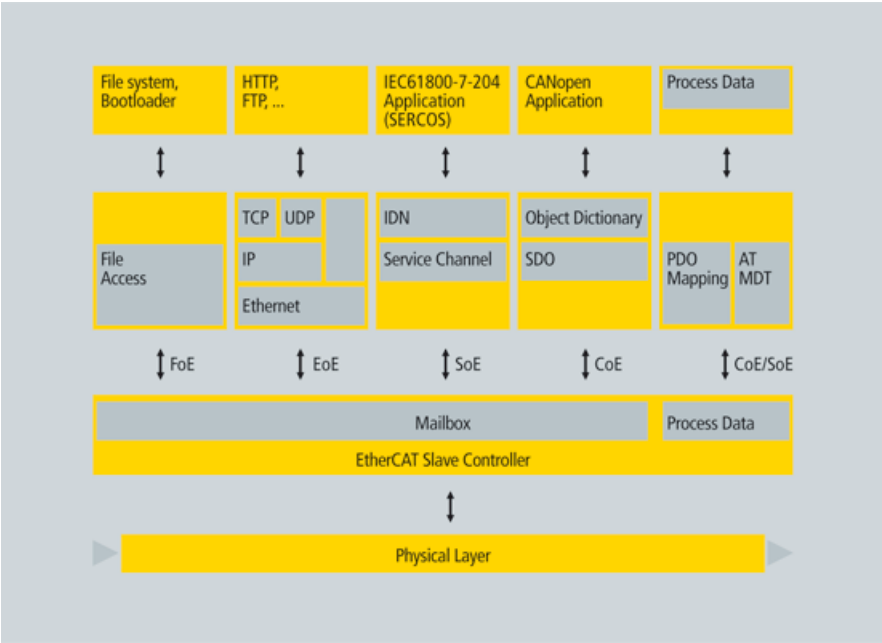


Figure 10-2: Several Device Profiles and Protocols can Coexist

7.6.3.4 EtherCAT Distributed Clock (Synchronization)

A distributed clock is an EtherCAT feature that allows synchronization, with a reference clock, of all EtherCAT slaves and the master. This solves problems related to clock-shifting between the master and the devices.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This mechanism leads to very low jitter of significantly less than 1µs. Even if the communication cycle jitters, it is still compliant with the IEEE 1588 Precision Time Protocol standard.

EtherCAT does not require special hardware in the master device and can be implemented in software on any standard Ethernet MAC, even without a dedicated communication coprocessor.

- The typical process of establishing a distributed clock is initiated by the master by sending a broadcast to all slaves at a specific address.
  - On reception of this message, all slaves latch the value of their internal clock twice, once when the message is received and once when it returns (remember EtherCAT has a ring topology).
  - The master thens read all latched values and calculate the delay for each slave.
- This process can be repeated as many times as required to reduce jitter and to average out values.
- Total delays are calculated for each slave depending on their position in the slave-ring and are uploaded to an offset register.
- Finally the master issues a broadcast read-write on the system clock, which makes the first slave the reference clock and forcing all other slaves to set their internal clock appropriately with the now known offset.

To keep the clocks synchronized after initialization, the master or slave must regularly send out the broadcast again to counter any effects of speed difference between the internal clocks of each slave.

- Each slave has to adjust the speed of their internal clock or implement an internal correction mechanism whenever they have to adjust.
- The system clock is specified as a 32-bit counter with a base unit of 1 ns starting at January 1st 2000, 0:00.

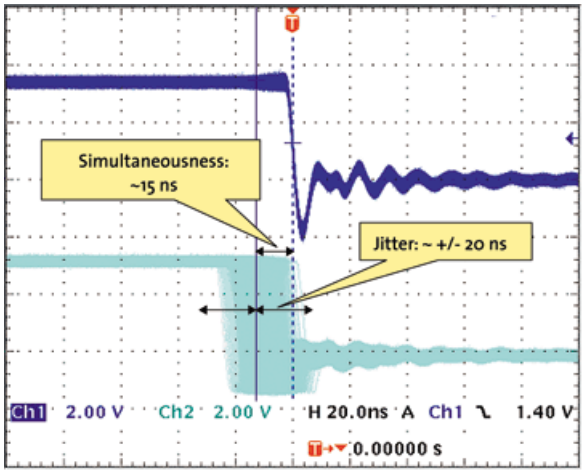


Figure 10-3: Synchronicity and Simultaneousness

Scope view of two distributed devices with 300 nodes and 120 m of cable between them.

7.6.3.5 File Access over EtherCAT (FoE)

This very simple protocol, similar to TFTP, enables access to any data structure in the device.

Standardized firmware upload to devices is possible, regardless of whether or not they support TCP/IP.

7.6.3.6 EtherCAT Gateways

Gateway devices are available for the integration of existing fieldbus components (e.g., CANopen, DeviceNet) into EtherCAT networks.

- Other Ethernet protocols can be used in conjunction with EtherCAT.
  - The Ethernet frames are tunneled via the EtherCAT protocol, which is the standard approach for Internet applications.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The EtherCAT network is fully transparent for the Ethernet device.
  - The real-time characteristics are not impaired since the master dictates exactly when the tunneled transfers are to occur and how much of the 100Mb/s media the tunneled protocols can use.
- Therefore, all Internet technologies can also be used in the EtherCAT environment.

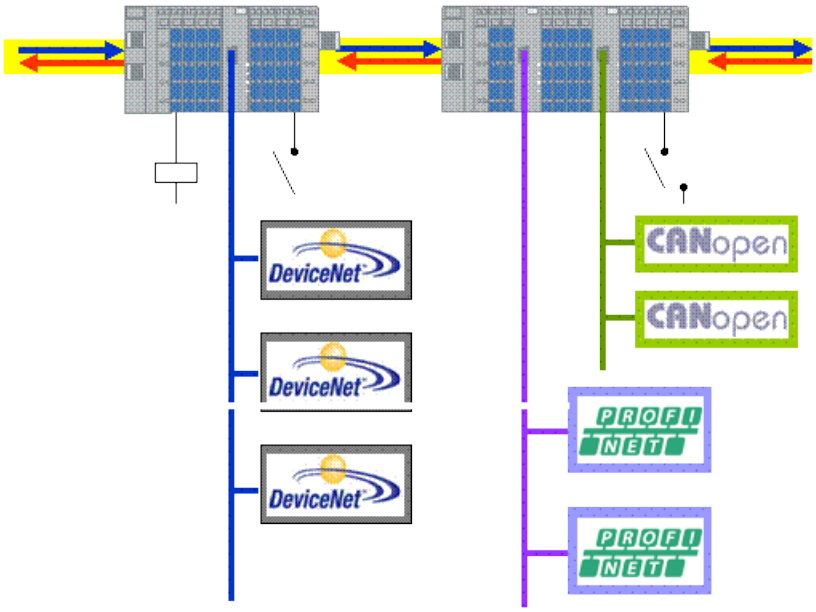


Figure 10-4: Fieldbus Gateway

7.6.3.7 EtherCAT Performance

Short cycle times can be achieved because the host microprocessors in the slave devices are not involved in the processing of the Ethernet packets to transfer the process images.

- All process data communication is handled by the slave controller hardware.
- Combined with these features, this makes EtherCAT a high-performance distributed I/O system:
  - Process data exchange with 1000 distributed digital I/O takes about 30 µs, which is typical for a transfer of 125 byte over 100Mb/s Ethernet. Data for and from 100 servo axes can be updated with up to 10 kHz.
- Typical network update rates are 1-30 kHz, but EtherCAT can be used with slower cycle times, too, if the DMA load is too high on your PC.

Process Data	Update Time
256 distributed digital I/O	11 µs = 0,01 ms
1000 distributed digital I/O	30 µs
200 analog I/O (16 bit)	50µs ↔ 20 kHz
100 Servo Axis, with 8 Bytes input and output data each	100 µs
1 Fieldbus Master-Gateway (1486 Bytes Input and 1486 Bytes Output Data)	150 µs

Figure 10-5: EtherCAT Performance Overview

The communication with 100 servo axes is also extremely fast: every 100µs, all axes are provided with command values and control data and report their actual position and status.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The "EtherCAT Distributed Clock (Synchronization)" (→ p. 418) technique enables the axes to be synchronized with a deviation of significantly less than 1 microsecond.
- Even at this pace, there is more than sufficient bandwidth for asynchronous communications such as TCP/IP, parameter download or diagnostic data upload.

7.6.3.8 EtherCAT Protocol

The EtherCAT protocol is optimized for process data and is transported directly within the standard IEEE 802.3 Ethernet frame using Ethertype 0x88a4.

- It can consist of several sub-datagrams, each serving a particular memory area of the logical process images that can be a maximum of four gigabytes in size.
- The data sequence is independent of the physical order of the nodes in the network and addressing can be in any order.
- Broadcast, multicast, and communication between slaves is possible and must be done by the master device.
- If IP routing is required, the EtherCAT protocol can be inserted into UDP/IP datagrams.
    - This enables any control with Ethernet protocol stack to address EtherCAT systems.

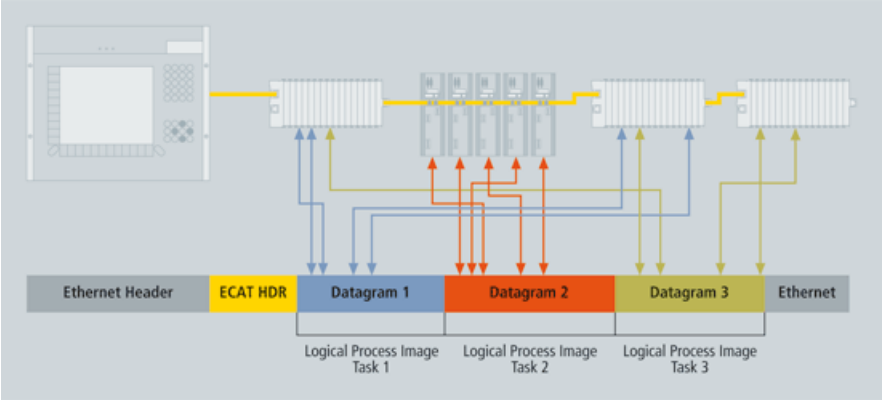


Figure 10-6: Process Data is Inserted in Telegrams

- KAS supports "CANopen over EtherCAT (CoE)" (→ p. 418).
    - It enables the advantages of EtherCAT in terms of transfer characteristics to be combined with proven, profile-specific drive functions.
- KAS uses "File Access over EtherCAT (FoE)" (→ p. 419) protocols to provide options for efficiently exchanging firmware via the bus.
    - See Figure: "**Several Device Profiles and Protocols can Coexist**".

7.6.3.9 Safety over EtherCAT

The protocol enhancement called Safety over EtherCAT (FcSoE) enables safety-related communication and control communication on the same network.

- The safety protocol is based on the application layer of EtherCAT, with no influence on the lower layers.
- It is certified according to IEC 61508 and meets the requirements of Safety Integrity Level (SIL) 3.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

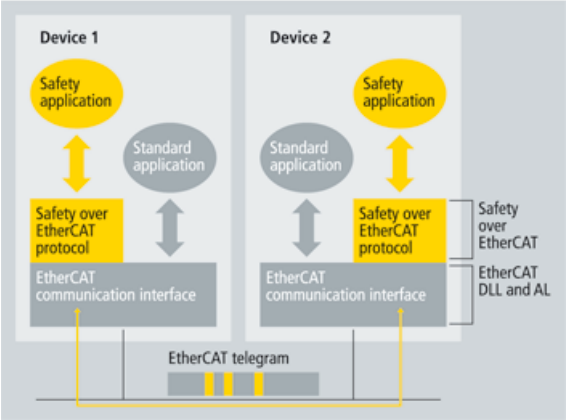


Figure 10-7: Safety over EtherCAT Software Architecture

7.6.3.10 EtherCAT Topology

Using full-duplex Ethernet physical layers, EtherCAT slave controllers close an open port automatically and return the Ethernet frame if no downstream device is detected.

- Slave devices can have several ports.
- Using these features, EtherCAT can support almost any physical topology (e.g., line, tree, or star).
    - The bus or line structure known from the fieldbuses is then available for Ethernet.

The combination of line and branches or stubs is also possible.

- Any EtherCAT device with three or more ports can act as a junction; no additional switches are required.
- The classic switch-based Ethernet star topology can be used either with switches configured to forward traffic directly between ports, or with special slave devices.
    - The switches are then located between the network master and the slave devices.

The special slave device assembly (i.e., standard slave devices don't have a MAC address) attached to one switch port together forms an EtherCATsegment. This is either addressed via its MAC address or via port-based VLANs.

- Since the 100BASE-TX Ethernet physical layer is used, the distance between any two nodes can be up to a maximum of 100 m (300 ft).
- Up to 65535 devices can be connected per segment.
- If an EtherCAT network is wired in ring configuration (requiring two ports on the master device), it can provide cable redundancy.

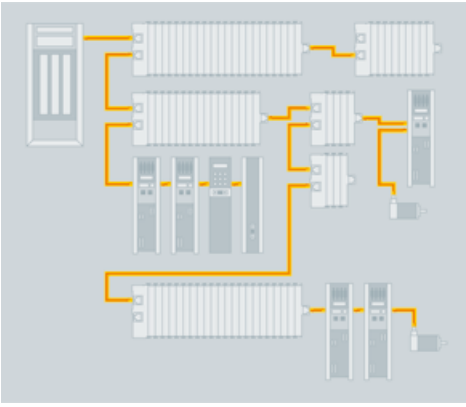


Figure 10-8: Flexible Topology: Line, Tree, or Star

- KAS controllers support line, tree, and star topologies.
- Ring topologies and cable redundancy is not supported because the KAS controllers have a singe EtherCAT port.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7.6.4 EtherCAT Implementation

EtherCAT was developed with very low cost devices in mind, like I/O terminals, sensors, and embedded controllers.

- EtherCAT only uses standard Ethernet frames according to IEEE 802.3.
  - These frames are sent by the master device, and the slave devices extract and/or insert data on the fly.
  - EtherCAT uses standard Ethernet MACs, where they really make sense: in the master device.
- EtherCAT slave controllers are used where such dedicated chips really make sense: in the slave device.
  - They handle the process data protocol in hardware and provide maximum real-time performance regardless of the local processing power or software quality.

7.6.4.1 EtherCAT Master Configuration

EtherCAT communicates a maximum of 1486 bytes of distributed process data with just one Ethernet frame.

- Unlike other solutions where the master device in each network cycle has to process, send, and receive frames for each node, EtherCAT systems typically only need one or two frames per cycle for the entire communication with all nodes.
  - EtherCAT masters do not require a dedicated communication processor.
  - The master functionality puts hardly any load on the host CPU, which can handle this task easily while processing the application program.
- EtherCAT can be implemented without special or expensive active plug-in cards, just by using a passive NIC card or the on-board Ethernet MAC.
  - Implementation of an EtherCAT master is very easy, particularly for small and medium-sized control systems and for clearly defined applications.

Example

- A PLC with a single process image: if it does not exceed the 1486 bytes, cyclic sending of a single Ethernet frame with the cycle time of the PLC is sufficient.
  - See Figure 10-9.
  - Because the header does not change at run-time, the only thing required is that a constant header be added to the process image and that the result be transferred to the Ethernet controller.
- The process image is already sorted, since with EtherCAT mapping does not occur in the master, but in the slaves - the peripheral devices insert their data at the respective points in the passing frame.
  - This further unburdens the host CPU.
- It was found that an EtherCAT master entirely implemented in software on the host CPU uses less of its processing power than much slower fieldbus systems implemented with active plug-in cards.
  - Servicing the DPRAM of the active card alone puts more load on the host.

System configuration tools provide the network and device parameters (including the corresponding boot-up sequence) in a standardized XML format.

See "Configure EtherCAT Motion Bus" (➔ p. 164).

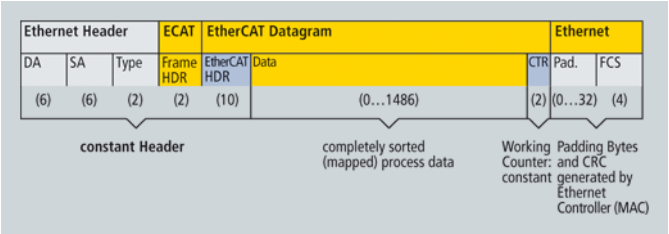


Figure 10-9: Master-Implementation with One Process Image

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

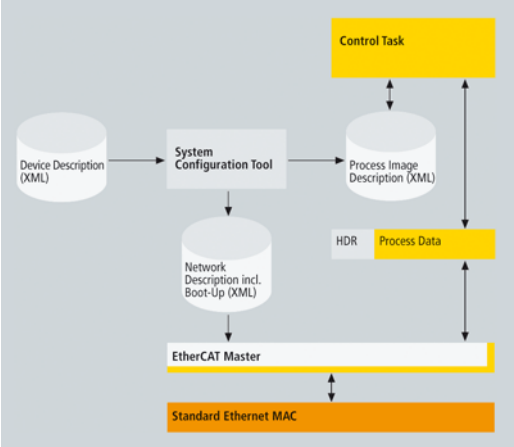


Figure 10-10: Structure of EtherCAT Master Implementation

7.6.4.1.1 ENI and ESI Files

The EtherCAT master uses the information from the ENI file to initialize and configure the EtherCAT network.

- The ESI files are provided by the vendor for each device.
  - They contain information about the device functionality and its settings.
- The ESI files are used by the KAS-IDE to generate the ENI file.
- The KAS controller's EtherCAT master uses the ENI file for network initialization and configuration.

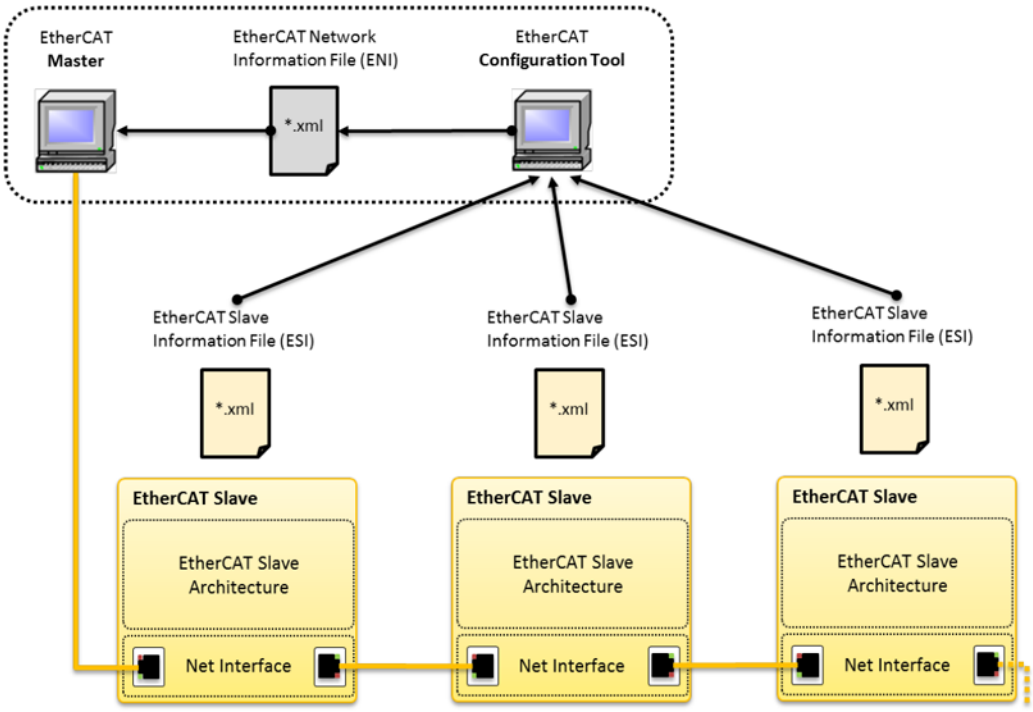


Figure 10-11: EtherCAT Network Architecture [1]

[1] Image courtesy of EtherCAT.org, http://www.ethercat.org/pdf/english/etg2200_v2i0i1_slaveimplementationguide.pdf

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

ENI File	A network configuration file in XML format. • The ENI file describes the network topology, the initialization commands for each device, and commands which have to be sent cyclically. • The ENI file is provided to the master, which sends commands according to this file. • The KAS-IDE creates the ENI file after a network discovery, which can be exported or imported. • A scan and compile should be redone, if the network changes, in order to regenerate the ENI file.
ESI File	A device description in XML format. • This is a fixed file provided by the supplier of a given EtherCAT device. • The ESI file contains information about the device's functionality and settings. • EtherCAT device vendors must provide an ESI file, which is used by the KAS-IDE to compile the network information (e.g., process data structures, initialization commands) and create the ENI file.

NOTE

KAS uses ENI schema version 1.3 (May 20, 2009).
See the ETG.2100 specification at www.ethercat.org.

7.6.4.2 EtherCAT Slave Configuration

A cost-effective EtherCAT slave controller (ESC) is used in the slave devices.

- With EtherCAT, the slave does not need a microcontroller.
  - Simple devices with an I/O interface can be implemented only with the ESC and the RJ45 connector.
  - The process data interface (PDI) to the slave application is a 32-bit I/O interface.
    - This slave without configurable parameters needs no software or mailbox protocol.
- The EtherCAT State Machine is handled in the ESC.
  - The boot-up information for the ESC comes out of the EEPROM that also supports the identity information of the slave.
  - More complex slaves that are configurable have a host CPU on board.
    - This CPU is connected to the ESC with an 8-bit or 16-bit parallel interface or via a serial connection.

7.6.4.2.1 EtherCAT Slave Controller

The slave controllers typically feature an internal DPRAM and offer a range of interfaces for accessing this application memory:

- The SPI (serial peripheral interface bus) is intended for devices with small process data quantity (e.g., analog I/O modules, sensors, encoders or simple drives).
- The parallel 8/16-bit microcontroller interface corresponds to conventional interfaces for fieldbus controllers with DPRAM interface.
  - It is particularly suitable for more complex devices with larger data volume.
- The 32-bit parallel I/O interface is suitable for the connection of up to 32 digital inputs/outputs, but also for simple sensors or actuators operating with 32 data bits.
  - Such devices do not need a host CPU at all (see Figure 10-13).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

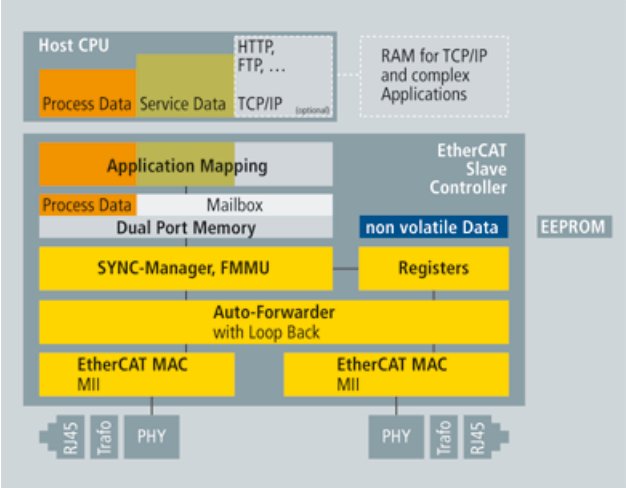


Figure 10-12: Slave Hardware: FPGA with Host CPU

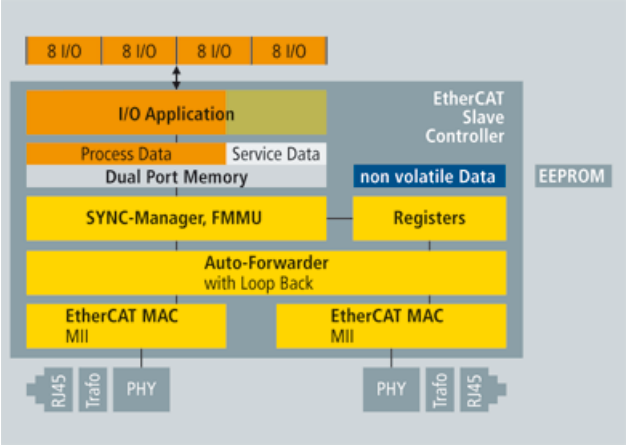


Figure 10-13: Slave Hardware: FPGA with direct I/O

7.6.4.3 EtherCAT State Machine

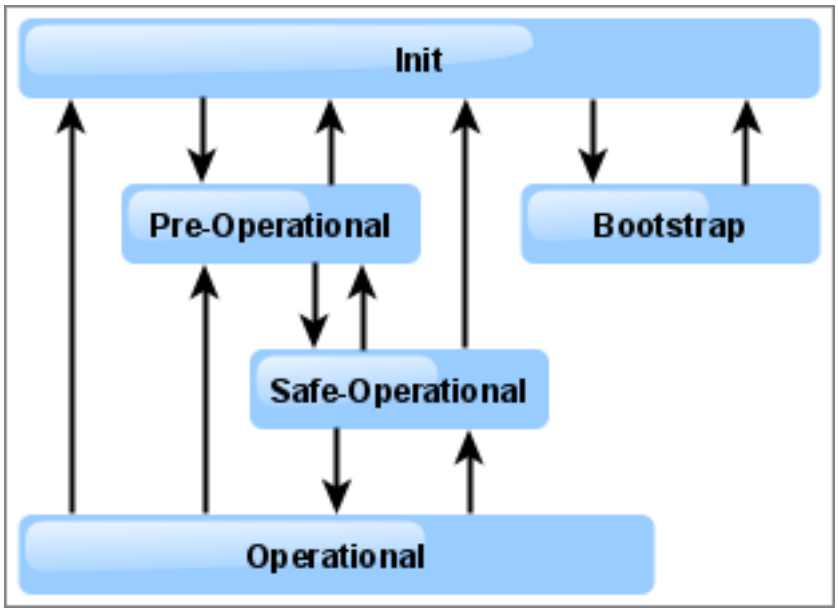


Figure 10-14: EtherCAT State Machine

Where the transitions are from:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- **Init** to **Pre-Operational (Pre-Op)**: Master configures the Sync Manager channels for Mailbox communication.
- **Pre-Op** to **Safe-Operational (Safe-Op)**: Master configures parameter using the Mailbox.
- **Safe-Op** to **Operational (Op)**: Master sends valid Outputs.

7.6.4.3.1 Different States

State	Description
Bootstrap	• Recommended if firmware updates are necessary. • No Process Data communication. • Communication via Mailbox on Application Layer. • Only the "File Access over EtherCAT (FoE)" (→ p. 419) protocol is available.
Init	• No communication on the Application Layer. • Master has access to the DL-Information registers.
Operational (Op)	Inputs and Outputs are valid.
Pre-Operational (Pre-Op)	• Mailbox communication on the Application Layer. • No Process Data communication.
Safe-Operational (Safe-Op)	• Mailbox communication on the Application Layer. • Process Data communication. • Only Inputs are evaluated (Outputs in **Safe** state).

7.6.4.4 PDOs for AKD, AKD2G, AKD-N, Kollmorgen Essentials Drive, S300, or S700

The KAS Motion Engine interacts with the AKD, AKD2G, AKD-N, S300, or S700 drives through CANopen objects in the selected PDOs.

Some of the ML and MC function blocks require specific CANopen objects in the PDOs.

- Defaults:
  - AKD, AKD2G, AKD-N, and KED PDOs include all of the CANopen objects needed by the motion engine and function blocks.
  - S300 and S700 PDOs include only the minimal CANopen objects required by the KAS motion engine.

7.6.4.4.1 PDO Types

- **RxPDO** – "From Controller to Drive (RxPDO)" (→ p. 428)
  - Drive and Objects
    - "RxPDO - AKD Drive CANopen Objects" (→ p. 429)
    - "RxPDO - AKD2G Drive CANopen Objects" (→ p. 428)
    - "RxPDO - AKD-N Drive CANopen Objects" (→ p. 430)
    - "RxPDO - KED Drive CANopen Objects" (→ p. 431)
    - "RxPDO - S300 and S700 Drive CANopen Objects" (→ p. 432)
- **TxPDO** – "From Drive to Controller (TxPDO)" (→ p. 433)
  - Drives and Objects
    - "TxPDO - AKD Drive CANopen Objects" (→ p. 435)
    - "TxPDO - AKD2G Drive CANopen Objects" (→ p. 433)
    - "TxPDO - AKD-N Drive CANopen Objects" (→ p. 437)
    - "TxPDO - KED Drive CANopen Objects" (→ p. 438)
    - "TxPDO - S300 and S700 Drive CANopen Objects" (→ p. 439)

7.6.4.4.2 Examples

These are examples where the PDO object is passed as an argument in the function block.

7.6.4.4.3.1 Example 1

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
MLSmpConECAT( PipeNetwork.SMP1, 1001, 16#2050, 0);
```

Where:

- `PipeNetwork.SMP1` is a Sampler block ID.
- `1001` is the EtherCAT address for the first device.
- `16#2050` (0x2050) is the object index and `0` is the subindex for the AKD's "Position actual value 2" object.

7.6.4.4.4.2 Example 2

```
MLCNVConnectEx(PipeNetwork.CNV1, PipeNetwork.AXIS1, EC_ADDITIVE_TORQUE_VALUE,
0 ) ;
```

The argument is a constant based on the object index.

7.6.4.4.5.3 Example 3

```
MLSmpConPNAxis(PipeNetwork.SMP1, PipeNetwork.AXIS1, ML_SECOND_FEEDBACK_
POSITION);
```

Where:

ML_SECOND_FEEDBACK_POSITION specifies the secondary actual position from the Pipe Network Axis.

NOTE

The ML_SECOND_FEEDBACL_POSITION constant has been deprecated and only AKD devices supported it. It is deprecated because the AKD2G supports more than one secondary feedback.

7.6.4.4.6 From Controller to Drive (RxPDO)

These Controller to Drive objects are available in the default PDO.

- "RxPDO - AKD Drive CANopen Objects" (→ p. 429)
- "RxPDO - AKD2G Drive CANopen Objects" (→ p. 428)
- "RxPDO - AKD-N Drive CANopen Objects" (→ p. 430)
- "RxPDO - KED Drive CANopen Objects" (→ p. 431)
- "RxPDO - S300 and S700 Drive CANopen Objects" (→ p. 432)

7.6.4.4.7.1 RxPDO - AKD2G Drive CANopen Objects

This table identifies:

- The CANopen objects required by the motion engine and function blocks.
- Whether the objects are available in the default PDO.
- Their corresponding drive parameter.

IMPORTANT

Review the table to verify the objects needed by the features the application uses are included in the PDO selection.

NOTE

Axis 1 objects begin with 0x60.
Axis 2 objects begin with 0x68.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

RxPDO - AKD2G Drive CANopen Objects					
Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated Drive Parameter
0x3601:1	Physical Outputs Used by Onboard I/O mappings	No	N/A	N/A	DOUT.STATES
0x3800:2	AOUT1.Fieldbus Value Used by Onboard I/O mappings	No	N/A	N/A	AOUT.VALUE
0x3801:2	AOUT2.Fieldbus Value Used by Onboard I/O mappings	No	N/A	N/A	AOUT.VALUE
0x6040:0 or 0x6840:0	CANopen Control Word	Yes	Needed for motion engine axis to work properly.	• Needed for motion engine axis to work properly: • MC_ ClearFaults • MC_Power	
0x6070A:0 or 0x6870:0	Target Position Axis 1 / 2	Yes	• Needed for motion engine axis to work properly: • Related to the **Axis** MLAxisCmdPos pipe block positions. • See "Associated Data on Positions" (→ p. 333).	• Needed for motion engine axis to work properly: • MC_ ReadParam • ID-1 in "Non-Boolean Parameters" (→ p. 355)	
0x60B8:0 or 0x68B8:0	Touchprobe Function Axis 1 / 2	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ MachRegist • MC_MarkRegist • MC_Reference • MC_StopRegist • MC_ TouchProbe	

7.6.4.4.8.2 RxPDO - AKD Drive CANopen Objects

This table identifies:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The CANopen objects required by the motion engine and function blocks.
- Whether the objects are available in the default PDO.
- Their corresponding drive parameter.

IMPORTANT

Review the table to verify the objects needed by the features the application uses are included in the PDO selection.

RxPDO - AKD Drive CANopen Objects

Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated Drive Parameter
0x20A4:0 or 0x2802:0	Latch control word	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ TouchProbe	• CAP0.EN • CAP1.EN • CAP0.MODE • CAP1.MODE
0x3470:3	AOUT.VALUE Used by Onboard I/O mappings	No	N/A	N/A	AOUT.VALUEU
0x6040:0	CANopen Control Word	Yes		• MC_ ClearFaults • MC_Power	
0x60B2:0 or 0x60F6:1	Additive torque value Torque Feed Forward	No	MLAxisAddTq	N/A	IL.BUSFF
0x60C1:1 or 0x6062:0	Position demand value	Yes	• Related to the **Axis** MLAxisCmdPos pipe block positions. • See "Associated Data on Positions" (➔ p. 333).	• MC_ ReadParam • ID-1 in "Non-Boolean Parameters" (➔ p. 355)	PL.CMD
0x60FE:1	Digital Outputs Used by Onboard I/O mappings	No	N/A	N/A	DOUT.STATES

7.6.4.4.9.3 RxPDO - AKD-N Drive CANopen Objects

This table identifies:

- The CANopen objects required by the motion engine and function blocks.
- Whether the objects are available in the default PDO.
- Their corresponding drive parameter.

IMPORTANT

Review the table to verify the objects needed by the features the application uses are included in the PDO selection.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

RxPDO - AKD-N Drive CANopen Objects					
Index - subindex	**Object Name**	**Required**	**Associated ML FB**	**Associated MC FB**	**Associated Drive Parameter**
0x20A4:0 or 0x2802:0	Latch control word	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ TouchProbe	• CAP0.EN • CAP1.EN • CAP0.MODE • CAP1.MODE
0x6040:0	CANopen Control Word	Yes		• MC_ ClearFaults • MC_Power	
0x60B2:0 or 0x60F6:1	Additive torque value Torque Feed Forward	No	MLAxisAddTq	N/A	IL.BUSFF
0x60C1:1 or 0x6062:0	Position demand value	Yes	• Related to the **Axis** MLAxisCmdPos pipe block positions. • See "Associated Data on Positions" (→ p. 333).	• MC_ ReadParam • ID-1 in "Non-Boolean Parameters" (→ p. 355)	PL.CMD
0x60FE:1	Digital outputs Used by Onboard I/O mappings	No	N/A	N/A	DOUT.STATES

7.6.4.4.10.4 RxPDO - KED Drive CANopen Objects

This table identifies:

- The CANopen objects required by the motion engine and function blocks.
- Whether the objects are available in the default PDO.
- Their corresponding drive parameter.

IMPORTANT

Review the table to verify the objects needed by the features the application uses are included in the PDO selection.

RxPDO - KED Drive CANopen Objects					
Index - subindex	**Object Name**	**Required**	**Associated ML FB**	**Associated MC FB**	**Associated Drive Parameter**
0x3601:1	Physical Outputs Used by Onboard I/O mappings	No	N/A	N/A	DOUT.STATES

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

RxPDO - KED Drive CANopen Objects					
Index - subindex	**Object Name**	**Required**	**Associated ML FB**	**Associated MC FB**	**Associated Drive Parameter**
0x6040:0	"CANopen Control Word" (→ p. 443)	Yes	Needed for motion engine axis to work properly	• Needed for motion engine axis to work properly • MC_ ClearFaults • MC_Power	
0x6070A:0	Target Position	Yes	• Needed for motion engine axis to work properly: • Related to the **Axis** MLAxisCmdPos pipe block positions. • See "Associated Data on Positions" (→ p. 333).	• Needed for motion engine axis to work properly: • MC_ ReadParam • ID-1 in "Non-Boolean Parameters" (→ p. 355)	
0x60B8:0	Touchprobe Function	Yes	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_AbortTrigger • MC_MachRegist • MC_MarkRegist • MC_Reference • MC_StopRegist • MC_TouchProbe	

7.6.4.4.11.5 RxPDO - S300 and S700 Drive CANopen Objects

This table identifies:

- The CANopen objects required by the motion engine and function blocks.
- Whether the objects are available in the default PDO.
- Their corresponding drive parameter.

IMPORTANT

Review the table to verify the objects needed by the features the application uses are included in the PDO selection.

RxPDO - S300 and S700 Drive CANopen Objects					
Index - subindex	**Object Name**	**Required**	**Associated ML FB**	**Associated MC FB**	**Associated Drive Parameter**
0x20A4:0 or 0x2802:0	Latch control word	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ TouchProbe	• CAP0.EN • CAP1.EN • CAP0.MODE • CAP1.MODE
0x6040:0	CANopen Control Word	Yes		• MC_ ClearFaults • MC_Power	

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

RxPDO - S300 and S700 Drive CANopen Objects					
Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated Drive Parameter
0x60B2:0 or 0x60F6:1	Additive torque value Torque Feed Forward	No	MLAxisAddTq	N/A	IL.BUSFF
0x60C1:1 or 0x6062:0	Position demand value	Yes	• Related to the **Axis** MLAxisCmdPos pipe block positions. • See "Associated Data on Positions" (→ p. 333).	• MC_ReadParam • ID-1 in "Non-Boolean Parameters" (→ p. 355)	PL.CMD

7.6.4.4.12 From Drive to Controller (TxPDO)

These Drive to Controller objects are available in the default PDO.

- "TxPDO - AKD Drive CANopen Objects" (→ p. 435)
- "TxPDO - AKD2G Drive CANopen Objects" (→ p. 433)
- "TxPDO - AKD-N Drive CANopen Objects" (→ p. 437)
- "TxPDO - KED Drive CANopen Objects" (→ p. 438)
- "TxPDO - S300 and S700 Drive CANopen Objects" (→ p. 439)

7.6.4.4.13.1 TxPDO - AKD2G Drive CANopen Objects

This table identifies:

- The CANopen objects required by the motion engine and function blocks.
- Whether the objects are available in the default PDO.
- Their corresponding drive parameter.

IMPORTANT

Review the table to verify the objects needed by the features the application uses are included in the PDO selection.

NOTE

Axis 1 objects begin with 0x60.
Axis 2 objects begin with 0x68.

TxPDO - AKD2G Drive CANopen Objects					
Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated Drive Parameter
0x3780:1	AIN1.VALUE	No	N/A	N/A	
0x3781:1	AIN2.VALUE	No	N/A	N/A	
0x6041:0 or 0x6841:0	Status word Axis 1 / 2	Yes	Needed for the servo axis mapped to the drive to work properly.	Needed for the servo axis mapped to the drive to work properly.	

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TxPDO - AKD2G Drive CANopen Objects					
Index - subindex	**Object Name**	**Required**	**Associated ML FB**	**Associated MC FB**	**Associated Drive Parameter**
0x6064:0 or 0x6864:0	Position Actual Value Axis 1 / 2	Yes	Needed for the servo axis mapped to the drive to work properly.	Needed for the servo axis mapped to the drive to work properly.	
0x6077:0 or 0x6877:0	Torque actual value Axis 1 / 2	No	MLAxisReadTq	MC_ReadParam	
0x60B9:0 or 0x68B9:0	Touch Probe Status Axis 1 / 2	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ MachRegist • MC_ MarkRegist • MC_ Reference • MC_ StopRegist • MC_ TouchProbe	
0x60D1:0 or 0x68D1:0	Touch Probe 1 Time Stamp positive value Axis 1 / 2	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ MachRegist • MC_ MarkRegist • MC_ Reference • MC_ StopRegist • MC_ TouchProbe	
0x60D4:0 or 0x68D4:0	Touch Probe 2 Time Stamp positive value Axis 1 / 2	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ MachRegist • MC_ MarkRegist • MC_ Reference • MC_ StopRegist • MC_ TouchProbe	
0x60E4:3	3rd additional Position actual value Axis 1	No	MLSmpConECAT	For a digitizing axis	

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TxPDO - AKD2G Drive CANopen Objects					
Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated Drive Parameter
0x60E4:4	4th additional Position actual value Axis 1		MLSmpConECAT	For a digitizing axis	
0x60E4:5	5th additional Position actual value Axis 1		MLSmpConECAT	For a digitizing axis	
0x60F4:0 or 0x68F4:0	Following error Actual value: Axis 1 / 2	No	MLAxisReadFEUU	MC_ReadParam	

7.6.4.4.14.2 TxPDO - AKD Drive CANopen Objects

This table identifies:

- The CANopen objects required by the motion engine and function blocks.
- Whether the objects are available in the default PDO.
- Their corresponding drive parameter.

IMPORTANT

Review the table to verify the objects needed by the features the application uses are included in the PDO selection.

TxPDO - AKD Drive CANopen Objects					
Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated Drive Parameter
0x6041:0	CANopen Control Word	Yes	N/A	N/A	N/A
0x6063:0 or 0x6064:0	Position Actual Value	No	• MLAxisFBackPos • MLAxisReadActPos	MC_ ReadActPos	PL.FB

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TxPDO - AKD Drive CANopen Objects					
Index - subindex	**Object Name**	**Required**	**Associated ML FB**	**Associated MC FB**	**Associated Drive Parameter**
0x2050:0 or 0x35C9:0	Position Actual Value 2	No	MLAxisRead2ndFB **NOTE** This FB is deprecated. The recommended method for accessing feedback 1-5 on an AKD2G, or a third-party device's position feedback, is to use the Sampler block and MLSmpConECAT.	For a **Digitizing** axis: • Secondary feedback can be read by reading the actual position of the axis which is assigned to the secondary feedback. • Digitizing axes always use the second feedback for the Drive. • KAS does not allow a digitizing axis on a drive which has not had a servo axis already assigned.	PL.FB (If DRV.CMDSOURCE = 1)
0x606C:0	Velocity actual value	No	MLAxisReadVel	MC_ReadActVel	VL.FB
0x6077:0	Torque actual value	No	MLAxisReadTq	MC_ReadParam	IL.FB
0x20A5:0 or 0x2901:0	Latch status word	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ TouchProbe	CAPx.STATE
0x20A6:0 or 0x2902:0	Latch position	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ TouchProbe	• CAPx.T - for time • CAPx.PLFB - for position
0x60FD:0	Digital Inputs Used by Onboard I/O mappings	No	N/A	N/A	DIN.STATES

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TxPDO - AKD Drive CANopen Objects					
Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated Drive Parameter
0x3470:4	AIN.VALUE Used by Onboard I/O mappings	No	N/A	N/A	AIN.VALUE
0x60F4	Following error	No	MLAxisReadFEUU	MC_ReadParam	PL.ERR

7.6.4.4.15.3 TxPDO - AKD-N Drive CANopen Objects

This table identifies:

- The CANopen objects required by the motion engine and function blocks.
- Whether the objects are available in the default PDO.
- Their corresponding drive parameter.

IMPORTANT

Review the table to verify the objects needed by the features the application uses are included in the PDO selection.

TxPDO - AKD-N Drive CANopen Objects					
Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated Drive Parameter
0x6041:0	CANopen Control Word	Yes	N/A	N/A	N/A
0x6063:0 or 0x6064:0	Position Actual Value	No	• MLAxisFBackPos • MLAxisReadActPos	MC_ReadActPos	PL.FB
0x606C:0	Velocity actual value	No	MLAxisReadVel	MC_ReadActVel	VL.FB
0x6077:0	Torque actual value	No	MLAxisReadTq	MC_ReadParam	IL.FB
0x20A5:0 or 0x2901:0	Latch status word	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ TouchProbe	CAPx.STATE
0x20A6:0 or 0x2902:0	Latch position	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ TouchProbe	• CAPx.T - for time • CAPx.PLFB - for position
0x60FD:0	Digital inputs Used by Onboard I/O mappings	No	N/A	N/A	DIN.STATES

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TxPDO - AKD-N Drive CANopen Objects					
Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated Drive Parameter
0x60F4	Following error	No	MLAxisReadFEUU	MC_ReadParam	PL.ERR

7.6.4.4.16.4 TxPDO - KED Drive CANopen Objects

This table identifies:

- The CANopen objects required by the motion engine and function blocks.
- Whether the objects are available in the default PDO.
- Their corresponding drive parameter.

IMPORTANT

Review the table to verify the objects needed by the features the application uses are included in the PDO selection.

TxPDO - KED Drive CANopen Objects					
Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated Drive Parameter
0x6041:0	Status word Axis	Yes	Needed for the servo axis mapped to the drive to work properly.	Needed for the servo axis mapped to the drive to work properly.	
0x6064:0	Position Actual Value Axis	Yes	Needed for the servo axis mapped to the drive to work properly.	Needed for the servo axis mapped to the drive to work properly.	
0x6077:0	Torque actual value Axis	No	MLAxisReadTq	MC_ReadParam	
0x60B9:0	Touch Probe Status	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ MachRegist • MC_ MarkRegist • MC_Reference • MC_StopRegist • MC_ TouchProbe	
0x60D1:0	Touch Probe 1 Time Stamp Positive Value	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ MachRegist • MC_ MarkRegist • MC_Reference • MC_StopRegist • MC_ TouchProbe	

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TxPDO - KED Drive CANopen Objects					
Index - subindex	**Object Name**	**Required**	**Associated ML FB**	**Associated MC FB**	**Associated Drive Parameter**
0x60D4:0	Touch Probe 2 Time Stamp Positive Value	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ MachRegist • MC_ MarkRegist • MC_Reference • MC_StopRegist • MC_ TouchProbe	
0x60E4:3	3rd Additional Position Actual Value	No	MLSmpConECAT	For a digitizing axis	
0x60E4:4	4th Additional Position Actual Value	No	MLSmpConECAT	For a digitizing axis	
0x60E4:5	5th Additional Position Actual Value	No	MLSmpConECAT	For a digitizing axis	
0x60F4:0	Following error Actual value	No	MLAxisReadFEUU	MC_ReadParam	

7.6.4.4.17.5 TxPDO - S300 and S700 Drive CANopen Objects

This table identifies:

- The CANopen objects required by the motion engine and function blocks.
- Whether the objects are available in the default PDO.
- Their corresponding drive parameter.

IMPORTANT

Review the table to verify the objects needed by the features the application uses are included in the PDO selection.

TxPDO - S300 and S700 Drive CANopen Objects					
Index - subindex	**Object Name**	**Required**	**Associated ML FB**	**Associated MC FB**	**Associated Drive Parameter**
0x6041:0	CANopen Control Word	Yes	N/A	N/A	N/A

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TxPDO - S300 and S700 Drive CANopen Objects					
Index - subindex	**Object Name**	**Required**	**Associated ML FB**	**Associated MC FB**	**Associated Drive Parameter**
0x6063:0 or 0x6064:0	Position Actual Value	No	• MLAxisFBackPos • MLAxisReadActPos	MC_ReadActPos	PL.FB
0x2050:0 or 0x35C9:0	Position Actual Value 2	No	MLAxisRead2ndFB **NOTE** This FB is deprecated. The recommended method for accessing feedback 1-5 on an AKD2G, or a third-party device's position feedback, is to use the Sampler block and MLSmpConECAT.	For a **Digitizing** axis: • Secondary feedback can be read by reading the actual position of the axis which is assigned to the secondary feedback. • Digitizing axes always use the second feedback for the Drive. • KAS does not allow a digitizing axis on a drive which has not a servo axis already assigned.	PL.FB (If DRV.CMDSOURCE = 1)
0x6077:0	Torque actual value	No	MLAxisReadTq	MC_ReadParam	IL.FB
0x20A5:0 or 0x2901:0	Latch status word	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ TouchProbe	CAPx.STATE
0x20A6:0 or 0x2902:0	Latch position	No	• MLAxisCfgFastIn • MLAxisTimeStamp • All Trigger MLTrig FB	• MC_ AbortTrigger • MC_ TouchProbe	CAPx.T - for time CAPx.PLFB - for position
0x60F4	Following error	No	MLAxisReadFEUU	MC_ReadParam	PL.ERR

7.6.5 CANopen

7.6.5.1 CANopenStatus Machine

The states of the status machine are revealed by using the "CANopenStatus Word" (→ p. 444).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

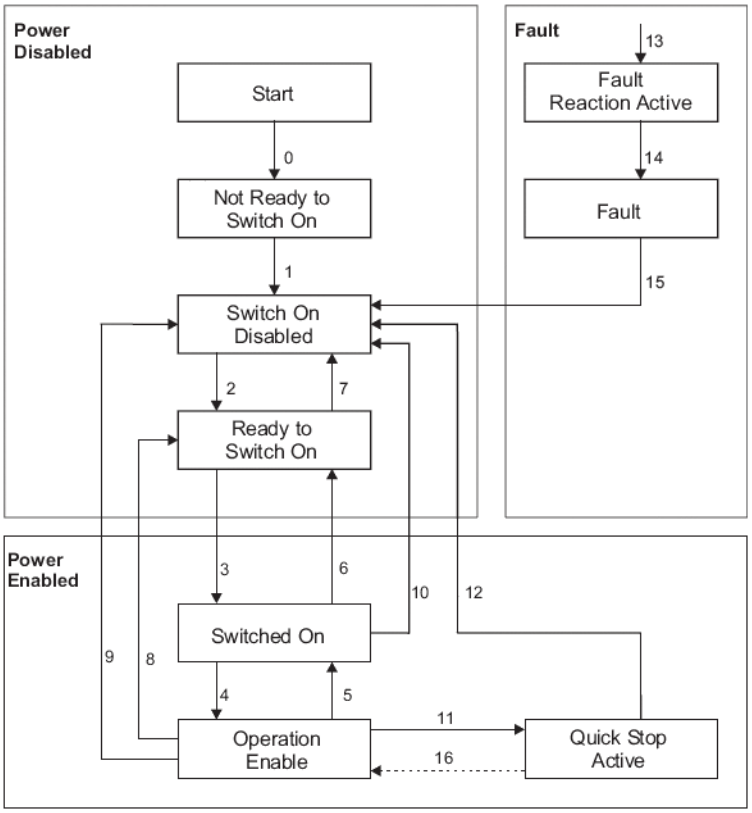


Figure 10-15: CANopen Status Machine

The start state is a pseudo-state indicating the start when the state machine is activated during the start-up sequence of the device drive's application software.

7.6.5.1.1 Status Description

Status	Description
Fault	• A fault is active. • The drive has been stopped and disabled.
Fault reaction active	• A fault has occurred. • The drive is stopped with the emergency stop ramp.
Not ready to switch on	The drive is not ready to switch on. • The controller has not indicated readiness for service. • The drive is still in the boot phase or in the fault status
Operation enabled	The drive is enabled and the setpoints are transferred from the EtherCAT interface. • No fault present. • Output stage is enabled. • Motion functions are enabled.
Quick stop active	• The drive has been stopped with the quick stop ramp. • Output stage is enabled. • Motion functions are not enabled.
Ready to switch on	The drive can be enabled via the control word. • DC-link voltage can be switched on. • Parameters can be transferred • Motion functions cannot be performed yet.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Status	Description
Switch on disabled	The drive cannot be enabled via the EtherCAT interface. Example: There is no connection to a power source.
Switched on	The drive is enabled but the setpoints are not yet transferred from the EtherCAT interface. • The drive is idle. • DC-link voltage must be switched on. • Parameters can be transferred but motion functions cannot be performed yet. • Output stage is switched on (enabled). • Operation Enable • No fault present; output stage is enabled; motion functions are enabled.

7.6.5.1.2 Transitions of the Status Machine

The drive device supports these transitions and actions.

- The event initiates the transition.
- The transition is terminated after the action has been performed.

7.6.5.1.3.1 Transition Events and Actions

Transition	Event	Action
0	Automatic transition after power-on or reset application.	Drive device self-test and/or self initialization has to be performed.
1	Automatic transition	Communication has to be activated.
2	Shutdown command from control device or local signal.	None
3	Switch on command received from control device or local signal.	The high-level power has to be switched ON, if possible.
4	Enable operation command received from control device or local signal.	• The drive function has to be enabled. • All internal setpoints cleared.
5	Disable operation command received from control device or local signal.	The drive function has to be disabled.
6	Shutdown command received from control device or local signal.	The high-level power has to be switched OFF, if possible.
7	Quick stop or disable voltage command from control device or local signal.	None
8	Shutdown command from control device or local signal.	• The drive function has to be disabled. • The high-level power has to be switched OFF, if possible.
9	Disable voltage command from control device or local signal.	• The drive function has to be disabled. • The high-level power has to be switched OFF, if possible.
10	Disable voltage or quick stop command from control device or local signal.	The high-level power has to be switched OFF, if possible.
11	Quick stop command from control device or local signal.	The quick stop function has to be started.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Transition	Event	Action
12	Either: • Automatic transition when the quick stop function is completed and quick stop option code is 1, 2, 3 or 4. • Disable voltage command received from control device (depends on the quick stop option code).	• The drive function has to be disabled. • The high-level power has to be switched OFF, if possible.
13	Fault signal	The configured fault reaction function has to be executed.
14	Automatic transition	• The drive function has to be disabled. • The high-level power has to be switched OFF, if possible.
15	Fault reset command from control device or local signal.	A reset of the fault condition is performed if no fault exists currently on the drive device. After leaving the Fault state, the Fault reset bit in the control word has to be cleared by the control device.
16	If the quick stop option code is 5, 6, 7, or 8, enable operation command from control device	The drive function has to be enabled.

7.6.5.2 CANopen Control Word

The status machine for the control word corresponds to the CANopen status machine.

- The control word indicates the received command controlling the state machine.
  - It is only read during **Operational** status.
- The control commands allow the manipulation of the state of a drive by setting its control word.
- Commands are built up from the logical combination of the bits in the control word and external signals (e.g., enable output stage).

7.6.5.2.1 Bit Assignment in Control Word

Bit	Name
0	Switch On
1	Disable Voltage
2	Quick Stop
3	Enable Operation
4	Operation-mode Specific
5	Operation-mode Specific
6	Operation-mode Specific
7	Reset Fault (only effective for faults)
8	Pause/halt
9	Reserved
10	Reserved

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Bit	Name
11	Reserved
12	Reserved
13	Manufacturer-specific
14	Manufacturer-specific
15	Manufacturer-specific

7.6.5.2.2 Coded Commands

Command	Bits of the Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2,6,8
Switch On	0	0	1	1	1	3
Switch On + Enable Operation	0	1	1	1	1	3 + 4 See Note.
Disable Voltage	0	X	X	0	X	7,9,10,12
Quick Stop	0	X	0	1	X	7,10,11
Disable Operation	0	0	1	1	1	5
Enable Operation	0	1	1	1	1	4,16
Fault Reset	Up	X	X	X	X	15

NOTE

Automatic transition to Enable operation state after executing SWITCHED ON state functionality.
Bits marked by an X are irrelevant.

7.6.5.3 CANopenStatus Word

- The status machine for the control word corresponds to the CANopen status machine.
- The status word is only updated and written by the drive in **Safe-Op** and **Operational** states.

7.6.5.3.1 Bit Assignment in Status Word

Bit	Name
0	Ready to switch on
1	Switched on
2	Operation Enable
3	Fault
4	Voltage Enabled
5	Quick Stop
6	Switch On Disabled
7	Warning
8	Manufacturer-specific (reserved)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Bit	Name
9	Remote (always 1)
10	Target Reached
11	Internal Limit Active
12	Operation-mode Specific (reserved)
13	Operation-mode Specific (reserved)
14	Manufacturer-specific (reserved)
15	Manufacturer-specific (reserved)

7.6.5.3.2 State Coding

These are the bit combinations coding the states.

Status Word MSB (15..12) (11 .. 8) (7 .. 4) (3 .. 0) LSB	State
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

NOTE

Bits marked by an X are irrelevant.

7.7 AKD Drive

- The **servo loops** in a KAS system are located in the AKD Drive.
- The **profile generator** used for all the motion in the application is located in the PCMM2G controller.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

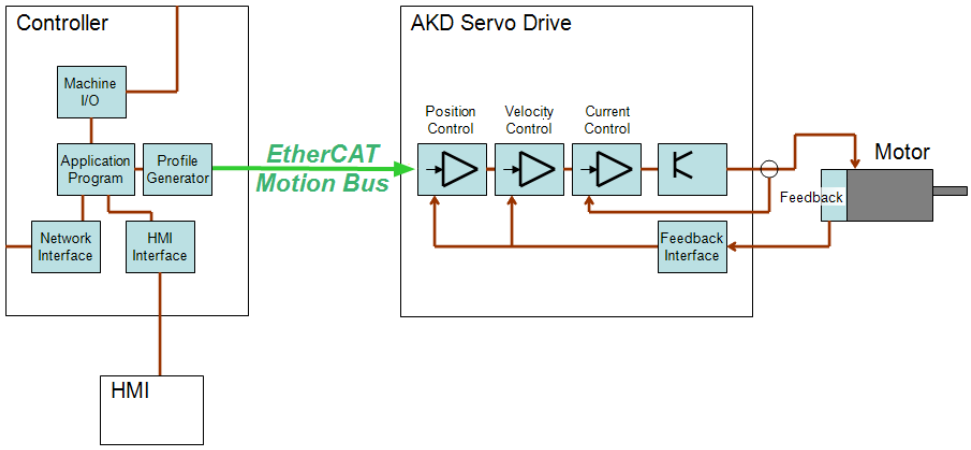


Figure 10-16: AKD into KAS

7.7.1 AKD Drive Configuration

7.7.1.1 Connection Modes

The AKD drive parameters can be configured when the KAS-IDE is connected to the controller and operating in one of these functional modes:

- "Online Configuration Mode" (→ p. 446)
- "Project Running on the Controller" (→ p. 446)

7.7.1.1.1 Disconnected

If the controller is not operating in one of these modes, then the AKD configurations are not accessible.

The Configuration tab indicates the AKD is Disconnected.

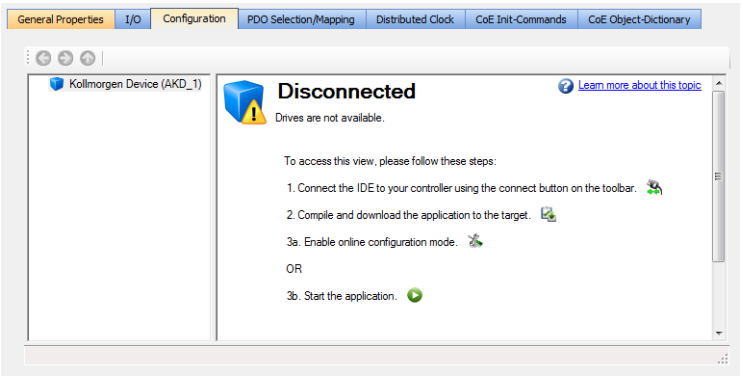


Figure 10-17: Configuration tab

7.7.1.1.2 Online Configuration Mode

In Online Configuration mode, parameters are updated directly to the AKD or AKD2G through TelNet commands with EtherCAT in Pre-Operational mode.

When you modify the value of a parameter, a command is sent to the drive and the corresponding parameter is updated.

7.7.1.1.3 Project Running on the Controller

When a project is running on the controller, parameters are updated directly to the AKD with EtherCAT in Operational mode.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- When the value of a parameter is modified, a command is sent to the drive and the corresponding parameter is updated.
- Beware that your PLC application is running and can also read or write parameters to the AKD.

NOTE

An AKD drive cannot be deleted while it is connected.
The AKD drive leaves Online Configuration mode and starts the application if you start an application while the drive is in that mode.

IMPORTANT

Before leaving Online Configuration mode (either by clicking the Online Configuration button or the Start button when the device is in Online Configuration mode), verify these conditions exist:

1. The drive is not enabled.
2. The service motion is not enabled.
3. Auto-tuning is not enabled.

Recommendation: Click the Disable button to force this scenario.

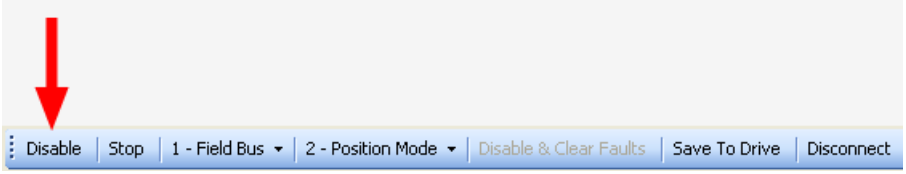


Figure 10-18: Disable button

7.7.1.2 AKD Configuration According to EtherCAT State

The drive configuration can only take place when the EtherCAT is in the Op or Pre-Op state.

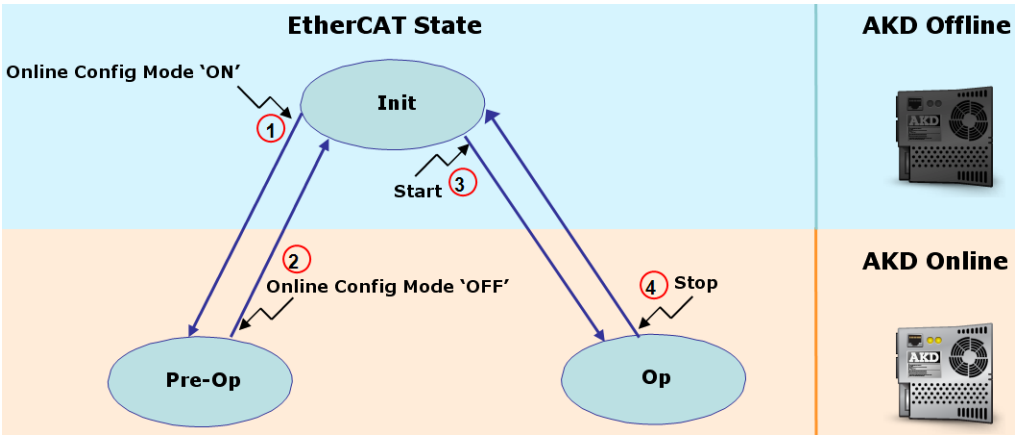


Figure 10-19: AKD Configuration According to EtherCAT State

Action	Name	Description
1	Online Configuration Mode ON	Online Configuration Mode • Sets the EtherCAT fieldbus to the Pre-Op state and allows AKD configuration. • See "Online Configuration Mode" (➔ p. 471) to access this button in the KAS-IDE. • See "Configure the Kollmorgen Servo Drive" (➔ p. 221)) for the procedure.
2	Online Configuration Mode OFF	• Sets the EtherCAT fieldbus to the Init state. • Does not allow AKD configuration. • All AKD drives are disconnected from the KAS-IDE.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Action	Name	Description
3	Start the Project	• Sets the EtherCAT fieldbus to the Operational state [1]. • See "Toolbar Overview" (➔ p. 469) to access this button. • This starts the application program running and allows AKD configuration (with some restrictions for the views: Service Motion and Performance Servo Tuner). • See "Kollmorgen AKD Series Servo Drive" (➔ p. 143).
4	Stop the Project	• This step stops the application from running. • Sets the EtherCAT fieldbus to the Init state. • Does not allow AKD configuration. • All AKD drives are disconnected from the KAS-IDE.

7.7.2 AKD Drive Configuration

7.7.2.1 Connection Modes

The AKD drive parameters can be configured when the KAS-IDE is connected to the controller and operating in one of these functional modes:

- "Online Configuration Mode" (➔ p. 448)
- "Project Running on the Controller" (➔ p. 448)

7.7.2.1.1 Disconnected

If the controller is not operating in one of these modes, then the AKD configurations are not accessible.

The Configuration tab indicates the AKD is Disconnected.

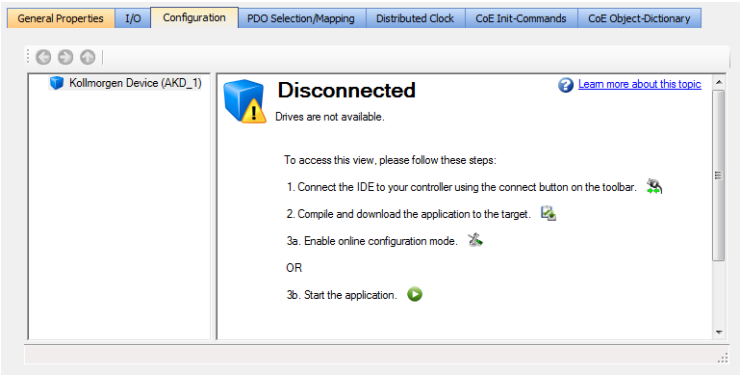


Figure 10-20: Configuration tab

7.7.2.1.2 Online Configuration Mode

In Online Configuration mode, parameters are updated directly to the AKD or AKD2G through TelNet commands with EtherCAT in Pre-Operational mode.

When you modify the value of a parameter, a command is sent to the drive and the corresponding parameter is updated.

7.7.2.1.3 Project Running on the Controller

When a project is running on the controller, parameters are updated directly to the AKD with EtherCAT in Operational mode.

---

[1] Depending on the number of AKD drives physically present in the EtherCAT network, the KAS-IDE can slow down when getting data.
The KAS Runtime is **not concerned** with this limitation.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- When the value of a parameter is modified, a command is sent to the drive and the corresponding parameter is updated.
- Beware that your PLC application is running and can also read or write parameters to the AKD.

NOTE

An AKD drive cannot be deleted while it is connected.
The AKD drive leaves Online Configuration mode and starts the application if you start an application while the drive is in that mode.

IMPORTANT

Before leaving Online Configuration mode (either by clicking the Online Configuration button or the Start button when the device is in Online Configuration mode), verify these conditions exist:

1. The drive is not enabled.
2. The service motion is not enabled.
3. Auto-tuning is not enabled.

Recommendation: Click the Disable button to force this scenario.

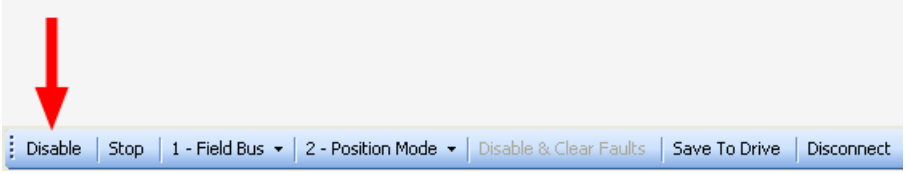


Figure 10-21: Disable button

7.7.2.2 AKD Configuration According to EtherCAT State

The drive configuration can only take place when the EtherCAT is in the Op or Pre-Op state.

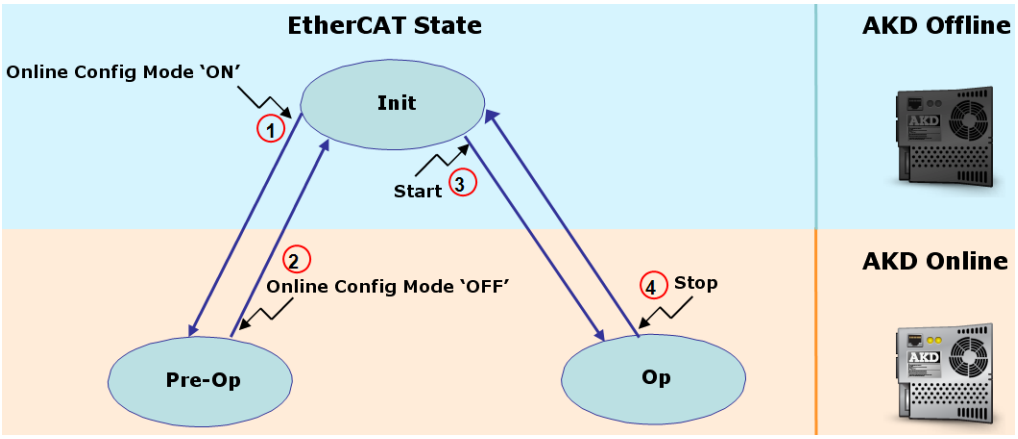


Figure 10-22: AKD Configuration According to EtherCAT State

Action	Name	Description
1	Online Configuration Mode ON	Online Configuration Mode • Sets the EtherCAT fieldbus to the Pre-Op state and allows AKD configuration. • See "Online Configuration Mode" (➔ p. 471) to access this button in the KAS-IDE. • See "Configure the Kollmorgen Servo Drive" (➔ p. 221)) for the procedure.
2	Online Configuration Mode OFF	• Sets the EtherCAT fieldbus to the Init state. • Does not allow AKD configuration. • All AKD drives are disconnected from the KAS-IDE.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Action	Name	Description
3	Start the Project	• Sets the EtherCAT fieldbus to the Operational state [1]. • See "Toolbar Overview" (➔ p. 469) to access this button. • This starts the application program running and allows AKD configuration (with some restrictions for the views: Service Motion and Performance Servo Tuner). • See "Kollmorgen AKD Series Servo Drive" (➔ p. 143).
4	Stop the Project	• This step stops the application from running. • Sets the EtherCAT fieldbus to the Init state. • Does not allow AKD configuration. • All AKD drives are disconnected from the KAS-IDE.

7.8 Tasking Model / Scheduling

In the KAS Runtime, both the Motion and Programmable Logic Controller (PLC) Programs are run every cycle.

- The cycle time is set when configuring the EtherCAT motion bus.
- See "EtherCAT Master Settings tab" (➔ p. 539).
- The cycle time becomes effective only when the Motion Engine is started (i.e., when the PLC code initializes the Motion by calling the MLMotionIni function block) and the application runs on a controller.
  - When the Motion Engine is **not** started, the rate the PLC code runs at is approximately 10 milliseconds.

7.8.1 Priority between Motion and PLC

The Motion computation is always executed each cycle, and occurs before executing the PLC programs application.

Figure 10-23 shows the execution in this order:

1. Motion command, position feedback from each axis, and other elements in the EtherCAT PDO object are sent and received on the EtherCAT motion bus.
   This includes servo drives and Remote I/O.
2. Motion engines are executed.
3. I/O related to the PLC program are serviced.
   See "EtherCAT Processing Time" (➔ p. 451).
4. PLC programs are executed.
5. NVRAM variables are saved.
   See "NVRAM Processing Time" (➔ p. 451).

[1] Depending on the number of AKD drives physically present in the EtherCAT network, the KAS-IDE can slow down when getting data.
The KAS Runtime is **not concerned** with this limitation.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Single Core Controller or Multi-Core Controller in Single-Core Compatibility Mode	Multicore Controller

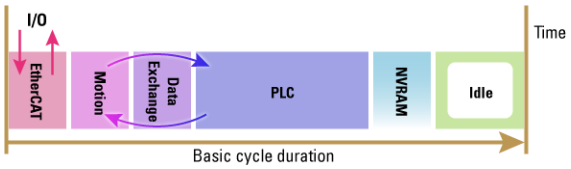


Figure 10-23: Cycle Execution on a Single-core Controller

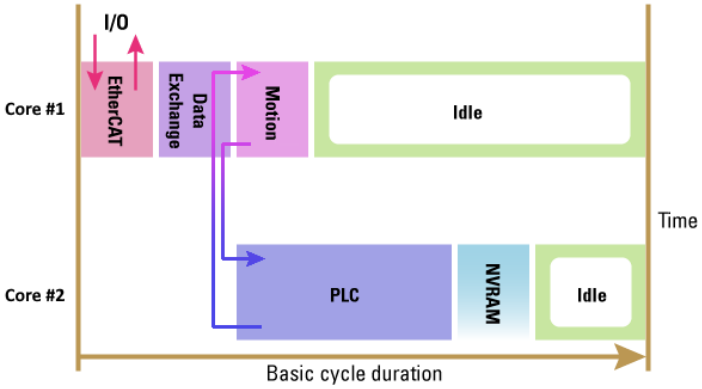


Figure 10-24: Multicore Controller Cycle Execution with Data Exchange (arrows)

NOTE

The Motion time in Figure 10-23 must be shorter than the basic cycle duration at each cycle.
This condition is checked at each cycle and, if the cycle is overran, Kollmorgen Automation Suite generates a fatal error and the application execution is stopped.

7.8.1.1 EtherCAT Processing Time

The EtherCAT frame is executed at the beginning of the cycle.

During this period, all the values related to EtherCAT are exchanged, including:

- **Inputs** are read
- **Outputs** are set

See "PDOs for AKD, AKD2G, AKD-N, Kollmorgen Essentials Drive, S300, or S700" (➔ p. 427).

- Based on the I/O mapping to PLC variables, the I/Os are updated before they are effectively used during the PLC period.
  - See "Map Input and Output to Variables" (➔ p. 526).
- As a consequence, when the PLC variables set an Output, it is updated during the EtherCAT frame of the next cycle.

7.8.1.1.1 Variation during the EtherCAT Processing

The EtherCAT period is subject to time variation along the cycles because of these reasons:

- Some EtherCAT function blocks use asynchronous SDO communication, which is not deterministic.
  - See EtherCAT Function Blocks that Work with SDOs.
- Some EtherCAT slave devices support mailbox protocols.
  - The master cyclically reads the mailbox of the EtherCAT slaves.
  - Polling of the mailbox is performed every 50 cycles
    - It is spread on several cycles depending on the number of EtherCAT slaves.

See the "FAQs" (➔ p. 781) about SDO communication.

7.8.1.2 NVRAM Processing Time

Due to slow processing when saving Retain "Variables" (➔ p. 289) to the NVRAM, this action is not performed each cycle.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The save operation is performed in the background every 20 seconds.
  - The frequency increases to each 2 seconds when the application is running.
- When executed during a cycle, it occurs after the PLC period.

7.8.1.3 Results when a PLC Program is Overrunning the Cycle Duration

Large application can require more than one cycle to completely execute all the PLC programs.

Single Core Controller or Multi-Core Controller in Single-Core Compatibility Mode	Multicore Controller

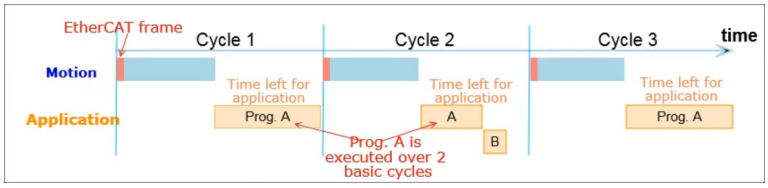


Figure 10-25: Application Overrunning the Basic Cycle on a Single-core Controller

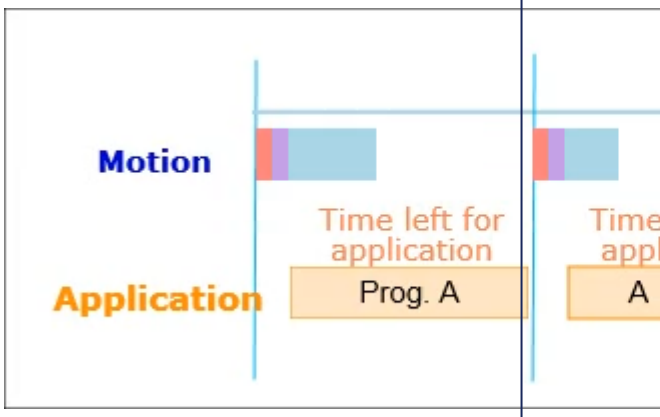


Figure 10-26: Application Overrun Multicore Controller

Figure 10-25 is an example of an application with two PLC programs (A and B). It takes two cycles to execute all the code in the two programs.

- Cycle 1 executes most of Prog. A
- Cycle 2 finishes Prog. A and executes Prog. B, which is set to run with a cycle period of every other cycle.
- Cycle 3 starts executing Prog. A again

Notes

- Even if there is time left over in the cycle, execution of Prog. A does not start until the next cycle.
- An application overrun has no effect other than a short delay in the application execution. Execution of the real-time application is recovered as soon as the overload disappears.
- When running with the KAS Simulator, there is no overrunning because the cycle is extended to include all the PLC programs, right after the Motion computation.

IMPORTANT

If Outputs are set when a program runs over several basic cycles, unexpected and potentially dangerous effects can happen.

See Also

"Change the Priority between PLC Programs" (→ p. 452)

7.8.2 Change the Priority between PLC Programs

PLC programs are assigned a priority.

At times of heavy demand for processing time, the operating system serves programs with higher priority first.

See:

- "Define the PLC Cycle" (→ p. 253)
- "Priority between Motion and PLC" (→ p. 450)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8 KAS-IDE Reference

This section is an in-depth reference to using elements of the KAS-IDE interface.

8.1 Choose a Workspace Layout

8.1.1 Move Child Windows

In the integrated workspace, all child windows are integrated into a single larger application window.

With the MDI/Tabbed workspace command in the Window menu, you can choose to display the child windows either as Tabbed Document Interface (TDI) or as Multiple Document Interface (MDI).

When in MDI mode, you can move and resize the displayed windows.

The Cascade command automatically rearranges all the windows to provide you with easier access to each of them.

8.1.2 Move Toolbox

All toolboxes can be moved within the workspace to a more appropriate location.

- To customize your workspace, click in the Toolbox header and move the window using drag-and-drop.
- The other toolboxes are adapted accordingly.

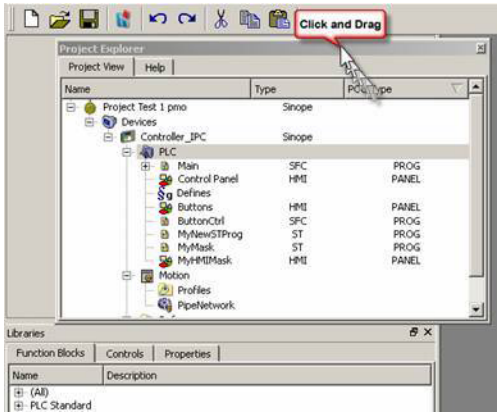


Figure 11-1: Moving Toolboxes

8.1.3 Dock Window

You can separate out a toolbox and change it to a docking window to be placed in the workspace independently of the other toolboxes.

How to change a toolbox to a Docking window?

To do so, click the icon (you can also double-click in the toolbox header).

TIP

Double-click to place the window back into its original position.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

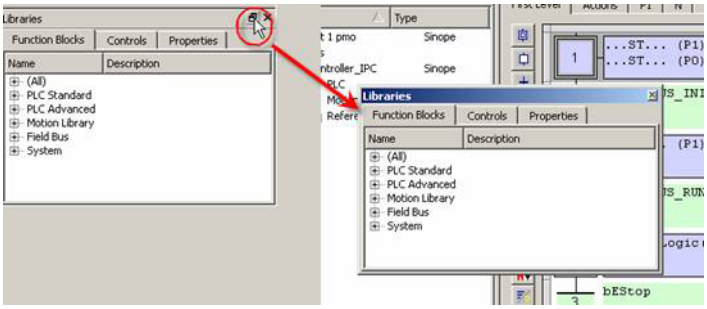


Figure 11-2: Docking window

NOTE

Moving a toolbox to a docking window can lead to problems which can be difficult to recover.

How to undock a window?

If problems arise, drag-and-drop the window to a toolbox border as shown below:

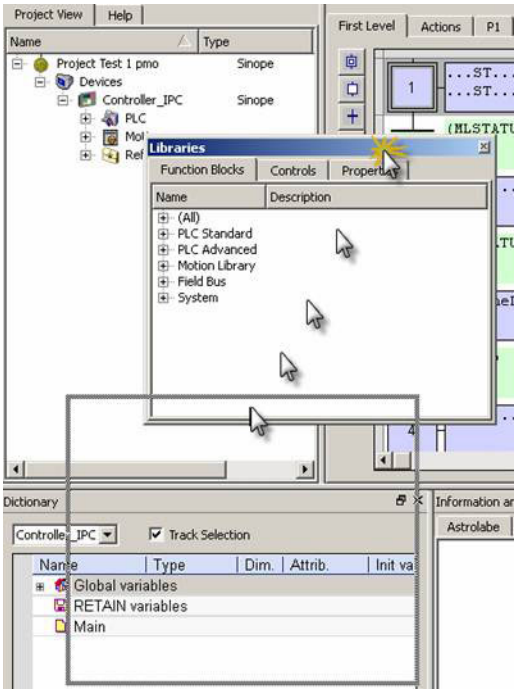


Figure 11-3: Undock window

Dropped in the bottom border of the Project Explorer toolbox, then the **Libraries** toolbox is moved nearby.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

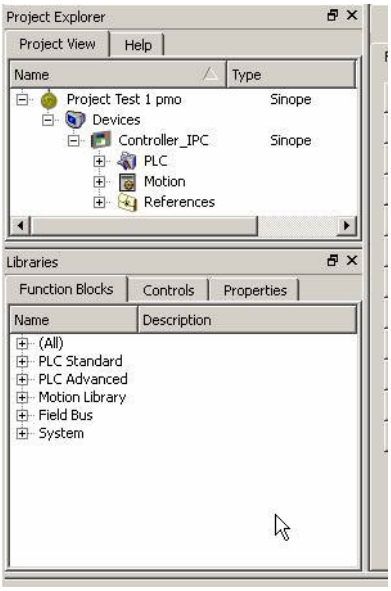


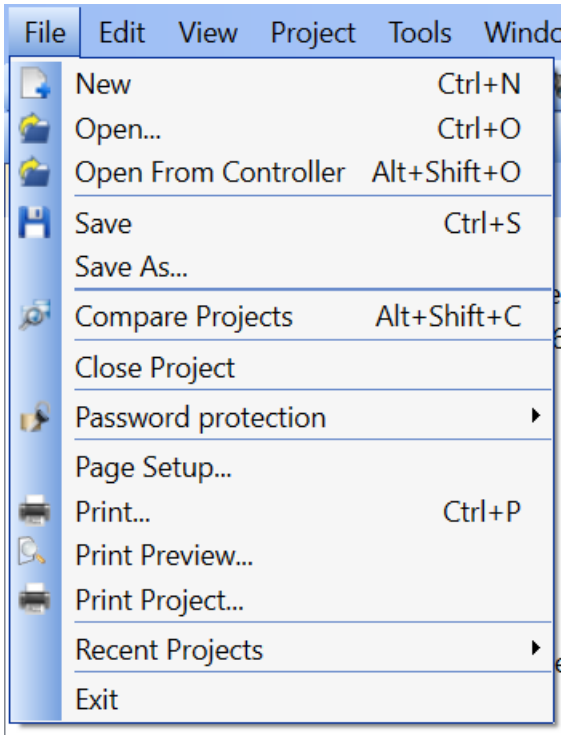
Figure 11-4: Undocked window completed

8.2 Menus Overview

The KAS-IDE contains these menus:

- "File menu" (→ p. 455)
- "Edit menu" (→ p. 460)
- "Project menu" (→ p. 461)
- "Tools menu" (→ p. 464)
- "Window menu" (→ p. 468)
- "Help menu" (→ p. 463)

8.2.1 File menu



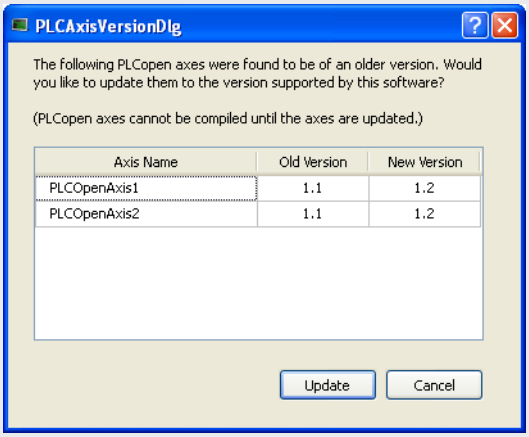
Menu Option	Description
New (Ctrl+N)	Close the current project, if any, and launch the project wizard to create a new one.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Menu Option	Description
Open... (Ctrl+O)	Open an existing project.

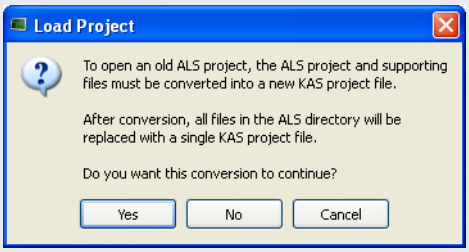
Projects from an older version

When you try to open a previous project, KAS-IDE proposes you to do the conversion to keep the compatibility.



ALS project

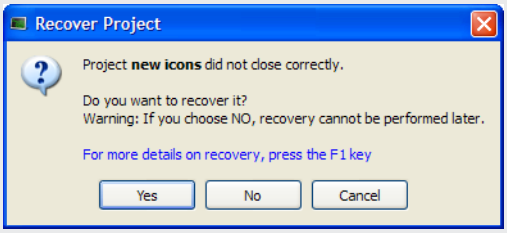
When you try to open a previous project with ALS format, the KAS-IDE proposes you to convert it to the current KAS format.



After conversion, all the files located in the folder structure are replaced with a single KAS file.

Auto Recovery

If the KAS-IDE crashes when you are working with a project, you can start it again to recover the project from the last successful **Save** (or auto save) operation.



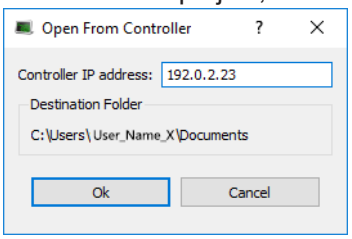
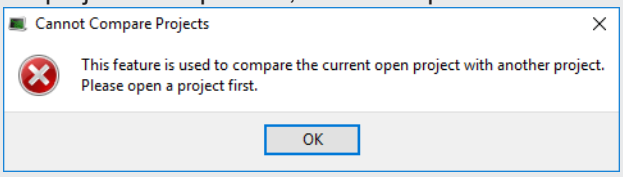
This file can be automatically saved periodically, so you recover recent work instead of the last manually saved file.

See "User Options" (→ p. 464) from the "Tools menu" (→ p. 464).

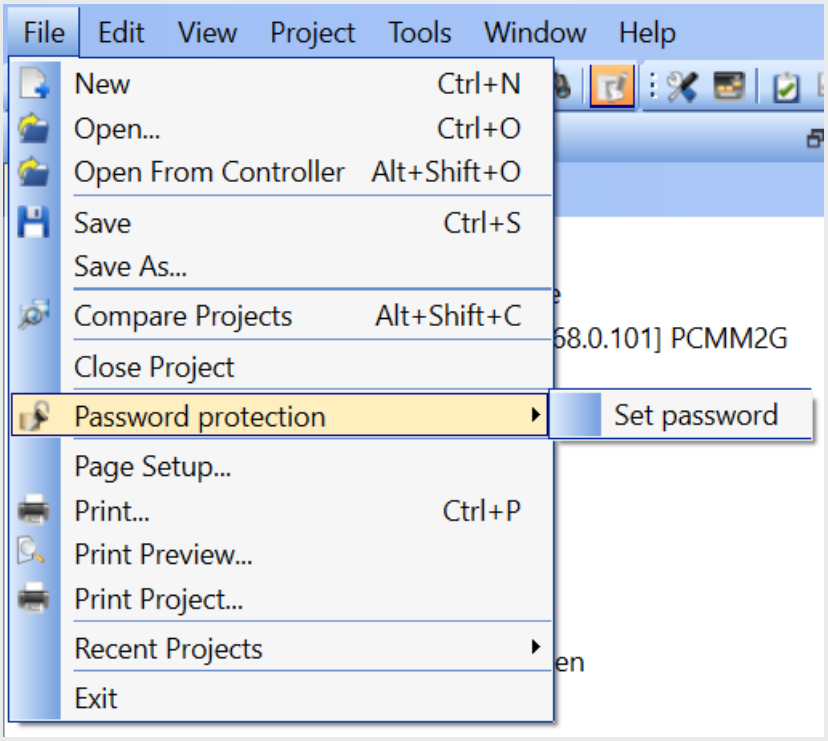
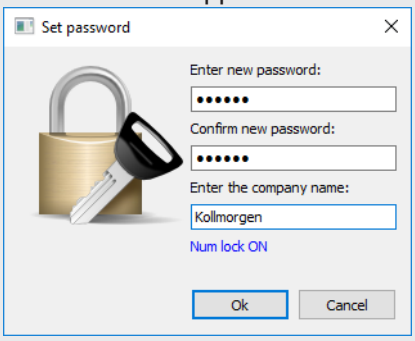
Corruption

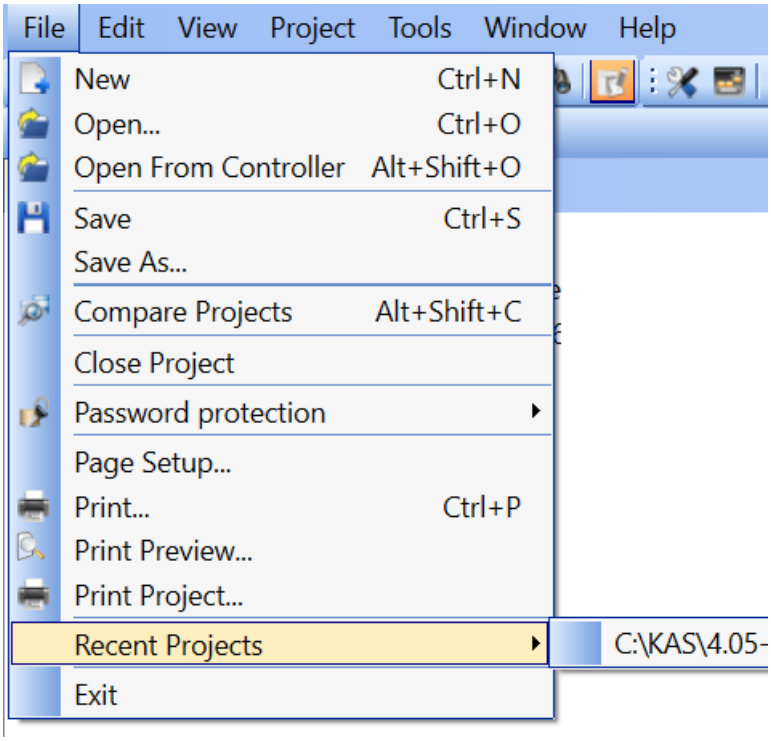
If your project is corrupted, KAS-IDE opens a pop-up window.



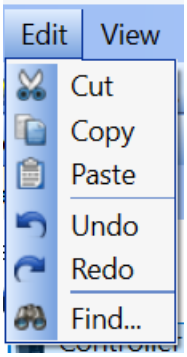
Menu Option	Description
	Or open an existing library.
Open From Controller (Alt+Shift+O)	Project files saved on a controller can be retrieved using this command. To retrieve the project, enter the controller's IP Address.  Successfully retrieved projects are saved in **My Documents** before being opened. See "Download the Project" (→ p. 265) for information on storing projects on the controller.
Save	Save the current project About Auto Save At compile time, an auto-save operation is performed to recover the project if a crash occurs. Saving when in Debug mode automatically switches the program to Edit mode.
Save As...	Save the current project in a location and with a name that you can define. The project name is also in the KAS-IDE window's title.
Compare Projects (Alt+Shift+C)	This option allows for comparison between an open project and another local project. • Once you have browsed for and selected the file for comparison, a window opens showing the differences between the files. • See "Compare PLC Programs" (→ p. 277) for more information on the comparison tool. • A project must be loaded in the IDE to use this function. • If a project is not present, this error opens. 
Close Project	Close the current project. If changes have not been saved, a prompt opens first.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Menu Option	Description
Password Protection	 This option provides the ability to set, change, and remove password protection on a project file to prevent unauthorized access. • The menu options are available when a project is open. • Passwords must be 6-20 characters in length. • They may consist of any alpha-numeric characters (a-z, 0-9, $, &, *, }, etc,). • Spaces, tabs, and apostrophes are not permitted. • A company name must be provided; this should be the name of the company which owns the application source code.  • The status bar has a project security status icon in the lower left corner: • Protected ( ) • Unprotected ( ) • To protect a project, which is stored on the controller, the Project must be compiled and downloaded.
Page Setup...	Define page setup, margins and header/footer.
Print...(Ctrl+P)	Print the project element currently open in the workspace.
Print Preview...	Display a printout on the screen so you can preview it before printing.

Menu Option	Description
Print Project...	Select among the complete project's elements those you want to print.
Recent Projects	 List the most recently used projects.
Exit	Quit KAS-IDE.

8.2.2 Edit menu



Menu Option	Description
Cut	Cut selected data and copy it to the clipboard.
Copy	Copy selected data to the clipboard.
Paste	Paste the data currently stored in the clipboard.
Undo	Undo the last command. **NOTE** This action is not possible for all operations.
Redo	Redo the last command.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Menu Option	Description
Find...	Show the Find and Replace tab in the "Information and Logs window" (→ p. 123) toolbox.

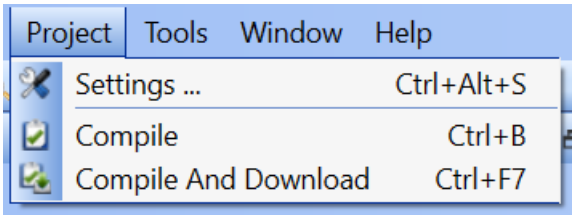
TIP

See "Can I copy and paste or drag and drop between instances of the KAS-IDE" (→ p. 784) about copy/paste functions including drag-and-drop.

IMPORTANT

- Closing the KAS-IDE automatically clears the clipboard.
- Anything that was copied is cleared from memory.
- This means you can not copy something, close the KAS-IDE, and then paste.

8.2.3 Project menu



Menu Option	Description
Settings (Ctrl + Alt + S)	Opens the "Project Settings dialog box" (→ p. 461). See "Configure Project Settings" (→ p. 157).
Compile (Ctrl + B)	Compile the whole project.
Compile and Download (Ctrl + F7)	Compile the whole project and download it to the controller.

8.2.3.1 Project Settings dialog box

Usage

The **Project Settings** dialog box is used to define the project settings stored in the KAS project file and loaded when the project is opened.

See:

- "Configure Project Settings" (→ p. 157)
- "Configure Custom Project Settings" (→ p. 159)

Access

Either:

- Click the "Project menu" (→ p. 461) and select **Settings**.
- On the toolbar, click the  button.

The Project Settings dialog box opens. (Figure 11-5)

Click to view the dialog box

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

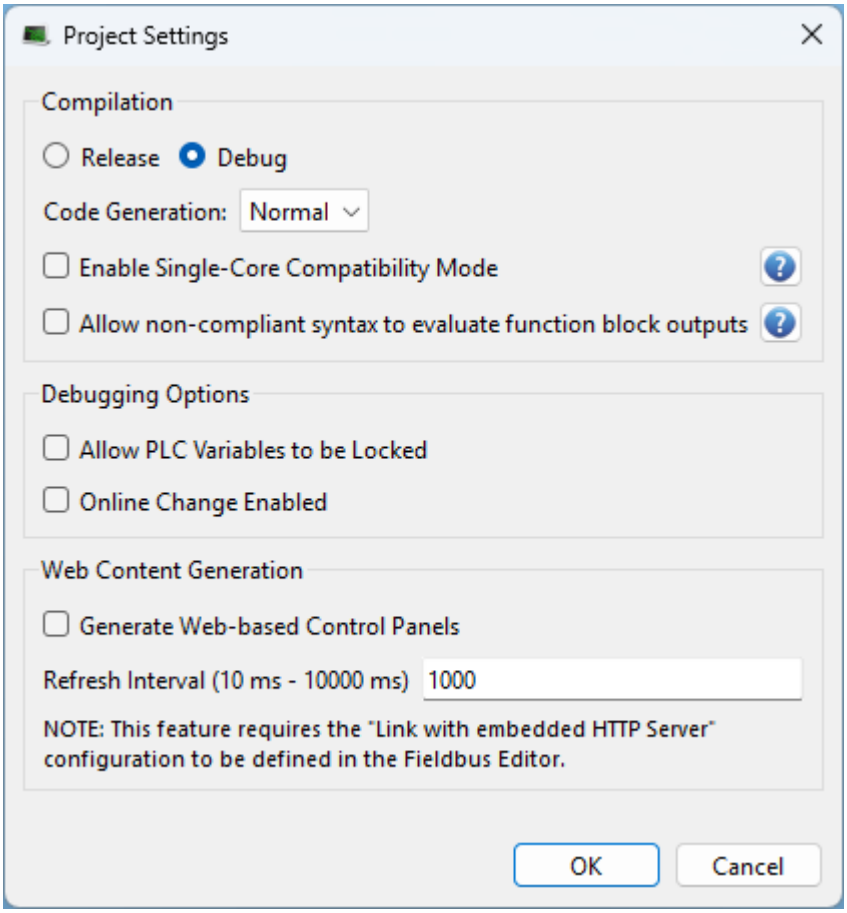


Figure 11-5: Project Settings dialog box

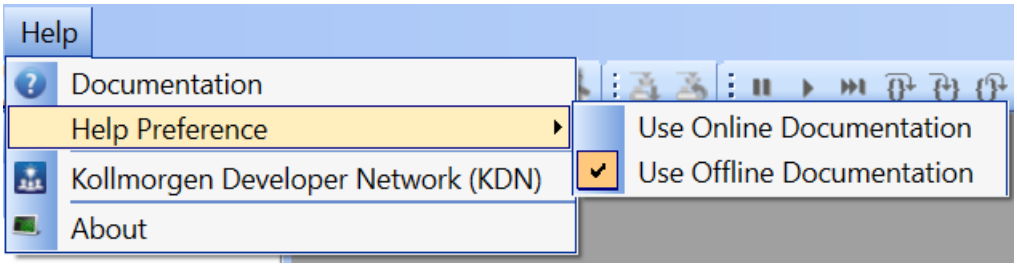
Dialog box Description

Section	Element	Description
Compilation section	**Release** option button	The **Compilation** area is used to define parameters for the target. Select the **Release** option button: • If step-by-step debugging is not required. • To give highest performance to the application.
	Debug option button	Select the **Debug** option button to use "Step-By-Step Debugging" (→ p. 268) during simulation or online testing.
	Code Generation drop-down	Click the **Code Generation** drop-down arrow and select either **Normal** or **Optimized**. • See "Normal" (→ p. 258). • See "Optimized" (→ p. 259).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Section	Element	Description
	Enable Single-Core Compatibility Mode check box	• This option is typically needed for KAS projects developed for AKD PDMM / PCMM single core controllers and migrating to a PCMM2G Controller. • See Migrate from a PxMM to a PCMM2G Controller. Select the **Enable Single-Core Compatibility Mode** check box if: • The application does not follow multi-core controller programming guidelines. • See "Program a Multi-Core Controller" (➔ p. 386). • Reduced performance on the PCMM2G is acceptable for the application.
	Allow non-compliant syntax to evaluate function block outputs check box	Select the **Allow non-compliant syntax to evaluate function block outputs** check box if: • The application was developed using a version of KAS earlier than v4.00. • You are encountering Unexpected complex data in expression compile errors. • You prefer not to update the KAS project to use compliant syntax for the corresponding function blocks. • See "Unexpected complex data in expression." (➔ p. 798).
Debugging Options section	**Allow PLC Variables to be Locked** check box	• Select the **Allow PLC Variables to be Locked** check box to enable PLC variables to be locked (forced). • See "Force a Variable" (➔ p. 279).
	Online Line Change Enabled check box	• Select the **Online Line Change Enabled** check box to update the PLC application while it is running on the controller. • See "PC Online Change" (➔ p. 861).
Web Content Generation section	**Generate Web-based Control Panels** check box	Select the **Generate Web-based Control Panels** check box to allow access to control panel files in the project from the web browser using the KAS Web server.
	Refresh Interval text box	In the **Refresh Interval** text box, specify the interval at which web pages are refreshed to receive PLC variable updates from the KAS Web server.
	OK button	Click **OK** to save the changes or selections and close the dialog box.
	Cancel button	Click **Cancel** to discard all changes or selections and close the dialog box.

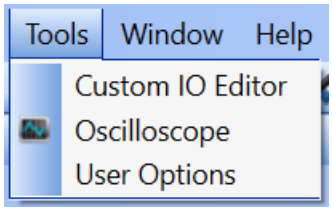
8.2.4 Help menu



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Menu Option	Description
Documentation	Opens the help system.
Help Preference	As stated in Is the Help System Online or Offline?, there is more than one type of help system with different advantages. • Click the **Use Online Documentation** option to open the online help. • This is the default selection. • Click the **Use Offline Documentation** option to open the locally installed help (if available).
Kollmorgen Support Network (KSN)	Opens a web browser to the Kollmorgen Support Network, a community support site.
About	• Shows version numbers and other information about the installed KAS-IDE. • See "Find the KAS-IDE Version" (→ p. 151).

8.2.5 Tools menu



Menu Option	Description
Custom IO Editor	**NOTE** This tool is reserved for PROFIBUS fieldbuses only. See "Custom Input/Output Editor" (→ p. 708).
Oscilloscope	Show the "Soft Oscilloscope" (→ p. 737).
User Options	Click to open the "User Options dialog box" (→ p. 464).

8.2.5.1 User Options dialog box

Usage

The **User Options** dialog box is used to define specific KAS loading, saving, backup settings.

Access

On the "Tools menu" (→ p. 464), click **User Options**.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

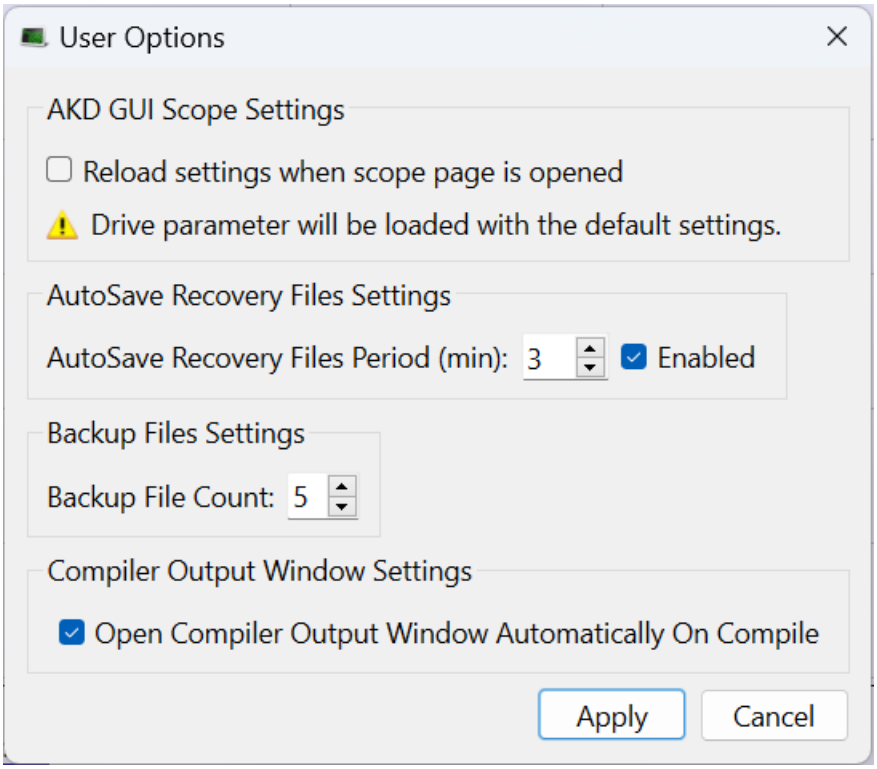
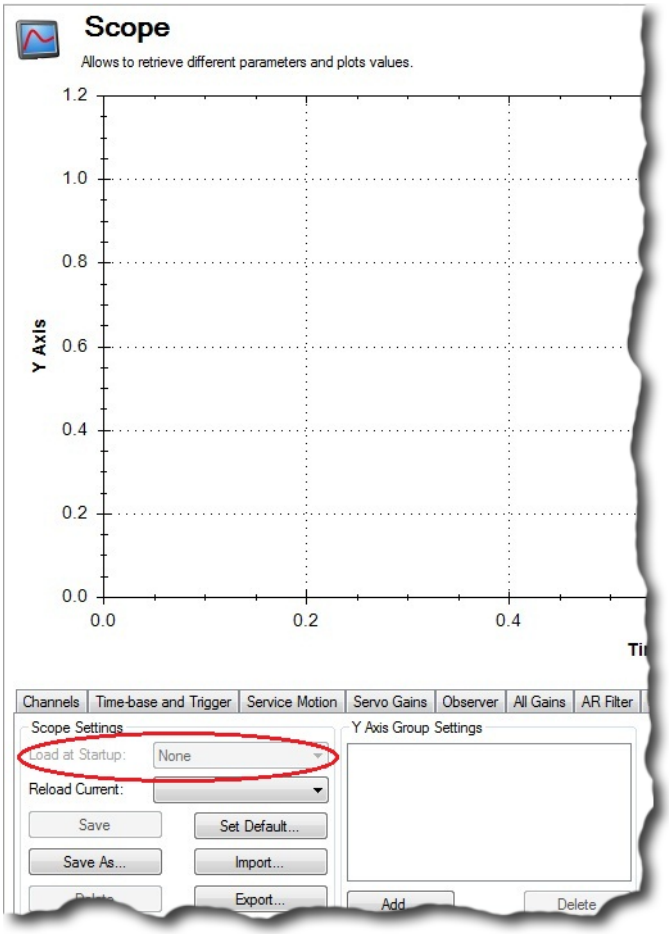
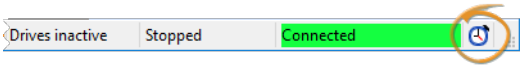


Figure 11-6: User Options dialog box

Dialog box Description

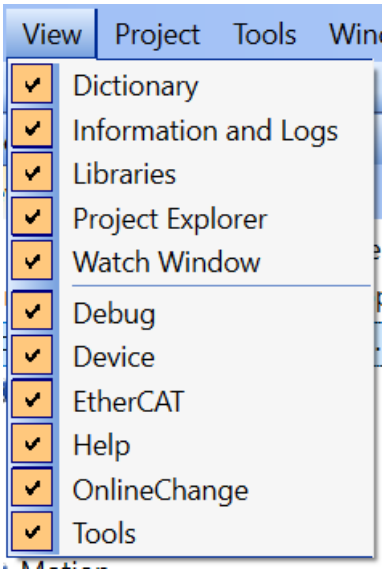
Section	Element	Description
AKD GUI Scope Settings section	**Reload settings when scope page is opened** check box	Select the check box in this area to enable or disable the **Load at startup** option in the AKD GUI Scope Settings tab. (Figure 11-7)  **Figure 11-7:** Scope Settings tab > Load at Startup drop-down
AutoSave Recovery File Settings section	**AutoSave Recovery Files Period (min):** spin box	This area is used to enable or disable automatically saving project recovery files at a specified interval. • In the **AutoSave Recovery Files Period (min):** spin box, enter the time to automatically save the project recovery files. • The default is 3 minutes.
	AutoSave Recovery Files Period (min): Enabled check box	• Once set, this applies to all further project recovery files for the user. • The stopwatch icon in the bottom right corner of the KAS-IDE indicates whether AutoSave Recovery Files is active ( ) or disabled ( ). • See "How does file recovery work?" (➔ p. 785) about recovery files. 

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Section	Element	Description
Backup File Settings section	**Backup File Count** spin box	This area is used to configure the number of automatic backup of KAS project or library files when the project or library is saved. • In the **Backup File Count** spin box, enter the maximum number of backups. • Backup files are stored in the same directory as the KAS project or library that is being saved. • Backup files have the extension `.backup[nn].kas` or `.backup[nn].kal`" where **[nn]** represents two digits. • The most recent backup of a KAS project is **.backup01.kas**; the next most recent backup is **.backup02.kas**, and so on.
Compiler Output Window Settings section	**Open Compiler Output Window Automatically On Compile** check box	Select the **Open Compiler Output Window Automatically On Compile** check box to automatically open the "Compiler Output tab" (→ p. 136) each time a KAS project is compiled.
	Apply button	Click **OK** to save the changes or selections and close the dialog box.
	Cancel button	Click **Cancel** to discard all changes or selections and close the dialog box.

8.2.6 View menu

The **View** menu is used to view or hide areas of the KAS window.

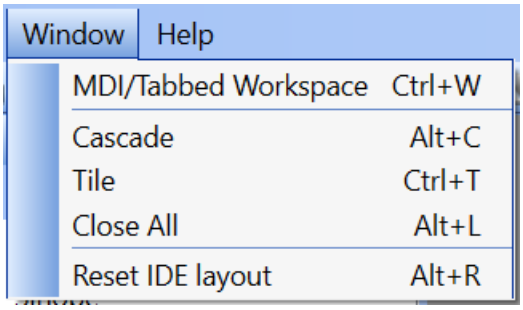


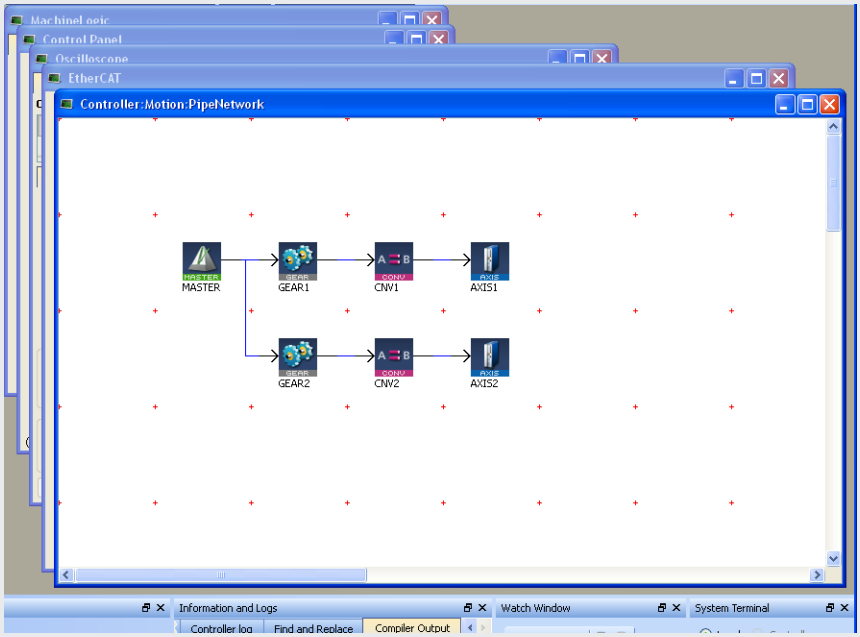
Menu Option	Description
Dictionary	Click to view or hide the "Dictionary" (→ p. 101).
Information and Logs	Click to view or hide the "Information and Logs window" (→ p. 123).
Libraries	Click to view or hide the "Libraries" (→ p. 100).
Project Explorer	Click to view or hide the "Project Explorer" (→ p. 85).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

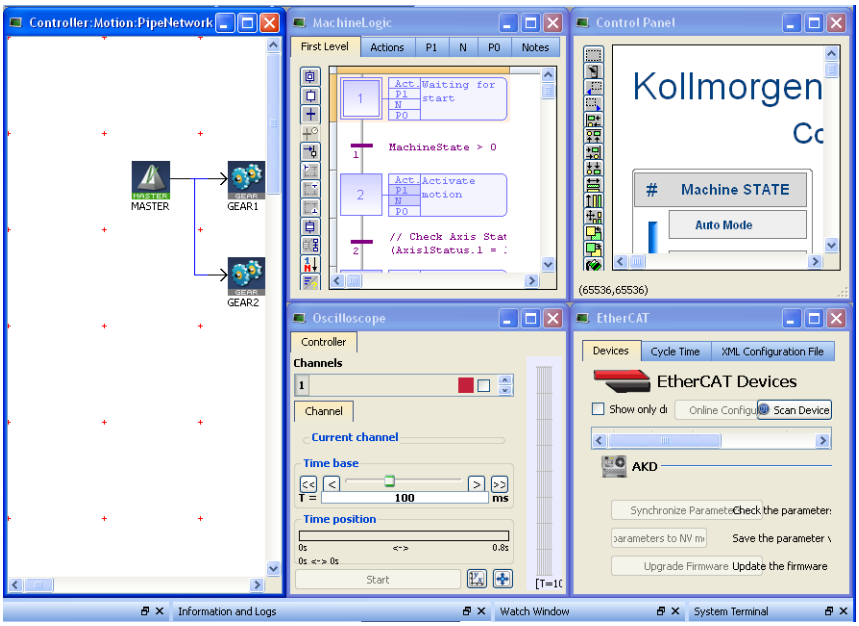
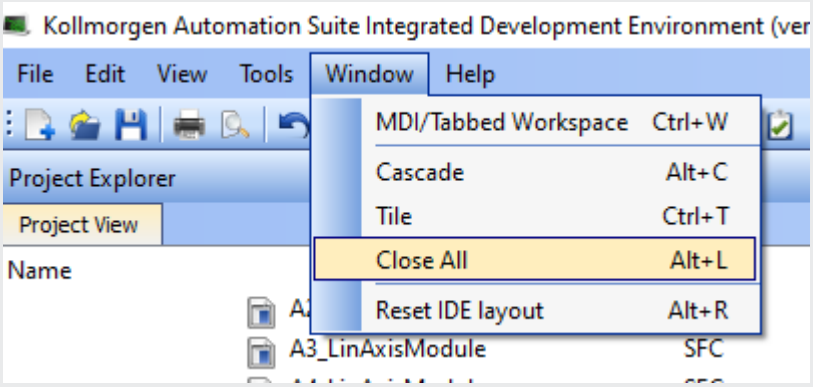
Menu Option	Description
Watch Window	Click to view or hide the "Watch Window" (→ p. 138).
Debug	Click to view or hide the "Debug toolbar" (→ p. 470).
Device	Click to view or hide the "Device toolbar" (→ p. 470).
EtherCAT	Click to view or hide the "EtherCAT Toolbar" (→ p. 471).
Help	Click to view or hide the Help toolbar.
OnlineChange	Click to view or hide the "Online Change toolbar" (→ p. 472).
Tools	Click to view or hide the "Main (Tools) toolbar" (→ p. 471).

8.2.7 Window menu



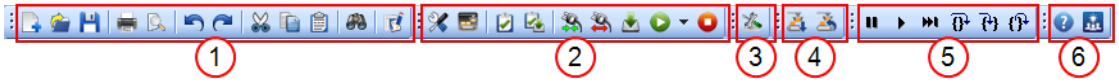
Menu Option	Description
MDI/Tabbed Workspace (Ctrl+W)	Toggle the workspace between the MDI and the tabbed mode.
Cascade (Alt+C)	Re-arrange all workspace children windows in cascade mode. 

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Menu Option	Description
Tile (Alt+T)	Re-arrange all workspace children windows in tile mode. 
Close All (Alt+L)	
Reset IDE Layout (Alt+R)	This option resets widgets to their default location and visibility. If an item is hidden or floating, it is made visible and returned to its default position. These interface items are affected by this option: • "Dictionary" (→ p. 101) • "Information and Logs window" (→ p. 123) • "Libraries" (→ p. 100) • "Project Explorer" (→ p. 85) • "Toolbar Overview" (→ p. 469) • "Watch Window" (→ p. 138)

8.3 Toolbar Overview

These are the KAS-IDE toolbars:



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. "Main (Tools) toolbar" (→ p. 471)
2. "Device toolbar" (→ p. 470)
3. "EtherCAT Toolbar" (→ p. 471)
4. "Online Change toolbar" (→ p. 472)
5. "Debug toolbar" (→ p. 470)
6. Links to the **Help** and Kollmorgen Developer Network (KDN).

A specific toolbar is available for the "Kollmorgen AKD Series Servo Drive" (→ p. 143).

See these sections about icons available in the graphical PLC editors:

- FBD toolbar
- FFLD toolbar
- SFC toolbar

8.3.1 Debug toolbar

This table describes each **Debug** toolbar button.

Button	Name	Description
	Pause	Pause the application in Cycle to Cycle mode.
	Restart	Restart the target in **Normal** execution mode (RUN).
	Cycle Step Execution	Execute a cycle step (from the current position up to the end of the last program).
	Step Over	**Step Over** the next instruction: If the next instruction is a call of a function block or a sub-program, the execution passes over to the following instruction.
	Step Into	**Step Into** the next instruction: • The next step is at the beginning of the called block. • If the next instruction is **not** a call of a function block or a sub-program, the **Step Into** behaves like the **Step Over**.
	Step Out	**Step Out** the current block: • If the current stepping position is in a called function block or a sub-program, the execution continues up to the end of the current block. • Otherwise, the **Step Out** behaves like the **Step Over**.

8.3.2 Device toolbar

This table describes each **Device** toolbar button.

Button	Name	Description
	Project Settings	Click to open the "Project Settings dialog box" (→ p. 461). See "Configure Project Settings" (→ p. 157).
	Simulator	Change the Controller IP address to connect with KAS Simulator. See "Using the KAS Simulator" (→ p. 586).
	Compile project	Compile the project.
	Compile and Download	Compile and download the project.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Button	Name	Description
	Connect	Establish a connection with the target Controller. For possible statuses, see the "Color Codes" (→ p. 146) table for the labels of the status bar.
	Disconnect	Close connection with the target Controller.
	Download	Download the application to the targeted Controller (Note that the application must **not** be running). See "Download the Project" (→ p. 265).
	Start	Start the application. It can be either a **Warm** or **Cold** start.
	Stop	Stop the application.

NOTE

- The **Allow Variables to be Locked** option cannot be changed while the **Online Change Enabled** option is selected.
- If Online Change is enabled, deselect the Online Change Enable option first, enable or disable PLC Variable Locking, then re-enable Online Change, and re-compile the project.

8.3.3 EtherCAT Toolbar

This table describes each **EtherCAT** toolbar button.

Button	Name	Description
	Online Configuration Mode	This action is only available when the KAS Runtime is connected with the target Controller. • Click this toggle button to change the mode (ON / OFF). • After the scan has been performed, and your project is compliant with the physical devices on the EtherCAT network, you can activate the **Online Configuration Mode**. ◦ Online Configuration Mode allows setting up AKD and AKD2G drives in an EtherCAT installation. ◦ In this mode, KAS-IDE communicates with the drives through the integrated views of the AKD Setup Screens or with the AKD Setup Wizard. ◦ KAS-IDE displays a quick status overview of all the drives. • The AKD Setup Screens allow functions (e.g., enabling / disabling the drive, service motion, tuning, and a scope) where you can plot up to six different parameters from the drive. ◦ See "AKD Setup Wizard" (→ p. 224). ◦ See "FAQs" (→ p. 781) for a potential issue when resetting the factory parameters.

8.3.4 Main (Tools) toolbar

This table describes each **Main** KAS-IDE (Tools) toolbar button.

Button	Name	Description
	New	Create a new project

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Button	Name	Description
	Open	Open an existing project
	Save	Save the project
	Print	Print the project item currently open in the workspace See (→ # 282, Print) for more information.
	Print Preview	Print preview
	Undo	Undo
	Redo	Redo
	Cut	Cut
	Copy	Copy
	Paste	Paste
	Find	Find
	Toggle	Toggle Edit/Debug mode (enabled when the application is running). See (→ # 861, PC Online Change).

8.3.5 Online Change toolbar

This table describes each **Online Change** toolbar button.

Button	Name	Description
	PLC Online Change	When Online Change has been activated in the PLC options, the new code is loaded even if the application is running. See the Warning in (→ # 864, Activate an Online Change).
	PLC Online Change Revert to previous version	Revert the changes made after an Online Change and return to the previous application.

8.4 Windows Standard Conventions

8.4.1 Windows Manipulation

These standards apply to the KAS-IDE windows:

- Move
- Resize
- Minimize
- Maximize
- Close (Alt+F4)

Press **Esc** to exit a pop-up window.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.4.2 Mouse Manipulation

- Double-click an item to open it (e.g. double-click a program in the Project Explorer to open it in the appropriate editor)
- Right-click to open the menu and give access to the relevant commands (e.g. to add a variable to the Dictionary)

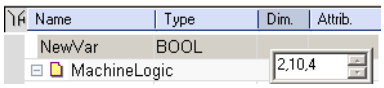
8.4.3 Table Manipulation

8.4.3.1 Sorting Items

- If the sort feature is implemented, you can click in the column header to sort all the items according to one of the available parameters.
- Click again to alternately sort in ascending or descending order.

8.4.3.2 Selecting a Cell

1. Click a cell in the table to select it.
2. Once selected, press **F2** to edit the value.

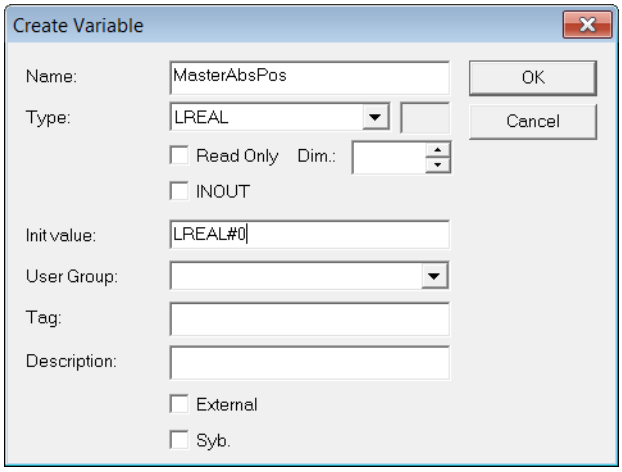


TIP

A double-click directly opens the pop-up window for editing.

8.4.3.3 Selecting a Row

1. When available, press the **Spacebar** to toggle the selection mode from cell to row.
2. Click a cell in the table to select the entire row.
3. Once selected, press **F2** to edit the values of the row.



8.4.3.4 Resize a Column

- If you want to enlarge a column width to make more content visible, put the mouse in the table header between two columns so the cursor change to the following  and move right or left to resize your column.
- After this operation, you need to scroll horizontally to see the other columns.

8.5 Keyboard Shortcuts

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This is a list of accelerator keys:

- "Common Keyboard Shortcuts" (→ p. 474)
- "FBD Editor Keyboard Shortcuts" (→ p. 475)
- "FFLD Editor Keyboard Shortcuts" (→ p. 476)
- "SFC Editor Keyboard Shortcuts" (→ p. 479)
- "ST Editor Keyboard Shortcuts" (→ p. 480)
- "Graphic Editor Keyboard Shortcuts" (→ p. 479)
- "Table Keyboard Shortcuts" (→ p. 480)
- "Trace Zoom Feature" (→ p. 754)

NOTE

A shortcut can be unavailable depending on the context.

8.5.1 Common Keyboard Shortcuts

Shortcut	Command
-	Collapse
+	Expand
Alt + Shift + C	Compare projects
Alt + Shift + O	Open from controller
Ctrl + B	Compile the project
Ctrl + C	Copy
Ctrl + F	"Find and Replace tab" (→ p. 129)
Ctrl + F7	Compile and Download the project
Ctrl + N	New
Ctrl + O	Open
Ctrl + P	"Print" (→ p. 282)
Ctrl + S	Save
Ctrl + Shift + F	Find All
Ctrl + V	Paste
Ctrl + X	Cut
Ctrl + Y	Redo
Ctrl + Z	Undo
Del	Delete
F1	Display the help
Page Up/Down	Scroll Page up/down
RETURN	Equivalent to double-click

TIP

See "Can I copy and paste or drag and drop between instances of the KAS-IDE" (→ p. 784) about copy/paste functions including drag-and-drop.

IMPORTANT

- Closing the KAS-IDE automatically clears the clipboard.
- Anything that was copied is cleared from memory.
- This means you can not copy something, close the KAS-IDE, and then paste.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.5.2 FBD Editor Keyboard Shortcuts

- "FBD Editor (common)" (➔ p. 475)
- "FBD Editor (when editing)" (➔ p. 475)
- "FBD Editor (during debug)" (➔ p. 475)

8.5.2.1 FBD Editor (common)

Shortcut	Command
Arrows	Scroll the window.
Ctrl + d	Display the FBD execution order.
Ctrl + mouse-wheel or + / - on the keypad	Zoom in / out.
Ctrl + page UP/DOWN	Go to the previous/next section.
Ctrl + Shift + End	Select all items from the cursor position to the end of the document.
Ctrl + Shift + Home	Select all items from the begin to the cursor position.
Ctrl+F2	Toggle Bookmark First, select the Network header.
Ctrl+Shift+F2	Go to the Previous Bookmark.
Escape	Cancel linking/resizing/dragging. • If selection: Deselect • If no selection: Select mode active
Page UP/DOWN	Scroll the page up/down.
Return	Equivalent to double-click.
Shift+F2	Go to the Next Bookmark.
Tab	Select the next position item.
Tab + shift	Select the previous position item.

8.5.2.2 FBD Editor (when editing)

Shortcut	Command
char	Start editing a symbol (e.g., variable, constant, instance). On jump/comment/break: open dialog box to enter text.
Ctrl + arrows	Align the selected items.
Del	Delete the selection.
Shift + arrows	Move the selection.
Shift + page UP/DOWN	Move the selection (4 cells).
Spacebar	Swap the item style.
Ctrl + Shift + down	Insert blank lines at the position of the mouse.

8.5.2.3 FBD Editor (during debug)

Shortcut	Command
Spacebar	Swap TRUE/FALSE Boolean value.
*	Lock var.
/	Unlock var.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.5.3 FFLD Editor Keyboard Shortcuts

- "FFLD Editor (when editing)" (→ p. 476)
  - "Bookmark" (→ p. 476)
  - "Delete" (→ p. 476)
  - "Edit" (→ p. 476)
  - "Find" (→ p. 477)
  - "Insert" (→ p. 477)
  - "Move" (→ p. 477)
  - "Select" (→ p. 478)
  - "Trace" (→ p. 478)
- "FFLD Editor (during debug)" (→ p. 478)

8.5.3.1 FFLD Editor (when editing)

8.5.3.1.1 Bookmark

Shortcut	Command
Ctrl+F2	Toggle Bookmark (you must first select the Network header)
Ctrl+Shift+F2	Go to Previous Bookmark
Shift+F2	Go to Next Bookmark

8.5.3.1.2 Delete

Shortcut	Command
Delete Key	Delete cell, selection, or row
Shift+Delete	Delete Network

8.5.3.1.3 Edit

Shortcut	Command
Ctrl + mouse-wheel or + / - on the keypad	Zoom in / out
Ctrl+C	Copy Item
Ctrl+S	Save
Ctrl+V	Paste Item
Ctrl+X	Cut Item
Ctrl+Y	Redo
Ctrl+Z	Undo
Esc or Shift-ESC	Close the rename widget. Exit Dialog
Return	Equivalent to double click
Space	Change contact or coil

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.5.3.1.4 Find

Shortcut	Command
Ctrl + Shift + F	Find All
Ctrl+F	Find
F3	Find Next

8.5.3.1.5 Insert

Shortcut	Command
Ctrl+Shift+A	Insert Contact NC, Negative Transition
Ctrl+Shift+B	Insert Wire (both)
Ctrl+Shift+C	Insert Contact NC
Ctrl+Shift+D	Insert Coil De-Energize
Ctrl+Shift+E	Insert Coil Energize
Ctrl+Shift+H	Insert Horizontal Wire
Ctrl+Shift+I	Insert Contact NC, Positive Transition
Ctrl+Shift+J	Insert Jump
Ctrl+Shift+K	Insert a positive coil to the destination cell
Ctrl+Shift+L	Insert a negative coil to the destination cell
Ctrl+Shift+M	Insert Data In
Ctrl+Shift+N	Insert Contact NO, Negative Transition
Ctrl+Shift+O	Insert Contact NO
Ctrl+Shift+P	Insert Contact NO, Positive Transition
Ctrl+Shift+Q	Insert Data Out
Ctrl+Shift+R	Insert Coil Reset (Unlatch)
Ctrl+Shift+S	Insert Coil Set (Latch)
Ctrl+Shift+T	Insert Return
Ctrl+Shift+V	Insert Vertical Wire
Ctrl+Shift+W	Insert Data In Inverted
F8	Insert FB
Insert Key	Insert Row
Shift+C	Insert Comment
Shift+Insert	Insert Network

8.5.3.1.6 Move

Shortcut	Command
Arrows	Move the focus cell or scroll through ladder.
Ctrl+End	Go to the End of Network.
Ctrl+End followed by Ctrl+End	Go to the End of Ladder.
Ctrl+Home followed by Ctrl+Home	Go to the Top of Ladder.
Ctrl+Home or Home followed by Home	Go to the Top of Network
Ctrl+Left Arrow or Home	Move the focus to begin of row.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Shortcut	Command
Ctrl+Page Down	Go to the Next Network.
Ctrl+Page Up	Go to the Previous Network.
Ctrl+Right Arrow or End	Move the focus to end of row.
Page Down	Scroll 1 page down.
Page up	Scroll 1 page up.
Shift+Tab	Move the focus cell left.
Tab	Move the focus cell right.

8.5.3.1.7 Select

Shortcut	Command
Ctrl+A	Select the contents of a network/rung.
Ctrl+A followed by Ctrl+A	Select the entire ladder.
Ctrl+Shift+ End	Select from the current cell to the end of network. Bottom element of network and the furthest to the right.
Ctrl+Shift+ Home	Select from current cell to beginning of network.
Ctrl+Shift+ left Arrow or Shift+Home	Select from current cell to beginning of line.
Ctrl+Shift+ right Arrow or Shift+End	Select from current cell to end of line.
Shift+Arrow	Select multiple cells.
Shift+down Arrow	Select Row above or below when focus is on the left rail.
Shift+left Arrow	Select the current cell and one cell to left.
Shift+Page Down	Selection Page-Down.
Shift+Page Up	Selection Page-Up.
Shift+right Arrow	Select the current cell and one cell to right.
Shift+up Arrow	Select the cell above or below when focus is on the cell.

8.5.3.1.8 Trace

Shortcut	Command
Ctrl+I	Trace the Vertical Wire Up.
Ctrl+J	Trace the Horizontal Wire Left.
Ctrl+K	Trace the Horizontal Wire Right.
Ctrl+M	Trace the Vertical Wire Down.

8.5.3.2 FFLD Editor (during debug)

Shortcut	Command
Spacebar	Swap TRUE/FALSE Boolean value
*	Lock var
/	Unlock var

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.5.4 Graphic Editor Keyboard Shortcuts

Shortcut	Command
Arrow	Scroll
Ctrl + arrow	Align on main selected item
Ctrl + F2	Toggle bookmark
Ctrl + mouse-wheel down or Shift+MINUS Sign (-) on the numerical keypad	Zoom out
Ctrl + mouse-wheel up or Shift+PLUS Sign (+) on the numerical keypad	Zoom in
Ctrl + Shift + End	Select all items from the cursor position to the end of the document
Ctrl + Shift + Home	Select all items from the begin to the cursor position
Del	Delete selection
Escape	Cancel resizing/dragging If selection: Unselect If no selection: Select mode active
Shift + Arrow	Move selection
Shift + F2	Go to next bookmark
Shift + Page UP/DOWN	Offset selection
Tab	Select next position item
Tab + shift	Select previous position item

8.5.5 SFC Editor Keyboard Shortcuts

Shortcut	Command
?	Show/Hide notes
arrows	Move caret
b or B	Insert macro body
c or C	Insert convergence
Ctrl + mouse-wheel or + / - on the keypad	Zoom in / out
Ctrl + return	Edit reference
d or D	Insert divergence
Del	Delete selection
i or I	Insert step initial
j or J	Insert jump
m or M	Insert macro
Page UP/DOWN	Scroll page up/down
Return	Equivalent to double-click

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Shortcut	Command
s or S	Insert step
Shift + arrows	Select multiple cells
Shift + Home	Select from left to caret
Shift + Page Up/Down	Selection Page Up/down
Spacebar	Swap item style
t or T	Insert transition
x or X	Insert the left side corner of a divergence/convergence

8.5.6 ST Editor Keyboard Shortcuts

- "ST Editor (common)" (➔ p. 480)
- "ST Editor (when editing)" (➔ p. 480)
- "ST Editor (during debug)" (➔ p. 480)

8.5.6.1 ST Editor (common)

Shortcut	Command
Arrows	Move caret
Shift+arrows	Selection
Ctrl+left/right arrow	Go to previous/next word
Shift+Ctrl+left/right arrow	Select previous/next word
Ctrl+F2	Toggle Bookmark First, select the Network header)
Shift+F2	Go to Next Bookmark
Ctrl+Shift+F2	Go to Previous Bookmark
Ctrl+mouse-wheel or + / - on the keypad	Zoom in / out

8.5.6.2 ST Editor (when editing)

Shortcut	Command
.	Select member of a structure or instance.
Ctrl + Spacebar	Auto completion or Open the variable selector dialog.
Ctrl+Shift+Spacebar	Opens a list of all the standard functions.
Ctrl + J	Auto completion or Open the variable selector dialog (an alternative method).

8.5.6.3 ST Editor (during debug)

Shortcut	Command
*	Lock variable
/	Unlock variable
Shift+double-click	Force a variable
Spacebar	Toggle the Boolean value or activate the dialog, lock, unlock the variable. Equivalent of Shift + double click.

8.5.7 Table Keyboard Shortcuts

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Shortcut	Command
Arrows	Move selection
Shift + Tab	Move selection to the left
Spacebar	Line selection/cell selection
Tab	Move selection to the right

8.6 Bookmarks

Bookmarks are used for navigating in a document.

- Insert bookmarks anywhere in a document.
  - Use them to jump from one bookmark to another with a single command for browsing the document.
- Bookmarks are supported in all program editors plus the Variable editor.

These are the available commands for using bookmarks:

Command	Description
Ctrl + F2	Toggle the bookmark at the current position.
Shift + F2	Go to the next bookmark.

According to the type of document, the possible locations for a bookmark are:

- In the FBD editor, a bookmark is placed on any FBD object (not on a line).
- In the FFLD editor, a bookmark is placed on a rung header.
- In the SFC editor, a bookmark is placed on an SFC symbol (step, transition, jump...).
- In the text editor, a bookmark is placed on a line of text.
- In the Variable editor, a bookmark is placed on any line of the grid (variable or group).

NOTE

Bookmarks are valid only while the editing window is open.
They are not stored in the document when the window is closed.

8.7 Programs: Structure and Syntax

This section provides details on the syntax, structure, and use of the declarations and statements supported by the KAS-IDE application language.

8.7.1 Project Structure

Structuring the application with care is important in creating your project.

See "Project Structure Guidelines" (→ p. 1007) in the "Advanced Topics" (→ p. 820).

8.7.2 IEC 61131-3 Editors

The KAS-IDE programming environment provides language dedicated editors for:

- "Free Form Ladder Diagram (FFLD) Editor" (→ p. 496)
- "Function Block Diagram (FBD) Editor" (→ p. 487)
- "Sequential Function Chart (SFC) Editor" (→ p. 505)
- "Structured Text (ST) / Instruction List (IL) Editor" (→ p. 511)

When must SFC be used?

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- SFC must be used when you need to manage sequences of stable process states.
- Using SFC avoids complex switches and the declaration of multiple flags in programs.

When must SFC **not** be used?

- SFC must never be used as a decision diagram or flow chart for describing an algorithm (e.g., think If / Then / Else...).
  - This leads to complex SFC charts and bad performances at run-time.
- Never use a step to represent an intermediate point within a calculation.
  - Use ST in this case.

See "Program Limitations" (➔ p. 310) and the "PC Online Change" (➔ p. 861) feature.

8.7.3 Tips

8.7.3.1 Drag-and-Drop

The editor provides a programming environment, including drag-and-drop features:

- Drag a variable from Dictionary and drop it into the program to insert it.
  - See "Variables tab" (➔ p. 102).
- Drag a definition from the "Libraries" (➔ p. 100) and drop it into the program to insert its name
- Drag a block and drop it into the program to insert it (you can even select the block from an external text file).
- Drag a function block to the variable list to declare an instance

8.7.3.2 Auto-completion

When you type the name of a function block instance (e.g., use either as an instance or a data structure), press the point (.) after the name of the instance opens a pop-up list with the names of possible elements.

Click the relevant element and validate it with the check mark.

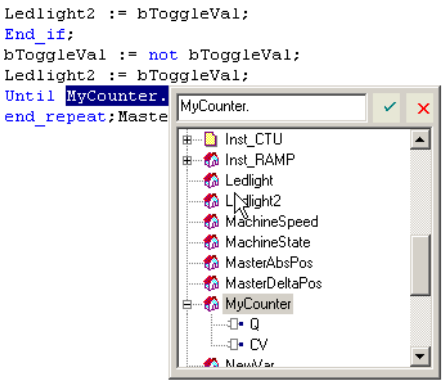
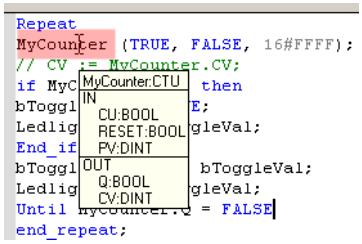


Figure 11-8: Autocompletion

8.7.3.3 Tooltip on Variable

When you leave the mouse cursor on a variable in Editors, a tooltip is shown to give you more details on the item.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

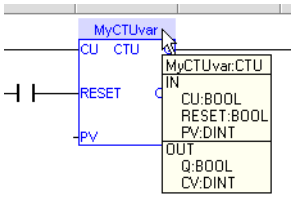


Figure 11-9: Tooltip on Variable

The header of the tooltip shows the name of the variable and its "Data Types" (→ p. 286).

8.7.3.4 Bookmarks

See "Bookmarks" (→ p. 481).

8.7.4 Select Function Blocks

All available Operators, functions and function blocks are listed in the "Libraries" (→ p. 100) toolbox.

- The list of available blocks is sorted into categories.
- The **(All)** category is used to see the complete list of available blocks.

Insert a Block in a Program

Select a block and drag-and-drop it to its position in the Editor.

TIP

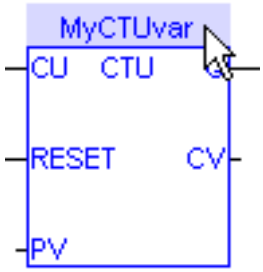
Drag a function block from the Libraries and drop it in the variable list (Dictionary) to declare a new instance. Then drag this instance from the Dictionary and drop it in the program.

8.7.5 Select Variables and Instances

Symbols of variables and instances are selected using the variable list in the Dictionary.

Selecting variables is available from all editors:

- In FBD diagrams, double-click on a variable box, an FB instance name, a contact or a coil to select the associated variable.
- In FFLD diagrams, double-click on a contact, a coil or a block input or output to select the variable. Double-click on the top of an FB rectangle to select an instance.



- When the variable editor is visible in the editor window, you can drag a variable from the list and drop it in the program to insert it.

Access a Single Bit of an Integer Variable

<variable>.<Bit number> (e.g., MachineState.7)

8.7.6 Autocompletion when Programming

When you type the name of a function block instance (use either as an instance or a data structure), pressing the point "." after the name of the instance opens a pop-up list with the names of possible elements.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Click the relevant element and validate it with the check mark.

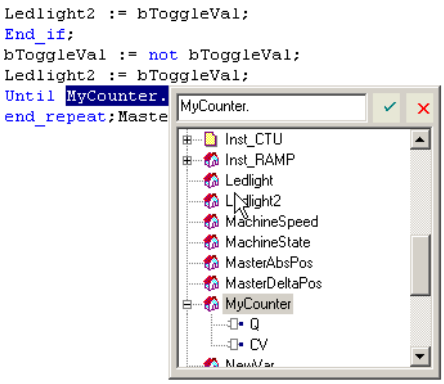


Figure 11-10: Autocompletion

See "Auto-completion of Words" (➔ p. 513).

8.7.7 Drag and Drop Programming

The editor provides you with an ideal programming environment, including drag-and-drop features:

- Drag a variable from the Dictionary and drop it into the program to insert it.
  - See "Variables tab" (➔ p. 102).
- Drag a definition from "Libraries" (➔ p. 100) and drop it into the program to insert its name.
- Drag a block and drop it into the program to insert it.
  - Select the block from an external text file.
- Drag a function block to the variable list to declare an instance.

8.7.8 Select Function Blocks

All available Operators, functions and function blocks are listed in the "Libraries" (➔ p. 100) toolbox.

The list of available blocks is sorted into categories. The “(All)” category enables you to see the complete list of available blocks.

To insert a block in a program, select it and drag-and-drop it to the desired position in the Editor.

TIP

Drag a function block from the Libraries and drop it in the variable list (Dictionary) to declare a new instance. Then drag this instance from the Dictionary and drop it in the program.

8.7.9 Select Variables and Instances

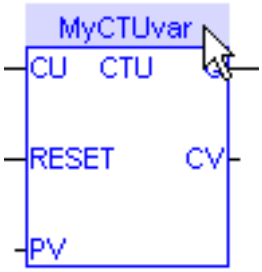
Symbols of variables and instances are selected using the variable list in the Dictionary.

Selecting variables is available from all editors:

- In FBD diagrams, double-click on a variable box, an FB instance name, a contact or a coil to select the associated variable.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- In FFLD diagrams, double-click on a contact, a coil or a block input or output to select the variable.
  - Double-click on the top of an FB rectangle to select an instance.



- When the variable editor is visible in the editor window, drag a variable from the list and drop it in the program to insert it.

Access a Single Bit of an Integer Variable

<variable>.<Bit number> (e.g. `MachineState.7`)

8.7.10 Variables Tooltips

When you leave the mouse cursor on a variable in Editors, a tooltip is displayed to give you more details on the item.

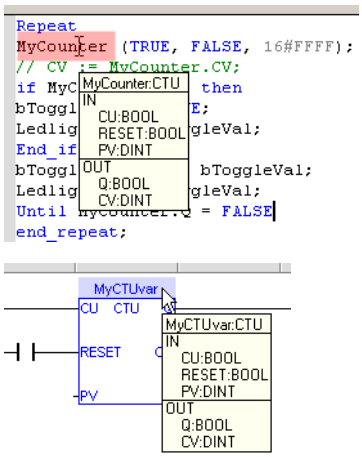


Figure 11-11: Tooltip on Variable

The header of the tooltip displays the name of the variable and its type.

8.7.11 Change Operators and Functions in FBD or FFLD Editors

Functions and operators can easily be edited and changed in the FBD or FFLD editors.

Double-clicking on a item in the editor opens a **Select** dialog box. (Figure 11-12)

This dialog box is used to change the entry to another function or operator, even of another type.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

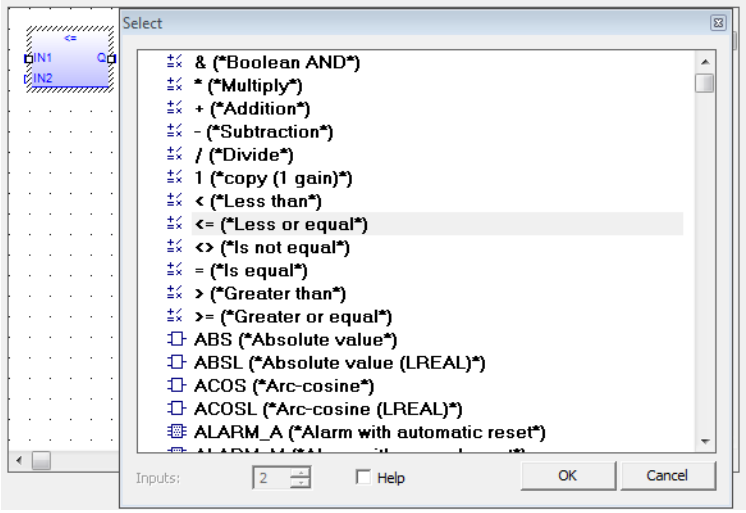


Figure 11-12: Select dialog box

- When appropriate, you can select the number of inputs.
  - This is most common for mathematical operations.
- Click the **Help** option opens the help topic for any selected operator, function, or function block.

TIP

This list is inclusive of function blocks.
A function or operator can be changed to a function block but variable assignment must be managed.

NOTE

This list includes numerous functions that are not valid for KAS.
They are either not supported or have been deprecated.

See also "Change Function Blocks in FBD or FFLD Editors" (➔ p. 486).

8.7.12 Change Function Blocks in FBD or FFLD Editors

- Double-click the name of a function block in the FBD or FFLD editors to select a different function block from a list of other FB instances. (Figure 11-13)

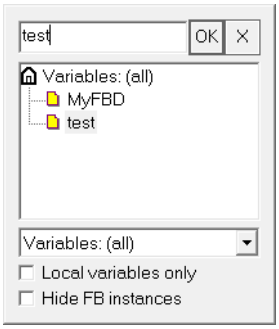


Figure 11-13: Click a function block name

- Double-clicking on the main area of a function block in the FBD or FFLD editors allows you to change to a different operator, function, or function block. (Figure 11-14)
  Be aware of any assigned variables likely to need modification.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

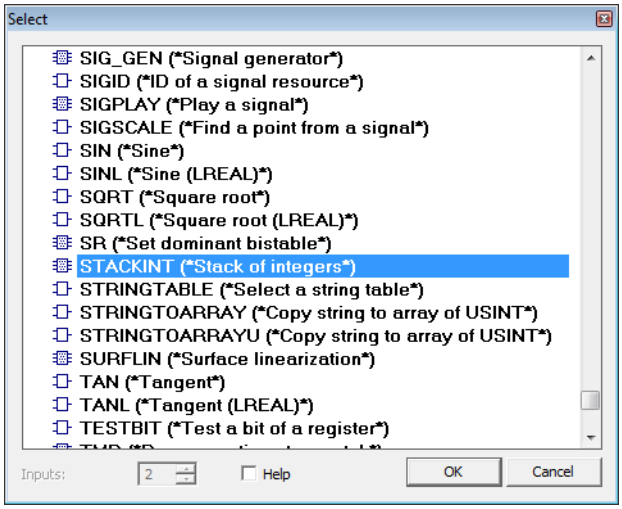


Figure 11-14: Select dialog box

NOTE

This list includes numerous functions that are not valid for KAS.
They are either not supported or have been deprecated.

See "Change Operators and Functions in FBD or FFLD Editors" (➔ p. 485).

8.7.13 Function Block Diagram (FBD) Editor

The FBD Editor is a graphic tool used to enter and manage function block diagrams according to the IEC 61131-3 standard. (Figure 11-15)

- The editor supports advanced graphic features such as drag-and-drop, object resizing, and connection line routing features to rapidly and freely arrange the elements of a diagram.
- Insert in an FBD diagram graphic elements of the FFLD (Ladder Diagram) language (e.g., contacts and coils).

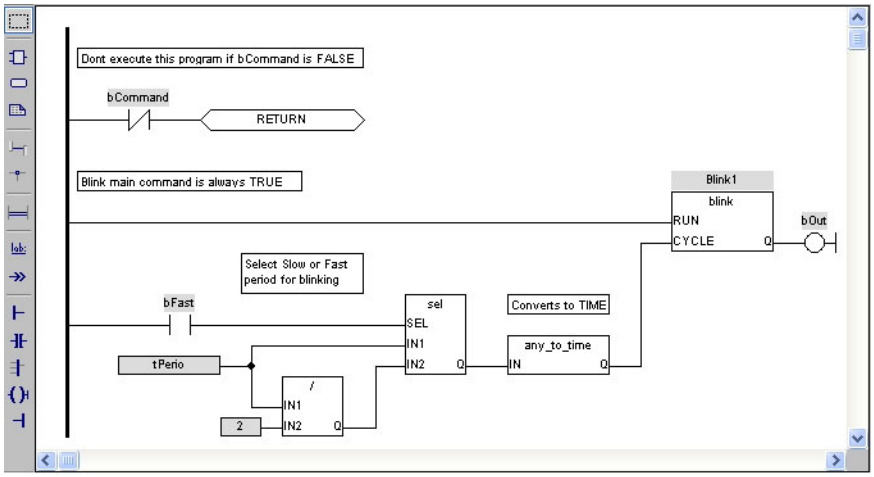


Figure 11-15: FBD main screen

FBD Diagram Components	Related Sections
"FBD Comments" (→ p. 490) "FBD Corners" (→ p. 491) "FBD Network Breaks" (→ p. 491) "FBD Variables" (→ p. 491)	"Bookmarks" (→ p. 481) "Draw FBD Connection Lines" (→ p. 492) "FBD Toolbar" (→ p. 489) "Insert FBD Objects on a Line" (→ p. 493) "Move or Copy FBD Objects" (→ p. 493) "Programs: Structure and Syntax" (→ p. 481) "Resize FBD Objects" (→ p. 494) "Select Function Blocks" (→ p. 484) "Select Variables and Instances" (→ p. 483) "View FBD Diagrams" (→ p. 495)

8.7.13.1 FFLD Components

- FFLD Contacts and Coils
- "OR" vertical rail
- Power rails

NOTE

When a contact or a coil is selected, press the **Spacebar** to change its type (e.g., normal, negated, pulse).

8.7.13.2 Toggle the Connection to Make it Negative

Boolean connections can be negative at the entry of a block.

1. Select the Boolean connection. ()

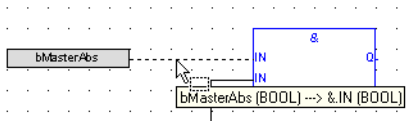


Figure 11-16: Boolean selected connection in FBD program

2. Press the **Spacebar** (a small circle is displayed)

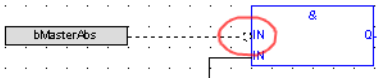


Figure 11-17: Toggle Connection in FBD Programs

8.7.13.3 Display the Execution Order

Data flow is executed from top left to bottom right (**CTRL+d** shows the execution order).

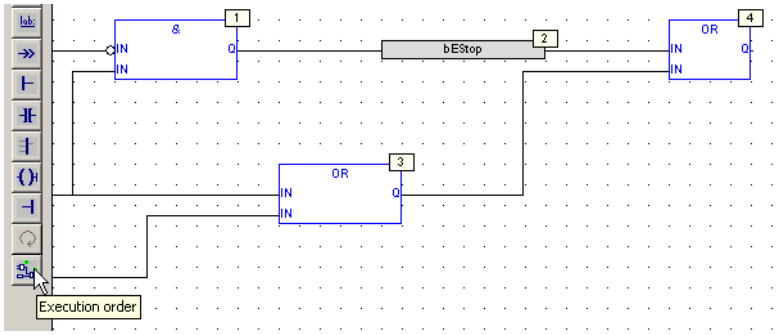


Figure 11-18: Execution Order on FBD

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.7.13.4 FBD Toolbar

The vertical toolbar on the left side of the editor contains buttons for all available editing features.

Click the applicable button before using the mouse in the graphic area.

Tool	Button	Description
Execution Order		Show the execution order of the elements in the diagram.
Insert "OR" Rail		• In this mode, the mouse is used to insert a rail that collects several Boolean data flows for an OR operation. • This is used to insert parallel contacts, as in Ladder Diagrams.
Insert Block		In this mode, the mouse is used for inserting blocks in the diagram. • Click in the diagram and drag the new block to the desired position. • The type of block inserted is the one currently selected in the list on the main toolbar.
Insert Coil		• In this mode, the mouse is used to insert a coil in the diagram (e.g., Ladder Diagrams). • It is not required that a coil be connected on its right side.
Insert Comment Text		In this mode, the mouse is used for inserting comment text areas in the diagram. • Comment texts can be entered anywhere. • Click in the diagram and drag the text block to the desired position. • The text area can be selected and resized.
Insert Connection Line		In this mode, the mouse is used to wire the input and output pins of the diagram objects. • The line must always be drawn in the direction of the data flow: from an output pin to an input pin. • The FBD editor automatically selects the best routing for the new line. • You can change the default routing by inserting corners on lines. • See "Insert Corner" (➔ p. 489). • You can drag a line from an output pin to an empty space. • In this case, the editor automatically finishes the line with a user-defined corner. • Continue drawing the connection to the designated pin and force the routing while drawing the line.
Insert Contact		In this mode, the mouse is used to insert a contact in the diagram (e.g., Ladder Diagrams).
Insert Corner		In this mode, the mouse is used for inserting a user-defined corner on a line. • Corners are used to force the routing of connection lines. • The FBD editor imposes a default routing only between two pins or user-defined corners. • Corners can be selected and moved to change the routing of existing lines.
Insert Jump		In this mode, the mouse is used to insert jump symbols in the diagram. • A jump indicates the execution must be directed to the corresponding label (having the same name as the jump symbol). • Jumps are conditional instructions. • Jumps must be linked on their left side to a Boolean data flow.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Tool	Button	Description
Insert Label		In this mode, the mouse is used for inserting a label in the diagram. • A label is used as a destination for jump symbols. • See "Insert Jump" (➔ p. 489).
Insert Left Power Rail		In this mode, the mouse is used to insert a left power rail in the diagram. • A left power rail is an element of the FFLD language. • It represents a TRUE state that used to initiate a data flow. • Power rails can be selected and resized vertically according to the desired network height.
Insert Network Break		In this mode, the mouse is used for inserting a horizontal line that acts as a break in the diagram. • Breaks have no meaning for the execution of the program. • Breaks aid in the understanding of big diagrams by splitting them into a list of networks.
Insert Right Power Rail		In this mode, the mouse is used to insert a right power rail in the diagram. • A right power rail is an element of the FFLD language. • It is used for terminating Boolean data flows. • It is not required to connect coils to power rails. • Right power rails have no meaning for the execution of the diagram.
Insert Variable		In this mode, the mouse is used for inserting variable tags. • Variable tags can be wired to the input and output pins of the blocks. • Click in the diagram and drag the new variable to the its position.
Selection		In this mode, you cannot insert any elements into the diagram. • The mouse is used to select object and lines, select tag name areas, or move or copy objects in the diagram. • Press the ESCAPE key to return to the Selection mode.
Swap Item Style		Change the text justification.

8.7.13.4.1 FBD Comments

Comment text area can be entered anywhere in an FBD diagram.

1. Select the Insert Comment Text button  in the FBD toolbar to insert a new comment area.
   In this mode, the mouse is used to insert comment text areas anywhere in the diagram.
2. Click in the diagram and drag the text block to the desired position.
3. Double-click on the comments area to enter or change the attached text.
   - When selected, comment texts can be resized.
   - See "Resize FBD Objects" (➔ p. 494).

NOTE

A hyperlink can be inserted In external files.
Only TXT and BMP extensions are allowed.
When the link is valid, the hyperlink is replaced with the file's content.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

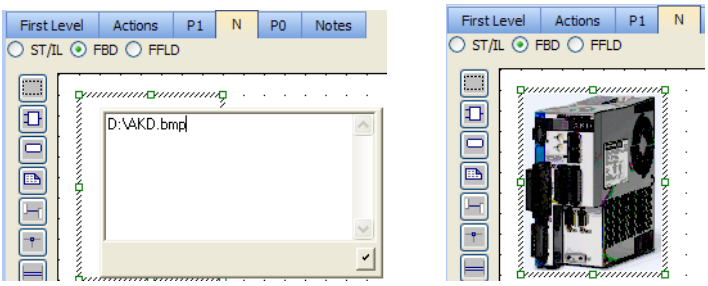


Figure 11-19: FBD Comments - Inserting Graphic

8.7.13.4.2 FBD Corners

Corners are used to force the routing of connection lines, as the FBD editor imposes a default routing only between two pins or user-defined corners.

All variable symbols and constant expressions are entered in FBD diagrams using small boxes.

Select the Insert Corner button in the FBD toolbar to insert a corner on a line:

Insert Corner
In this mode, the mouse is used to insert a user-defined corner on a line.

- Drag a new line from an output pin to an empty space.
  - In this case, the editor automatically finishes the line with a user-defined corner.
  - Continue drawing the connection to the desired pin and force the routing while you are drawing the line.
- Corners can then be selected and moved to change the routing of existing lines.

8.7.13.4.3 FBD Network Breaks

Network breaks can be entered anywhere in an FBD diagram.

- Breaks have no meaning for the execution of the program; they just help in understanding big diagrams, by splitting them into a list of networks.
- Select the Insert Network Break button in the FBD toolbar to insert a new break:

In this mode, the mouse is used for inserting a horizontal line that acts as a break in the diagram.

- The break line is drawn on the whole diagram width.
  - No other object can overlap a network break.
  - Break lines can then be selected and moved vertically to another location.
- Network breaks can also be used to browse the diagram.
  - Press the **Ctrl+Page Up** or **Ctrl+Page Down** keys to move the selection to the next or previous network break.

8.7.13.4.4 FBD OR Vertical Rail

The FBD Editor enables the drawing of FFLD rungs.

- The OR rail can be inserted on a rung in order to connect parallel contacts together.
- In the FBD toolbar , click the **Insert OR Rail** button ( ) to insert a new OR rail.
  - In this mode, the mouse is used to insert a rail that collects several Boolean data flows for an OR operation to insert parallel contacts (e.g., Ladder Diagrams).
- The OR rail has exactly the same meaning as an OR block regarding the execution of the diagram.

8.7.13.4.5 FBD Variables

All variable symbols and constant expressions are entered in FBD diagrams using small boxes.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. Click this button in the FBD toolbar to insert a variable tag:
   
   - In this mode, the mouse is used for inserting variable tags.
   - Click in the diagram and drag the new variable to the desired position.
2. Double-click a variable tag to open the variable selection box.
   See "Select FBD Variables and Instances" (➔ p. 495).
3. Either select the symbol of a variable or enter a constant expression.
   Variables tags must be linked to other objects (e.g., block inputs and outputs) using connection lines.
4. Resize a variable box vertically to display, together with the variable name, its tag (short comment text), and its description text.
   The variable name always appears at the bottom of the rectangle:



Description
Tag
Name

See "Variables tab" (➔ p. 102).

8.7.13.5 Draw FBD Connection Lines

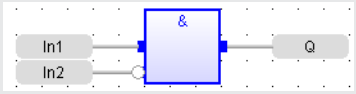


Click this button before inserting a new line.

NOTE

The editor is used to terminate a connection line with a Boolean negation represented by a small circle:

```
(* use of a negated link: Q is IN1 AND NOT IN2 *)
```



To set or remove the Boolean negation, select the line and press the <Spacebar>.

- Connection lines must always be drawn in the direction of the data flow (i.e., from an output pin to an input pin).
- The FBD editor automatically selects the best routing for the new line.
- Connection lines indicate a data flow between these possible objects:

NOTE

The line color is red when the two linked items are not the same type.

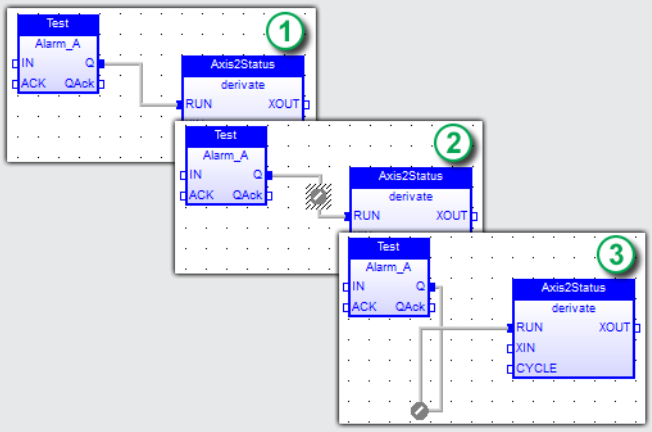
Tool	Button	Description
Block		See the help for the block for the description of its input and output pins and the expected data types for the coherence of the diagram.
Coil		• A coil must be connected to a Boolean data flow from it's left-hand side. • It is not mandatory that a coil be connected on its right-hand side.
Contact		A contact must be connected to a Boolean data flow on both its left- and right-hand side.
Jump		A jump must be connected to a Boolean data flow from its left-hand side.
Left power rail		• Left power rails represent a TRUE state. • It can be connected to an unlimited number of objects on their right-hand side.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Tool	Button	Description
OR rail		This type of rail collects several Boolean data flows for an "OR" operation, in order to insert parallel contacts (e.g., Ladder Diagrams). • An OR may have several connections on both its left- and right-hand side. • All connected data flows must be Boolean.
Right power rail		A right power rail is an element of the FFLD language, and is commonly used for terminating Boolean data flows. • It has an unlimited number of connections on its left-hand side. • It is not mandatory to connect coils to power rails.
Variable		• A variable can be connected on its right-hand side (to initiate a flow) or on their left-hand side to force the variable, if it is not read-only. • The flow must fit the data type of the variable.

TIP

Connection lines automatically move and follow FBD elements (1).
You can also manually specify the corners.
Double-click the line to show a handle (2) which can be repositioned (3).



8.7.13.6 Insert FBD Objects on a Line

The FBD editor is used to insert an object on an existing line and automatically connect it to the line.

- This feature is available for all objects having one input pin and one output pin (e.g., variable boxes, contacts, coils).
- This feature is mainly useful when entering elements of Ladder Diagrams.
  - Draw a horizontal line between left and right power rails: it is the rung.
  - Then insert contacts and coils on the line to build the FFLD rung.

8.7.13.7 Move or Copy FBD Objects



Click the Selection Mode button or press <Esc> before selecting objects.

The FBD Editor is used to drag-and-drop for moving or copying objects.

- To move objects, select and drag them to the desired position.
- To copy objects, press <Ctrl> while dragging.
  - It is also possible to drag pieces of diagrams from one program to another if both are open and visible on the screen.
- At any time while dragging objects, press <Esc> to cancel the operation.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Use the Copy / Cut / Paste commands from the Edit menu.
  - When you run the Paste command, the editor changes into Paste mode, with a special mouse cursor.
  - Click in the diagram and move the mouse cursor to the desired position for inserting pasted objects.

8.7.13.7.1 Move Objects

When graphic objects are selected, press these keys to move them in the diagram:

Keys	Description
<Shift + Up>	Move to the top.
<Shift + Down>	Move to the bottom.
<Shift + Left>	Move to left.
<Shift + Right>	Move to right.

8.7.13.7.2 Extend Object Selection

When an object is selected, press these keys to extend the selection:

Keys	Description
<Shift + Control + Home>	Extend to the top: select all objects before the selected object.
<Shift + Control + End>	Extend to the bottom: select all objects after the selected object.

8.7.13.7.3.1 Insert or Delete Space in the Diagram

1. Select an object.
2. Press <Shift+Ctrl+End> to extend the selection.
3. Move selected objects up or down.

8.7.13.7.4 Auto-alignment

When objects are selected, press these keys to automatically align them:

Keys	Description
<Control + Up>	To the top.
<Control + Down>	To the bottom.
<Control + Left>	To left.
<Control + Right>	To right.

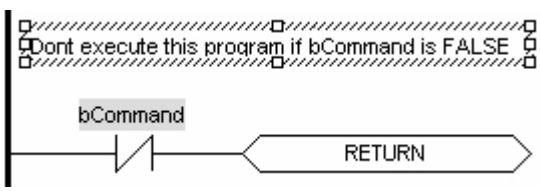
8.7.13.8 Resize FBD Objects

When an object is selected, small square boxes indicate how to resize it with the mouse.



Click this button or press <Esc> before selecting objects.

Click the small square boxes to resize the object in the desired direction.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Not all objects can be resized.

This table indicates possible operations:

Operation	Description
Block	Horizontally
Comment area	In all directions.
Labels and jumps	Horizontally
OR rail	Vertically
Power rails	Vertically
Variable	Horizontally and vertically (*)

(*) Resizing a variable box vertically enables you to display together with the variable name its tag (short comment text) and its description text.

The variable name is always shown at the bottom of the rectangle:

description
tag
name

8.7.13.9 Select FBD Variables and Instances

 Click this button or press <Esc> before any selection.

To select the name of the declared variable to attach to a graphic symbol, you must be in Selection mode.

Double-click on the tag-name gray area.

These types of object must be linked to valid symbols:

Tool	Button	Description
Block		For a function block, you must specify the name of a valid declared instance of the corresponding type.
Coil		Must be attached to a declared Boolean variable.
Contact		Must be attached to a declared Boolean variable.
Jump		Must have the same name as its destination label.
Label		• Must have a name. • The name must be unique within the diagram.
Variable		• Must be attached to a declared variable. • Alternatively, a variable box may contain the text of a valid constant expression.

Symbols of variables and instances are selected using a variable list, that can be used as the variable editor.

1. Enter a symbol or constant expression in the **Edit** box and click **OK**.
2. Select a name in the list of declared objects or
   Click the **Create** button to declare a new variable.

See Select Variables and Instances.

8.7.13.10 View FBD Diagrams

The diagram is entered in a logical grid.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- All objects are snapped to the grid.
- The (x,y) coordinates of the mouse cursor appear in the status bar.
  - This helps locate errors detected by the compiler or to align objects in the diagram.
- Press <CTRL+ scrolling mouse wheel> to zoom in or out of the edited diagram or press the <+> and <-> keys of the numeric keypad.

8.7.14 Free Form Ladder Diagram (FFLD) Editor

The FFLD Editor is a graphic tool used to enter and manage ladder diagrams according to the IEC 61131-3 standard.

This editor enables:

- Free drawing and arrangement of FFLD items.
- Supports these graphic features:
  - "Change Ladder Diagram Size" (→ p. 496)
  - "Color-coded Functions" (→ p. 496)
  - "Coordinate-based Positions" (→ p. 497)
  - "FFLD Move and Copy Items" (→ p. 504)
  - "Select the Entire Ladder Diagram" (→ p. 497)
  - "Split Window Capability" (→ p. 498)

See Also

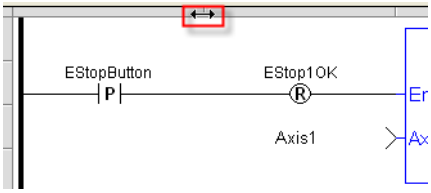
- "FFLD Networks" (→ p. 500)
- "FFLD Runtime" (→ p. 501)

8.7.14.1 Advanced Graphic Features

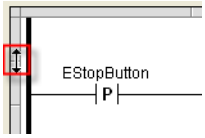
8.7.14.1.1 Change Ladder Diagram Size

Diagram Size

Drag the column separator to increase or decrease the size of the columns.



Drag the row separator to increase or decrease the size of the rows as well as the texts.

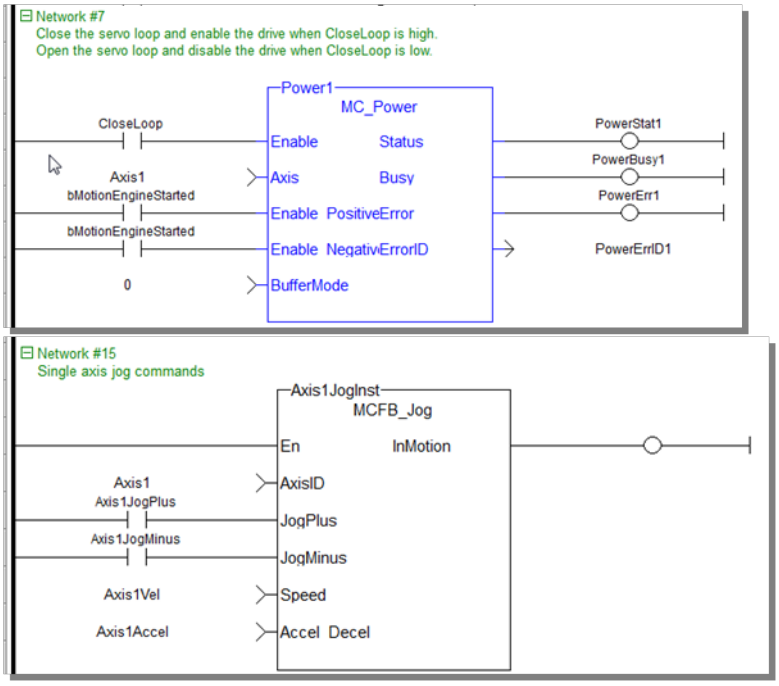


8.7.14.1.2 Color-coded Functions

Color Coded Function

- Functions and Function Blocks are color-coded as blue.
- Sub-programs and Kollmorgen UDFBs are black.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



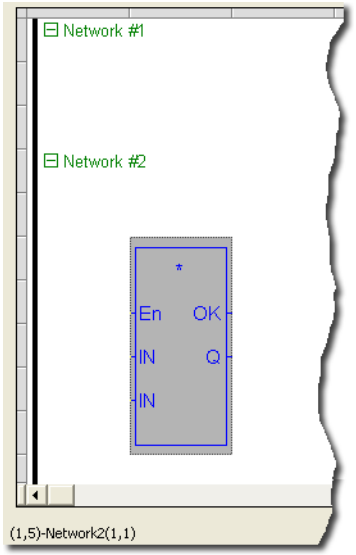
8.7.14.1.3 Coordinate-based Positions

Coordinate-based Position

The coordinates for a selected function are displayed at the bottom-left corner of the editor.

The coordinates are shown in this format:

(absolute coordinates)-(network number).(relative coordinates) in (column, row)



NOTE

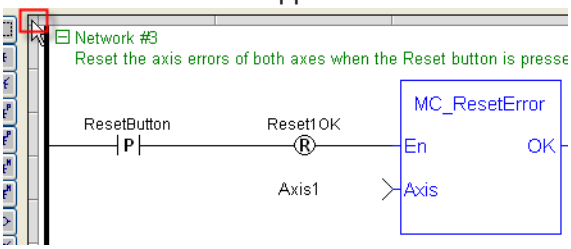
Relative coordinates are measured from below the Network label.

8.7.14.1.4 Select the Entire Ladder Diagram

Select All

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Click the border in the upper left corner to select the entire ladder.

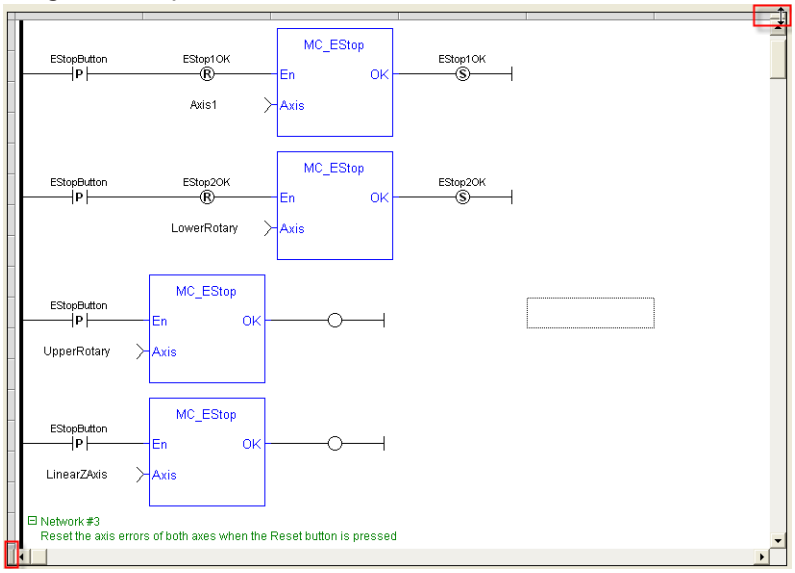


8.7.14.1.5 Split Window Capability

Split Window

Allows multiple views of the same ladder program to be displayed simultaneously.

Drag the two splitters located in the vertical and horizontal scroll bars to organize the windows.



FFLD Diagram Components	Related Sections
• "FFLD Contacts and Coils" (➔ p. 502) • "FFLD Data In / Data Out" (➔ p. 502) • "FFLD Function Blocks" (➔ p. 502) • "FFLD Jumps and RETURN" (➔ p. 503) • "FFLD Networks" (➔ p. 500) • "FFLD Power Rail and Lines" (➔ p. 501)	• "FFLD Move and Copy Items" (➔ p. 504) • "FFLD Runtime" (➔ p. 501) • "FFLD Selection Grid" (➔ p. 503) • "FFLD Toolbar and Right-click Menu" (➔ p. 498)

TIP

When a contact or a coil is selected, press the **Spacebar** to change its type (e.g., normal, negated, pulse).

8.7.14.2 FFLD Toolbar and Right-click Menu

This section describes the "Toolbar" (➔ p. 498) buttons and "Right-click Menu" (➔ p. 500) of the FFLD Editor.

See (➔ # 476, FFLD Editor Keyboard Shortcuts) about the FFLD accelerator keys.

8.7.14.2.1 Toolbar

The vertical toolbar on the left side of the FFLD Editor uses buttons to insert items in the diagrams.

Items are inserted at the current position in the diagram.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Button	Shortcut	Description
		Mode selection
	Ctrl+Shift+O	Insert a contact to the destination cell.
	Ctrl+Shift+C	Insert an inverted contact to the destination cell.
	Ctrl+Shift+P	Insert a Pulse contact to the destination cell.
	Ctrl+Shift+I	Insert an inverted Pulse contact to the destination cell.
	Ctrl+Shift+N	Insert a N contact to the destination cell.
	Ctrl+Shift+A	Insert an inverted N contact to the destination cell.
	Ctrl+Shift+E	Insert a coil to the destination cell.
	Ctrl+Shift+D	Insert an inverted coil to the destination cell.
	Ctrl+Shift+S	Insert a set coil to the destination cell.
	Ctrl+Shift+R	Insert a reset coil to the destination cell.
	Ctrl+Shift+K	Insert a positive coil to the destination cell.
	Ctrl+Shift+L	Insert a negative coil to the destination cell.
	Ctrl+Shift+H	Trace a horizontal line to the destination cell.
	Ctrl+Shift+V	Trace a vertical line to the destination cell.
	Ctrl+Shift+B	Trace a vertical and horizontal line to the destination cell.
		Toggle trace mode. Click and move the mouse to draw a line spanning on several adjacent cells.
	Shift+Insert	Insert a network.
	Ctrl+Shift+J	Insert a jump.
	Ctrl+Shift+T	Insert a return.
	Ctrl+Shift+F	Insert a data in.
	F8	Insert a function block.
	Ctrl+Shift+Q	Insert a data out.
	Spacebar	Swap item style of the current cell for a contact or coil.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Button	Shortcut	Description
		Define a network label, pragma, and/or comment.
	Shift+C	Define a comment in a cell. Click the button, then select the cell to add a comment to. See "Right-click Menu" (→ p. 500) about cell comments.

8.7.14.2.2 Right-click Menu

Right-click in the FFLD workspace to use these commands:

- Insert Comment
  - Comments can be added to cells.
  - If the text is long, it is shown beyond the bounds of the cell (to the right, and below) until another item is found. (Figure 11-20)

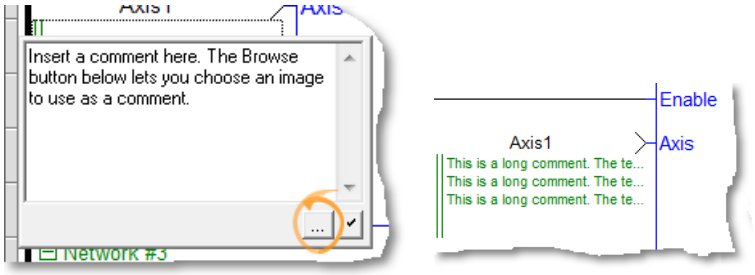


Figure 11-20: Example: Long comment text

- Delete Cell
- Delete Network
- Delete Row
- Insert Network
- Insert Row

8.7.14.3 FFLD Networks

A program is entered as a list of independent networks.

- Networks are executed sequentially from the top to the bottom.
- The head of a network is drawn on a full row in the editor, grouping this information:
  - The number of the network (from 1 to N)
  - (Optional) A label name used as a target for jump operations.
    - See "FFLD Jumps and RETURN" (→ p. 503).
  - (Optional) A directive for "Conditional Compiling" (→ p. 259).
  - (Optional) A multiple line description (comment).

8.7.14.3.1 Header Row

No item can be put on a network header row. (Figure 11-21)

- No line can go through it.
- The end of a program is marked with a special "End of module" row.
- Nothing can be inserted after this row.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

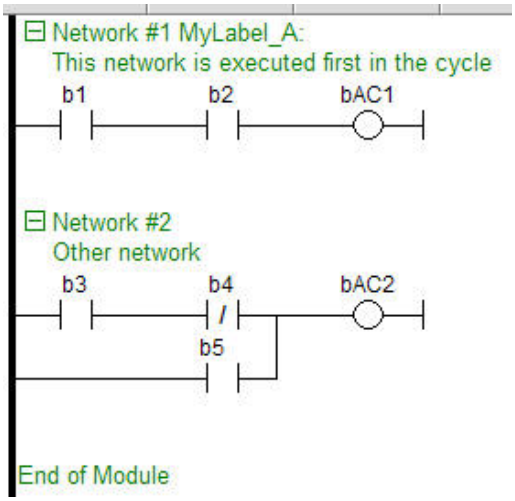


Figure 11-21: Example: Header Row

- Double-click the header of a network to enter its label, directive (sometimes called pragma) and description.
- Network headers are green.
  - They became blue when a directive is defined.
  - See (➔ # 259, Conditional Compiling).

8.7.14.3.2 New Networks

New networks can be inserted on empty rows.

- When a network is selected, pressing <DEL> merges its content with the previous network.
  - When the first network is selected, pressing <DEL> removes the network and all its contents.
- There cannot be two networks having the same label in a program.
  - If this situation occurs in the case of a copy operation, you are prompted to either:
    - Specify another label name for the new network.
    - Remove the label on the new network.

Collapse/expand a network with the minus/plus sign located next to the Network number in the header. (Figure 11-22)

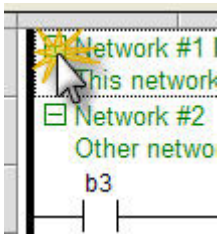


Figure 11-22: Example: Collapse Header row

8.7.14.4 FFLD Runtime

When your application is running, you can force and lock a variable or a contact directly in the editor with a double-click operation.

See "Force a Variable" (➔ p. 279).

NOTE

In FFLD, when a function, function block, or UDFB is not connected on the left, remove it at compiling time.

8.7.14.5 FFLD Power Rail and Lines

Vertical power rails are used in the FFLD Language to represent the limits of a rung.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The power rail on the left represents the TRUE value and initiates the rung state.
  - Any object connected to this rail is always powered.
- Horizontal lines always represent a data flow from the left to the right.
- If a vertical line has several items connected on the left, then it represents an OR / ORN operation.
- You can insert a segment of horizontal line at any location to freely draw flow lines.
  - The vertical line button is used to set or remove (toggle) a segment of vertical line on the right of the selected cell.

8.7.14.6 FFLD Contacts and Coils

This is a list of the FFLD contact and coil types:

Contacts	Coils
Negative Transition -|N|-	Reset (Unlatch) -(R)-
Normally Closed -|/|-	De-energize -(/)-
Normally closed negative transition -|/N|-	Negative transition sensing coil -(N)-
Normally closed positive transition -|/P|-	Positive transition sensing coil -(P)-
Normally Open -| |-	Energize -( )-
Positive Transition -|P|-	Set (Latch) -(S)-

See Also

- FFLD Coils
- FFLD Contacts

8.7.14.7 FFLD Function Blocks

Functions and function blocks can be used in FFLD diagrams.

- Blocks are always connected to the flow line (powered) by their first input and first output.
  - If the first input of a block is not Boolean, a special input called EN is added.
  - This means the block is not executed if the input flow is FALSE.
  - If the first output is not Boolean, a special output called OK is added.
    - The special OK output always has the same state as the first input (the flow).
- In a function block, the instance of the block must be specified and is shown on the top of the block.
  - Double-click the top of the block to select the instance.
  - Double-click elsewhere in the block to change its type.
- Boolean inputs and outputs of blocks can be directly linked to contacts and coils.
  - Block inputs and outputs can be specified using specific "FFLD Data In / Data Out" (→ p. 502) items.

NOTE

Function and function blocks cannot be put in column 1 of the grid.
This would not make sense because data inputs require a column.

- You cannot change a function block after it has been inserted.
- When a function is not connected on the right, it is ignored (removed at compiling time).
  - It is a case for functions only - **not** for function blocks.
  - A function is just part of an expression (same as a contact) and is intended to provide a result.
  - With FFLD, KAS accepts that the output is not connected because it accepts pending dead expressions to be removed at compiling time (same as contacts with no coil or FBs after).

TIP

If you want another function block, you first have to select it in the Libraries toolbox before inserting it.

8.7.14.8 FFLD Data In / Data Out

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The data in and data out items are used to either:

- Initiate a flow (line) with the value of a variable.
- Force a variable on output with the value of a flow:

```
VarIn>- .......... ->VarOut
```

When used with a block, the data in and data out items can be put close to the block, without any line in-between to connect a variable to an input or output of the block.

Example

The **@** symbol in front of the V variable indicates it is used for input and output. (Figure 11-23)

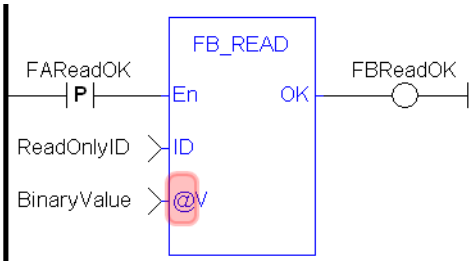


Figure 11-23: Input / Output @ Symbol

8.7.14.9 FFLD Jumps and RETURN

A jump to a label branches the execution of the program after the specified label.

- In the FFLD Language, the >> symbol (followed by the target label name) is used as a coil at the end of a rung. (Figure 11-24)

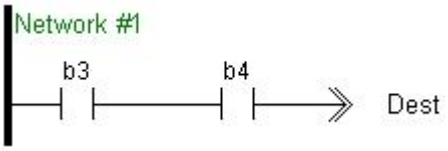


Figure 11-24: Example: >> Symbol

- The jump is performed only if the rung state on input is TRUE.
  - The destination label must be specified on a network of the same program.
  - To specify the destination, double-click the cell to display a drop-down menu that lists all the available labels.

NOTE

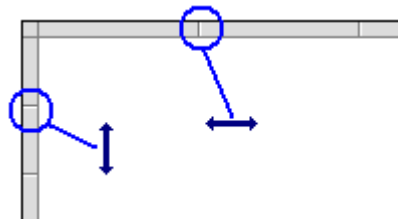
The special <RETURN> destination specifies a jump to the end of the program.

8.7.14.10 FFLD Selection Grid

The diagram is entered in a logical grid, and all objects are snapped to the grid.

- Use the **View** menu commands to see hiding grid points.
  - This helps locate errors detected by the compiler or aligning objects in the diagram.
- Zoom in or out of the edited diagram.
  - See "FFLD Toolbar and Right-click Menu" (➔ p. 498).
- Drag the separation lines in vertical and horizontal rulers to freely resize the cells of the grid:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



NOTE

If a split window is in use, the zoom applies only to the currently selected split window (each split window can be zoomed to different levels).

- The current position in the grid is always highlighted by a dotted cell.
  - Its coordinates (row, column) are displayed at the bottom left-hand corner of the editor.
- Click the current position, the cell is drawn as gray, meaning that it can be dragged somewhere else in the diagram.
  - Select multiple cells with the mouse, or use the arrows of the keyboard with the <SHIFT> key pressed.
- Click the power rail (gray ruler at the left border) to select a full row.

These commands are available from the keyboard:

Keys	Description
Arrows	Move focus cell or scroll through ladder.
Ctrl-Shift-F2	Go to previous bookmark.
Ctrl + A	Selects the whole network. If pressed again, selects the whole program.
Ctrl + F	Performs a Search and Replace (+ add hyperlink on the topic) within the whole program.
Ctrl + Home/End twice	Moves the cursor to the beginning or the end of the program.
Ctrl + Page Up Down	Moves the cursor to previous or next network header.
End	Moves the cursor to the end of the line. If pressed again, moves the cursor to the end of the network.
Esc / Shift-Esc	Close the rename widget.
Home	Moves the cursor to the left of the line. If pressed again, moves the cursor to the head of the network.
Page Up / Down	Scroll 1 page.
Return	Equivalent to a double-click.
Shift-Arrow	Multi-select cells.
Shift-Page Up / Down	Selection page up or down.
Shift-Tab	Move focus cell left.
Space	Change contact or coil.
Tab	Move focus cell right.

8.7.14.11 FFLD Move and Copy Items

When you click on the current position, the cell is drawn as gray, meaning that it can be copied or moved.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Click the selection again to drag it with the mouse. (Figure 11-25)



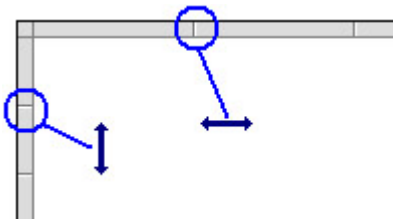
Figure 11-25: Example: Click and Drag

- Dragging the selected items moves them to the specified location.
  - If you press the <Ctrl> key while dragging, then items are copied.
    - For shortcuts, see "FFLD Toolbar and Right-click Menu" (→ p. 498).
- To move a function block, you must select it entirely.
- If you move or copy items on a non-empty area, you are prompted to confirm the overwriting of items in the area.
- When you move or copy items only on a network header, the network is automatically moved to make the required extra space for moved items.
- The Copy / Cut / Paste commands can be used as an alternative to drag-and-drop.
- A rectangular selection within the diagram cannot cross a network header (i.e., all selected items must be within the same network).
  - To select a complete network or more, you must select complete rows.
  - To do this, move the caret to the left border or click on the left-hand side ruler (gray).

8.7.14.12 View FFLD Diagrams

The diagram is entered in a logical grid. All objects are snapped to the grid.

- The (x,y) coordinates of the mouse cursor are displayed in the status bar.
  - This helps to locate compiler detected errors or to align objects in the diagram.
- Zoom in or out of the edited diagram using either:
  - A <Ctrl + mouse-wheel> operation.
  - Press the <+> and <-> keys of the numerical keypad.
- Drag the separation lines in vertical and horizontal rulers to freely resize the cells of the grid:



- The FFLD editor adjusts the size of the font according to the zoom ratio so that the name of variables associated with contacts and coils are always visible.
- If cells have sufficient height, variable names are completed with other these pieces of information about the variable:
  - Tag (short description).
  - Description text.
  - I/O name (%...) if the variable has a user-defined name.

8.7.15 Sequential Function Chart (SFC) Editor

The SFC Editor is a graphic tool used to enter and manage sequential function charts according to the IEC 61131-3 standard.

- The editor supports graphic features (e.g., drag-and-drop) to rapidly arrange the elements of the diagram. (Figure 11-26)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- It supports automatic chart formatting when inserting or deleting items using quick inputs with the keyboard.

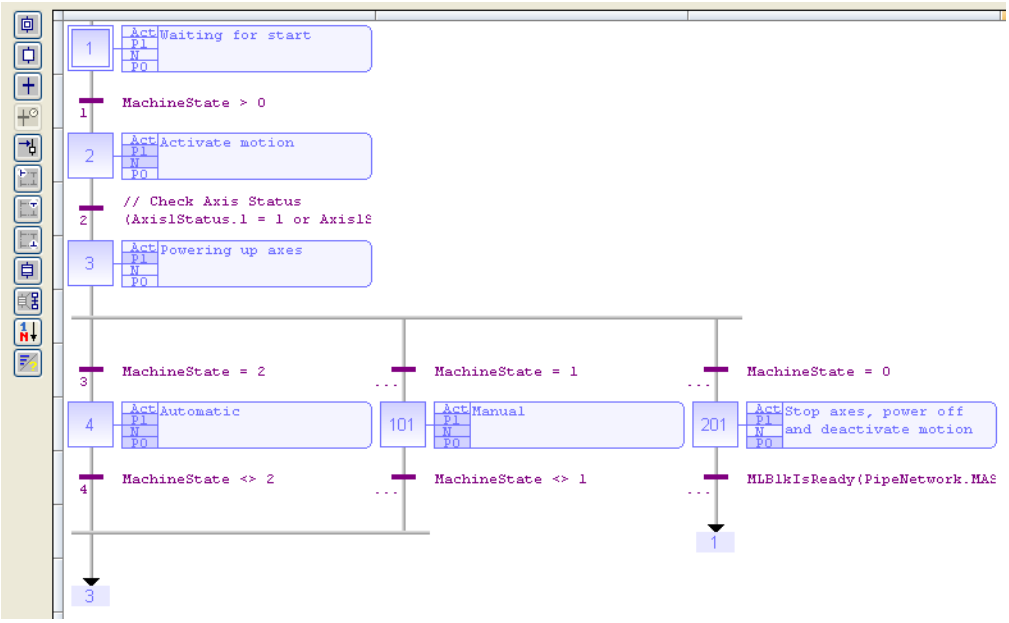


Figure 11-26: Example: SFC chart

NOTE

For each step, the cells referring to P1, N, and P0 actions are colored when they are defined.

TIP

- Select a step, transition, or jump and press **Ctrl+ENTER** to change its number.
- Press the **Spacebar** on the main corner (on the left) of a ulergence or convergence, to set either double or single horizontal line style.

8.7.15.1 SFC Toolbar

The vertical toolbar on the left side of the editor contains buttons for inserting items in the chart.

- Items are always inserted before the selected item.
- The chart is automatically re-arranged when a new item is inserted.

Use these keyboard commands when an item is selected:

1. **ENTER**: Edit Level 2 of a step or transition.
- **Ctrl+ENTER**: Change the number of a step, transition, or jump.

Toolbar Button	Description
Initial step (I)	Inserts an initial step.
Step (S)	Inserts a step.
Transition (T)	Inserts a transition.
Set Timer on Transition	Sets a timer on a transition. **Example**: 5 on step number 4 leads to a transition with GS4.T>T#5ms.
Jump (J)	Inserts a jump to a step.
Divergence (X)	Inserts the main (left side) corner of a divergence or convergence.
Divergence (D)	Inserts a divergence corner.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Toolbar Button	Description
Convergence (C)	Inserts a convergence corner.
Insert Comment	Inserts a comment.
Change Display	Swaps between possible overviews of level 2 in the level 1 chart: • Show all. • Show the code of actions and conditions. • Show notes attached to steps and transitions.
Swap Item Style	Allows changes to the item style. Changes either: • A step to an initial step (and the reverse) . • An OR divergence / convergence to an AND divergence / convergence (and the reverse).
Renumber	Renumbers all steps and transitions. Previously, the window showing the content of a step or a transition must be closed.
Edit Reference	Edits the number of a step, transition or jump. Previously, the window showing the content of a step or a transition must be closed.
Show one qualifier in level 2	Shows one level of the edition in the level 2 of the SFC Editor. **Example**: Shows only the P1 of a step.
Show/Hide split window in level 2	Shows two levels of the edition in the level 2 of the SFC Editor. **Example**: Show P1 and Notes of a step.
Show all qualifiers in level 2	Shows every level of the level 2 on the SFC Editor.

8.7.15.1.1 Edition window

Open the Edition window using a click on ENTER, when an item is selected.

- The level 2 of a step or a transition can be edited in this window.
- The different levels of editions in the level 2 of the SFC Editor can be selected using the buttons at the top of the Edition window.

Toolbar Button	Description
Action	Shows/hides the action box in the SFC level 2.
P1	Shows/hides the P1 box in the SFC level 2.
N	Shows/hides the N box in the SFC level 2.
0	Shows/hides the P0 box in the SFC level 2.
Notes	Shows/hides the Notes box in the SFC level 2.
Close level 2	Closes the SFC level 2.

8.7.15.2 Draw SFC Divergences

When using the SFC Editor, you need to place items in the grid.

The editor calculates and draws lines automatically to link the steps and transitions and jumps to your place in the chart.

The same method is used for drawing divergences:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

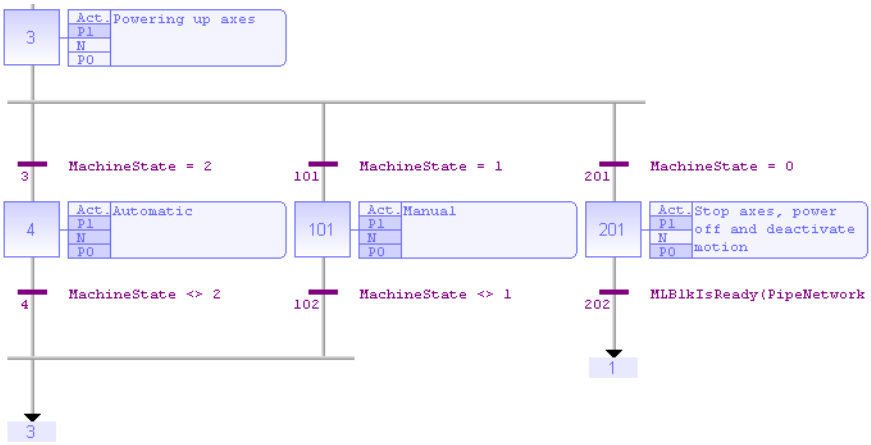
- Place the corners that identify divergences, convergences, and branches.
- The editor takes care of drawing vertical and horizontal lines.

Use these buttons in the SFC toolbar:

Button	Description
	Insert the main (left side) corner of a divergence or convergence.
	Insert a divergence corner.
	Insert a convergence corner.

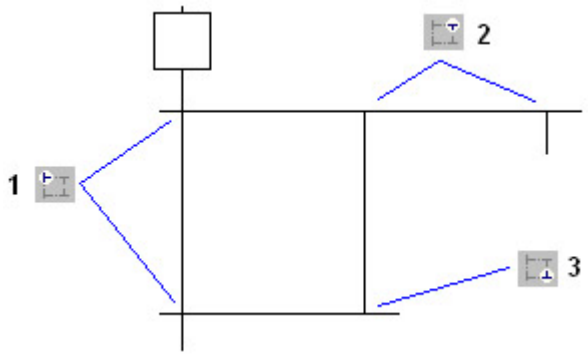
IMPORTANT

Divergences are always drawn from left to right.
The first branch, on the left, contains the corners that identify the divergence.
It must be aligned with the preceding step or transition:



Procedure

1. Insert the main corner (on the left-hand side branch) of the divergence and the convergence.
2. Insert corners at the top of each branch (divergence).
3. Insert corners at the bottom of the branches where a divergence is required.



8.7.15.2.1 Simple or Double Divergence Lines

You can change the drawing of a divergence or convergence horizontal line, for drawing simple or double lines according to the SFC definition.

Move the selection on the main corner (on the left) and press the **Spacebar**.

8.7.15.3 Enter Actions of a Step

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Actions and notes attached to a step (level 2) are entered in a separate window.

See:

- Actions in an SFC Step
- "Enter Notes for Steps and Transitions" (→ p. 510)

To open the level 2 editing window of a step or transition, double-click its symbol in the chart or select it and press <Enter>.

The level 2 editing window has these views for entering different types of level 2 information:

- Simple actions entered as text.
- P1 actions than can be programmed in ST/IL text, FFLD, or FBD.
- N actions than can be programmed in ST/IL text, FFLD, or FBD.
- P0 actions than can be programmed in ST/IL text, FFLD, or FBD.
- Text notes.

Use the tab buttons in the level 2 editing window to select a view:

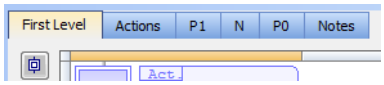


Figure 11-27: Editing window - Tabs

When editing P1, N, or P0 actions, use the radio buttons to select the programming language.

- The first view (Action) contains all simple actions for controlling a Boolean variable or a child SFC chart.
- It is possible to directly enter action blocks programmed in ST together with other actions in this view.
- Use this syntax for entering ST action blocks in the first pane:
  - Where `qualifier` is P1, N, or P0.

```
ACTION ( qualifier ) :
     statements...
END_ACTION;
```

8.7.15.4 Enter the Condition of a Transition

The conditions and notes attached to a transition (level 2) are entered in a separate window.

See:

- Condition of an SFC Transition
- "Enter Notes for Steps and Transitions" (→ p. 510)

To open the level 2 editing window of a step or transition, double-click its symbol in the chart or select it and press <Enter>.

The level 2 editing window has two views for entering different types of level 2 information:

- Condition programmed in ST / IL text or FFLD.
- Text notes.

Use the tab buttons in the level 2 editing window for selecting a view:



When editing the condition, use the **Edit / Set Language** menu command to select the programming language.

- This command is not available if the condition is not empty.
- FBD cannot be used to program a condition.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.7.15.5 Enter Notes for Steps and Transitions

The SFC editor supports the definition of text notes for each step and transition.

The notes are entered in the level 2 editing window of steps and transitions.

See these topics about the level 2 editing window:

- "Enter Actions of a Step" (➔ p. 508)
- "Enter the Condition of a Transition" (➔ p. 509)

Notes can be displayed in the chart.

- Notes have no meaning for the execution of the chart.
- Entering notes for steps and transitions enhances the auto-documentation of your programs.
- It provides a way to write and exchange specifications of an SFC program before actions and conditions are programmed.

8.7.15.6 Move or Copy SFC Charts

The SFCEditor fully supports drag-and-drop for moving or copying items.

Move an Item

To move an item, select the item and drag it to its new position.

Copy an Item

To copy an item, select the item, press the **Ctrl** key, and drag it to its new position.

- It is also possible to drag pieces of a chart from one program to another if both are open and visible on the screen.
- At any moment, while dragging items, press **Esc** to cancel the operation.
- Alternatively, use the Copy / Cut / Paste commands from the Edit menu.
  - The Paste action is performed at the current position.

8.7.15.7 Re-number Steps and Transitions

Each step or transition is identified by a number.

- A jump to a step is identified by the number of the destination step.
  - The SFC Editor allocates a new number to each step or transition inserted in the chart.

Select a step, transition, or jump and press **Ctrl+ENTER** to change its number.

- It is not possible to change the number of a step or a transition if its level 2 is currently open for editing.
  - The number is used for identifying the step or transition in the level 2 editing window.

In compiler reports:

- A step is identified by its number prefixed by GS.
- A transition is identified by its number prefixed by GT.

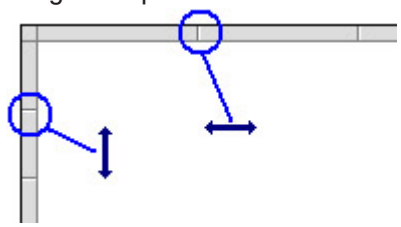
8.7.15.8 View SFC Charts

The chart is entered in a logical grid. All objects are snapped to the grid.

- The (x,y) coordinates of the mouse cursor are in the status bar.
  - This helps locate errors detected by the compiler or to align objects in the chart.
- Zoom in or out of the edited diagram using the Ctrl + mouse-wheel operation.
- Press the [+] and [-] keys of the numerical keypad to zoom the diagram in or out.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Drag the separation lines in vertical and horizontal rulers to resize the cells of the grid:



The SFC Editor adjusts the size of the font according to the zoom ratio.

When a cell is wide enough, a text is displayed with the contents of the step or transition (level 2).

Icon	Description
	Swap between possible overviews of level 2 in the level 1 chart: • Display code of actions and conditions. • Display notes attached to steps and transitions.

8.7.16 Structured Text (ST) / Instruction List (IL) Editor

The ST / IL editor is a language-sensitive text editor dedicated to IEC 61131-3 languages.

The editor supports advanced graphic features such as drag-and-drop, syntax coloring, and active tooltips for efficient input and test of programs in ST and IL. (Figure 11-28)

```
Blinker (TRUE, t#2s);
Trigger (Blinker.Q);

bSig := Trigger.Q;

Counter (
    bSig,           (* blinking input *)
    not bCommand,   (* reset the counter if command *)
    255
    );
iValue := Counter.CV;
```

Figure 11-28: Code example

See Also

- "Auto-completion of Words" (➔ p. 513)
- "Bookmarks" (➔ p. 481)
- "Read the Output of an MC Function Block in ST" (➔ p. 514)
- "Select Variables and Instances" (➔ p. 483)
- Select function blocks
- "ST / IL Drag-and-Drop" (➔ p. 514)
- "ST / IL Language Selection" (➔ p. 511)
- "ST / IL Syntax Coloring" (➔ p. 512)
- Tooltips in the ST / IL Editor

TIP

Press <Ctrl + Spacebar> to open the Variable Editor dialog box.

8.7.16.1 ST / IL Language Selection

The KAS-IDE allows you to mix IL Language and ST Language in a textual program.

- ST Language is the default.
- When you enter IL Language instructions, the program must be entered between BEGIN_IL and END_IL keywords.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

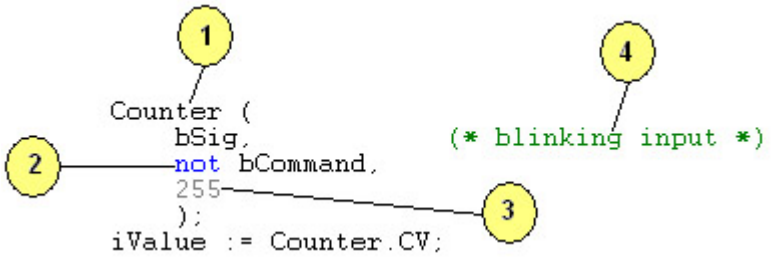
IL Language Example

```
BEGIN_IL
     FFLD   var1
     ST     var2
END_IL
```

8.7.16.2 ST / IL Syntax Coloring

The ST / IL editor supports syntax coloring according to the selected programming language (ST or IL).

The editor uses different colors for these kinds of words:



1. Black text: Default (identifiers, separators).
2. Blue text: Reserved keywords of the language.
3. Gray text: Constant expressions.
4. Green text: Comments.

TIP

Reserved words such as function names are blue.
Subprograms, including Kollmorgen UDFBs, are not reserved and show as black.

8.7.16.3 Intellisense

These features are available with Intellisense in ST and FBD programs.

NOTE

They do not apply to actions in an SFC step.

8.7.16.3.1 Conditional Compiling Coloring

Parts of conditional compiling code (declared with #ifdef pragmas) that are not validated are gray. (Figure 11-29)

```
#define CONDITION


#ifdef CONDITION


if tryGetSpike = true then
   MachineState := 2;
   MachineSpeed := 2000;
end_if;

#else


Printf('Manual mode', 0, 0, 0, 0);

// Start motion
MLMstRun(PipeNetwork.Master, TravelSpeed);


#endif
```

Figure 11-29: Example: Invalid code (gray text)

Commenting the CONDITION changes the active part of the program. (Figure 11-30)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
// #define CONDITION


#ifdef CONDITION


if tryGetSpike = true then
   MachineState := 2;
   MachineSpeed := 2000;
end_if;

#else


Printf('Manual mode', 0, 0, 0, 0);

// Start motion
MLMstRun(PipeNetwork.Master, TravelSpeed);


#endif
```

Figure 11-30: Example: Commented code (green text)

NOTE

Save your project to have the code with the correct colors.

8.7.16.3.2 Auto-indentation

Lines are automatically indented on the left when you enter structured ST statements.

See "Auto-completion of Words" (→ p. 513).

8.7.16.4 Auto-completion of Words

The ST / IL editor includes commands for automatic completion of typed words according to declared variables and data types. (Figure 11-31)

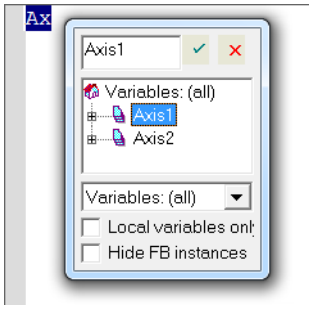


Figure 11-31: Example: Auto-completion

Press **Ctrl + Space** or **Ctrl + J** to activate auto-completion.

- If you use auto-complete and there is no match for a variable name in the Dictionary, press **ENTER** to open the Jog Speed dialog box. (Figure 11-32)
- Use the dialog box to define a new variable.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

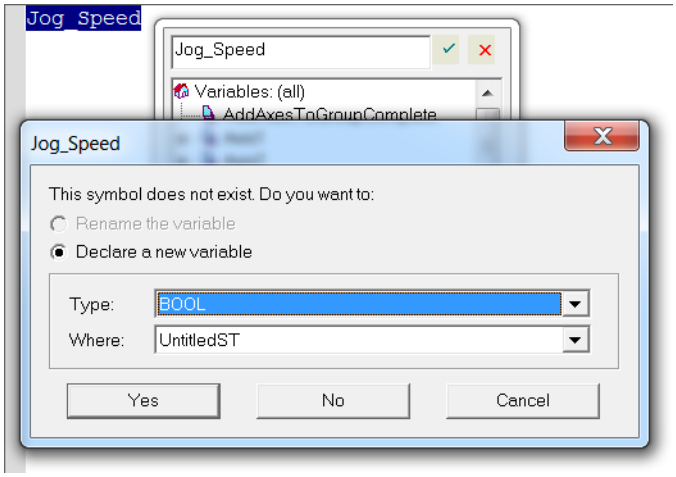


Figure 11-32: Jog Speed dialog box

These features are available:

Auto-completion of:	Action
Variable Name	• If you enter the first letters of a variable name, you can press <Ctrl+J> to automatically complete the name. • A pop-up list opens with possible choices if several declared variable names match the typed characters.
FB Member	Type the name of a function block instance (used either as an instance or a data structure). Press the point (.) after the name of the instance; a pop-up list opens with the names of possible members.

8.7.16.4.1 Other Syntax-related Commands

When lines are selected, you can automatically indent them.

Press <Tab> or <Shift+Tab> to shift the lines to the left or right, by adding or removing blank characters on the left.

8.7.16.5 ST / IL Drag-and-Drop

The ST / IL Editor supports drag-and-drop to develop and test programs.

Use the editor to:

- Drag text (words or lines) from the ST / IL editor to another application (e.g., a text editor).
  - Do the opposite.
- Drag a variable symbol from the variable editor to the ST / IL editor.
- Drag a variable symbol from the ST / IL editor to the watch list.
  - When dragging the symbol of an array to the watch list, all items in the array are added to the watch list.

8.7.16.6 Read the Output of an MC Function Block in ST

In this ST Language example:

```
A6_Inst_MC_MoveRelative( 1, Axis6, -90, 5, 300000, 300000, 0, 0 );
```

- A6_Inst_MC_MoveRelative is an instance of MC_MoveRelative PLCopen Motion function block.
- The values given in parenthesis correspond to the 8 inputs of this FB.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The syntax to read one of the outputs in ST for this instance is:

```
      <FUNCTION BLOCK NAME>.<OUTPUT>
```

This FB has these 5 outputs: Busy, Active, CommandAborted, Error, and Error ID (e.g., the **Active** output has this ST expression: A6_Inst_MC_MoveRelative.**Active**).

8.7.16.6.1.1 Example 1

```
UserVariable1 := A6_Inst_MC_MoveRelative.Error ;
```

8.7.16.6.2.2 Example 2

```
      IF A6_Inst_MC_MoveRelative.Active THEN
UserVariable2 := 1 ;
ELSE
UserVariable2 := 0 ;
END_IF;
```

8.8 Create Variables

TIP

It is recommended to use the initial to reflect the variable type (e.g., Boolean with **b**; long integer with **L**) as a naming convention for variables.

8.8.1 Use the Dictionary

For explanations on dictionary usage, including how to create and "Rename Variables" (→ p. 111) see click here....

8.8.2 Create Variables from the Editors

Create variables directly from the IEC 61131-3 editors:

FBD editor

1. Click the dedicated button.

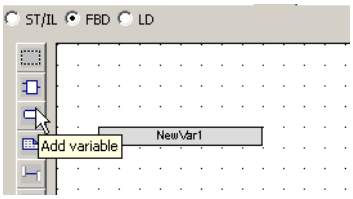


Figure 11-33: Add Variable in FBD Editor

2. Click a location in the editor (or double-click the variable if it is already created).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

3. Edit the name in the Variable Editor (or select an existing variable within the list which is already filtered according to their relevant data type).

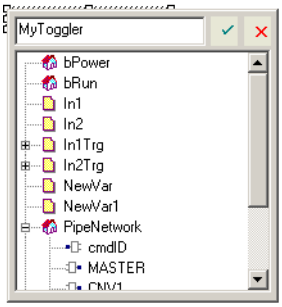


Figure 11-34: Define Variable Name in FBD Editor

4. The KAS-IDE automatically checks if the variable already exists. If it is new, you have to:
   - Select its type in the drop-down menu: for FBD and FFLD, it is set by default according to the In or Out data type of the function block
   - Specify where it is defined: the default is the current PLC program, but you can choose to make the variable Global or declared as a retain variable

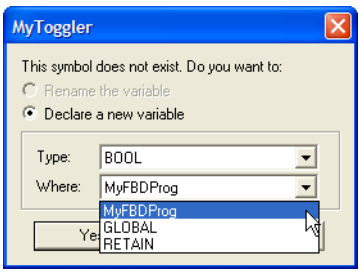


Figure 11-35: Define Variable Type in FBD Editor

See "FBD Variables" (➔ p. 491) for more information.

FFLD editor

1. Double-click the in or out pins of the function block

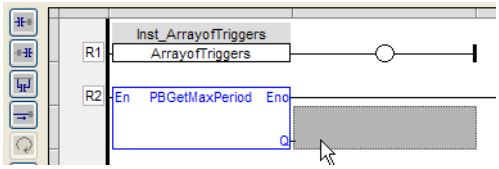


Figure 11-36: Add a Variable in the FFLD Editor

2. Edit the name (or select an existing variable within the list which is already filtered according to their relevant data type)

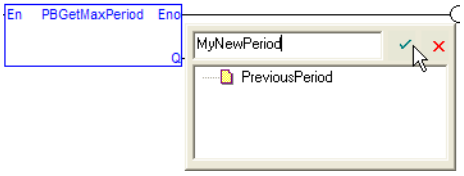


Figure 11-37: Define a Variable Name in the FFLD Editor

3. The KAS-IDE automatically checks if the variable already exists. If it is new, you have to:
   - Select its type in the drop-down menu (by default, it is set according to the In or Out data type of the function block)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Specify where it is defined

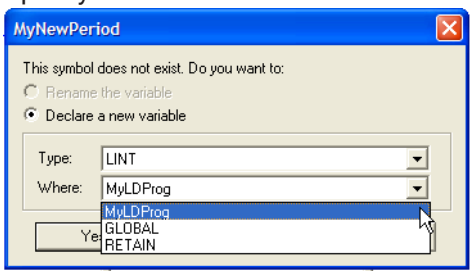


Figure 11-38: Define a Variable Type in the FFLD Editor

8.8.3 Data Types

You can create a variable of available Data Types.

How to declare an array?

1. Double-click in the corresponding cell of the variable editor (i.e. the **Dim.** column)
2. Enter its dimension (**Note**: for a multi-dimension array, enter dimensions separated by commas (ex: 2,10,4)

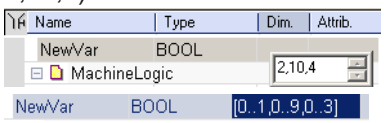


Figure 11-39: Declare an Array for an Internal Variable

also see "Arrays" on page 287

8.8.4 Complex Structures

Complex variables are arrays, structures, and instances of function blocks. The following features are allowed for programming:

- Use arrays of structures
- Use arrays of FB instances
- Pass any complex data (array, structure, instance) to a UDFB or sub-program

There is almost no limitation in the amount of complex data declared (theoretically up to 4GB, but practically limited by the memory available in the runtime)

For more explanations on the **Structure** concept, refer to (→ # 288, Structure)

8.8.4.1 Declare the Structure

1. Right-click in the Dictionary to open the menu.
2. Select the **Add structure** command.

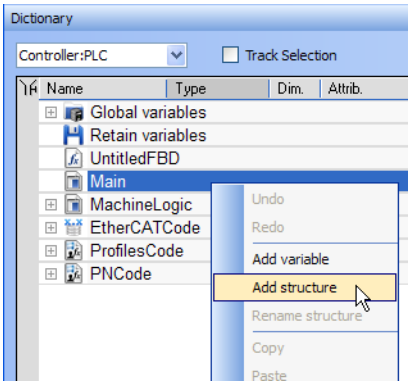


Figure 11-40: Add Structure

3. Right-click on the new structure and select the **Rename structure** command.

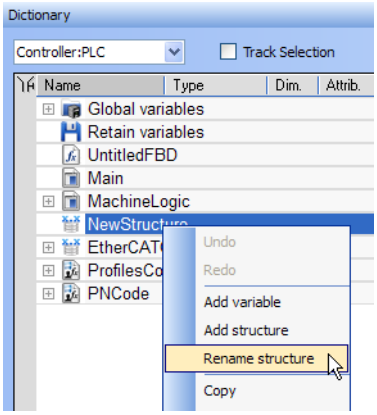


Figure 11-41: Rename Structure

4. Right-click on the new structure and select the **Add variable** command.

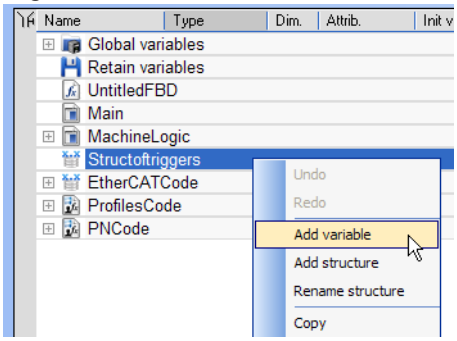
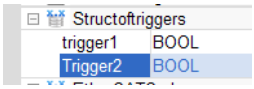


Figure 11-42: Add Variable

5. Expand the new structure.
6. Double-click on the new nested variable and define its name and type.



7. Repeat steps 5 and 6 to add all the requested variables.

8.8.4.2 Create an Instance of the Structure

When finalized, drag-and-drop the structure from the library in the **(Project)** node to a program.
A new instance is automatically created.

1. Select the new structure and move it with a drag-and-drop operation to the program declaration in the Dictionary.

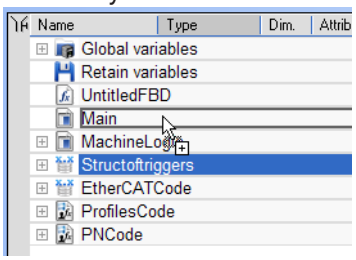
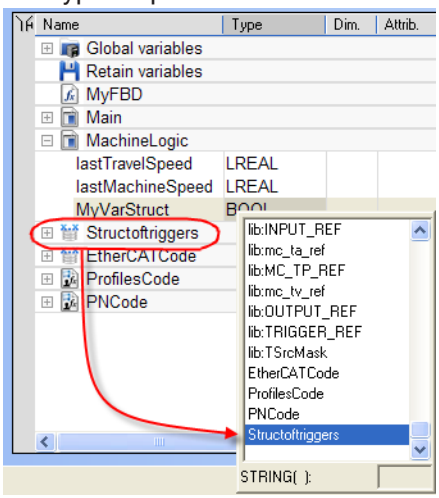


Figure 11-43: Create an Instance of the Structure

2. You can also add a variable in the Dictionary with the **Add variable** command.
Then double-click on the new variable to define its type by selecting the structure type which is displayed in

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

the Type drop-down menu.



3. Then you can drag this new instance and drop it in your program like any other variable

8.8.5 Variable Editor

You can edit variables directly from each IEC 61131-3 editor.

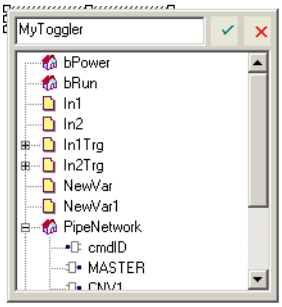


Figure 11-44: Edit the Name in the Variable Editor

TIP

Ctrl + **Spacebar** opens this dialog box.

KAS-IDE automatically checks if the variable already exists. When the variable is new, you have to:

- Select its type in the drop-down menu: for FBD and FFLD, it is set by default according to the In or Out data type of the function block
- Specify where it is defined: the default is the current PLC program, but you can choose to make the variable Global or declared as a retain variable



Figure 11-45: Define Type and Scope of the Variable

8.8.6 Edit Variables as Text

Variables can be edited as text.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Text editing applies to all the variables of a group.
- Select **Edit variables as text** from the right-click menu to open a dialog box with all of the variables in the group.
    - From here, variables may be added, deleted, or edited using the IEC 61131-3 format.

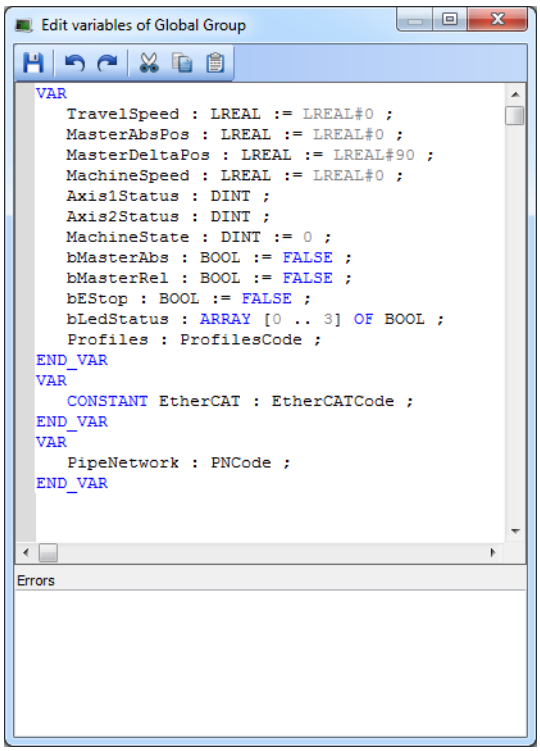


Figure 11-46: Edit Variables as text

The editing dialog contains the basic text-editing functions: Save, Undo, Redo, Cut, Copy, and Paste.

- When changes are saved, the variables are validated.
- If there are errors in the declaration, the changes are not saved and errors are listed in the "Errors" section of the dialog box.

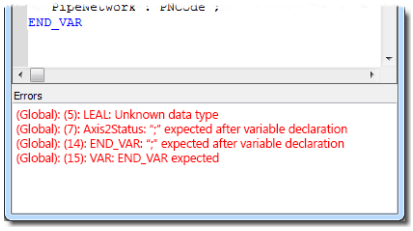


Figure 11-47: Errors caused by editing variables.

Each error includes the group name, the line number where the error occurs, and what the error is.

Double-click an entry to place the cursor at the point of the error.

IMPORTANT

If variables that are used in PDO mapping are renamed or deleted, the mapping is lost when the changes are saved.
If this occurs, an alert appears.

See "Edit Variables as Text using IEC 61131-3 Syntax" (→ p. 116) for more information about editing variables as text.

8.9 Create Functions and Function Blocks

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

For explanation about the difference between functions and function blocks, see "Program Organization Units" (→ p. 306).

8.9.1 Declare Functions or Function Blocks

1. Expand the PLC node in the Project Explorer and right-click on the "Subprograms" (→ p. 90) item.
2. Select the action to perform (i.e., **New Function** or **New UDFB**). (Figure 11-48)

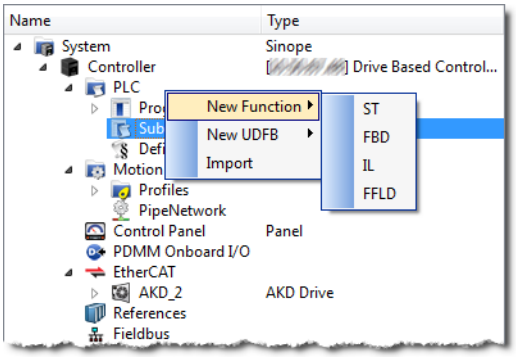


Figure 11-48: Subprogram > New Function

The **New Program** dialog box opens. (Figure 11-49)

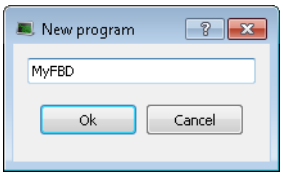


Figure 11-49: New Program dialog box

3. Enter the program name (e.g., MyFBD) and click OK.
   The **Program Properties** dialog box opens. (Figure 11-50)

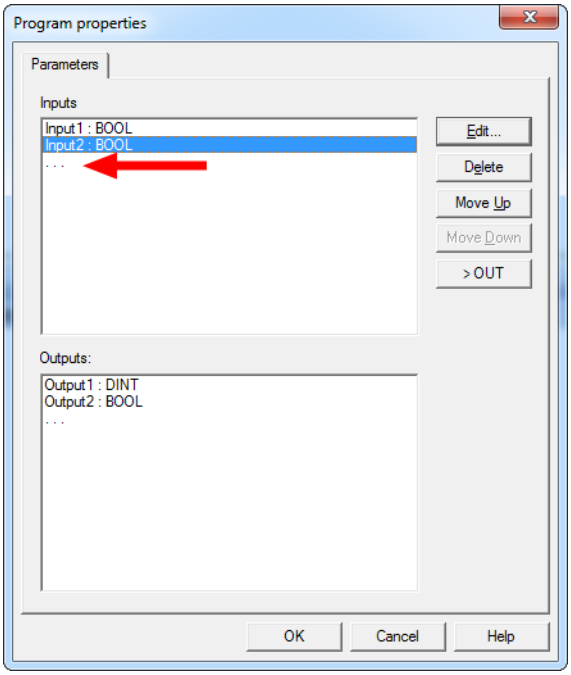


Figure 11-50: Program Properties dialog box

4. In either the **Inputs** or **Outputs** editable space, click the … to enable the **Edit** button.
   Click the Edit button or double-clicking on the editable space.
   The **Parameters** dialog box opens. (Figure 11-51)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

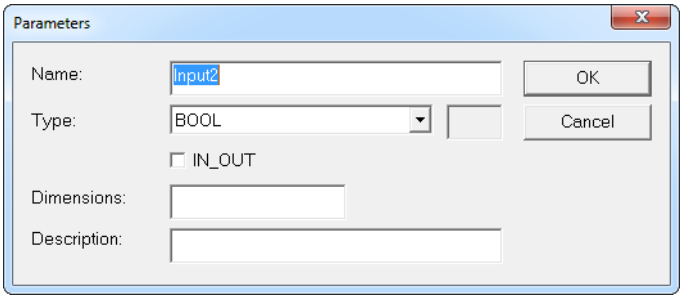


Figure 11-51: Parameters dialog box

Item	Description
Edit	Click this button to change the definition of the selected parameter. When the last line (...) of the list is selected, click this button to edit a new parameter.
Delete	Click this button to remove the selected parameter.
Move Up/Down	Click these buttons to move the selected parameter in the list to arrange the order of parameters. The order is very important because it defines the calling prototype of the UDFB or sub-program.
Swap	Click this button to move the selected parameter between the Inputs and Outputs lists. When an button is selected the button's text changes to show where the parameter is moved.

NOTE

New UDFBs are added to the **(Project)** node in the Library toolbox.

8.9.2 Define Parameters and Private Variables

For a Function or UDFB, input and output parameters (as well as private variables) are declared in the Dictionary toolbox as local variables of the item.

The **Add variable** command is used to add these items:

- Input Parameter - Externally supplied, not modifiable within the organization unit.
- Output Parameter - Supplied by the organization unit to external entities.
- Private Variable - Supplied by external entities; can be modified within organization unit.

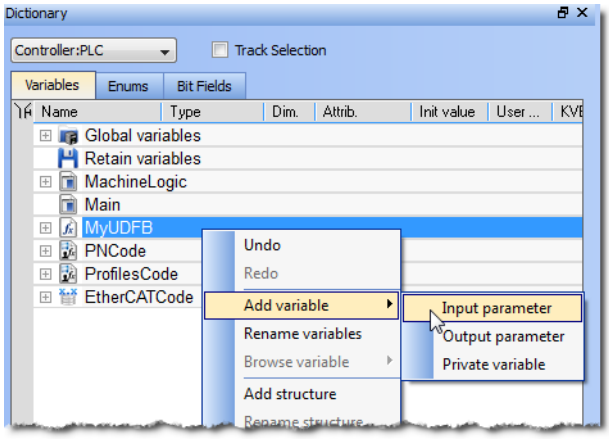


Figure 11-52: Parameters and Private Variables

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Input and Output parameters always appear at the beginning of a UDFB group.

Pressing the **INSERT** key when the item is selected adds a private variable.

NOTE

There are some things to be aware of with UDFB parameters:

- UDFB cannot contain parameters being both for Input and Output.
- Simple parameters (scalar type) can be either IN, OUT, or IN_OUT.
- Complex parameters (an array or structure) should be declared as IN but is systemically considered to be IN_OUT.
- UDFB cannot have more than 32 input parameters or 32 output parameters.
- Output parameters can only be simple "Data Types" (→ p. 286).

8.9.3 Finalize Functions or Function Blocks

Double-click the item in the Project Explorer to open and complete it in its corresponding editor.

8.9.4 Call Functions or Function Blocks

When finalized, you can drag-and-drop UDFBs from the library in the **(Project)** node to a program just like any other function block.

A new instance is automatically created. (Figure 11-53)

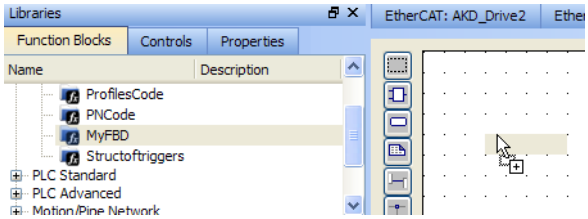


Figure 11-53: Create an Instance of UDFB in a Program

NOTE

- A single data type parameter defined as IN is passed by the calling program to the UDFB and the body UDFB cannot change its value.
- A single data type parameter defined as OUT is set in the body UDFB and always actuated in the calling program after the call.
- A parameter which is an array or a structure is always declared as IN (visible on the left of the block).
- Both the calling program and the body of the UDFB can read and write such a parameter.

8.10 Use the Defines List

The **Defines** list consists of defined constants, (an expression with a fixed value).

- **Defines** are:
  - Both pre-defined (internal) and user-created (global and local).
  - Used to determine which parts of a program's code are compiled using an `ifdef` statement.
    - See "Conditional Compiling" (→ p. 259) for more information.
  - This creates more efficient code for a given machine type.
  - Example: You can write a program that covers many machine types but compile for a specific machine with more efficient code.

Defined constants have these levels of scope:

Level	Scope
"Internal Defines" (→ p. 524)	All the projects present on your machine.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Level	Scope
"Global Defines" (→ p. 525)	All the programs in your project. These are user-defined.
"Local Definitions" (→ p. 526)	Only the current program currently open.

IMPORTANT
Pay special attention to the "Data Types" (→ p. 286) of variables used in the expression to guarantee precision when evaluating the expression.
Example: When mixing **LREAL** and **REAL**, the precision is that of **REAL**.

KAS-IDE supports "Alias Definitions" (→ p. 318).

8.10.1 Internal Defines

These are the pre-defined, common constant definitions declared for all projects.

View the Internal Defines

In the Project Explorer tree, double-click Internal Defines.

IMPORTANT
Do **not** modify these declarations.
This is to maintain consistency.

8.10.1.1 Example: Pre-defined Constants

This is an example of pre-defined constants that may be in the system.

NOTE
The exact contents of the list depends on the version of the KAS-IDE.

```
//=======================================
// Math constants
//=======================================
#define PI                                          3.141592653589793238
#define TWO_PI                                      6.283185307179586477
#define HALF_PI                                     1.570796326794896619
#define SQRT2                                       1.414213562373095049
#define SQRT3                                       1.732050807568877294
#define K_DEG2RAD
0.01745329251994329577
#define K_RAD2DEG                                   57.29577951308232088
#define PI_L
LREAL#3.141592653589793238
#define TWO_PI_L
LREAL#6.283185307179586477
#define HALF_PI_L
LREAL#1.570796326794896619
#define SQRT2_L
LREAL#1.414213562373095049
#define SQRT3_L
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
LREAL#1.732050807568877294
#define K_DEG2RAD_L
LREAL#0.01745329251994329577
#define K_RAD2DEG_L
LREAL#57.29577951308232088


//=====================================================
// Enumerated values for MLMotionStatus
// (Motion Engine Status returned values)
//=====================================================
#define MLSTATUS_NOT_INITIALISED                    0       // Motion not
initialised (British English spelling)
#define MLSTATUS_NOT_INITIALIZED                    0       // Motion not
initialized (American English spelling)
#define MLSTATUS_RUNNING                            1       // Motion is
running
#define MLSTATUS_STOPPED                            2       // Motion is
stopped
#define MLSTATUS_ERROR                              3       // Motion is
in error
```

8.10.2 Global Defines

Global Defines are user-generated constants used in a program.

Global Defines are used to write code and add an `ifdef` statement to call the Define only if it is used for a particular machine.

They are created and edited in the Project Explorer toolbox under **PLC**.

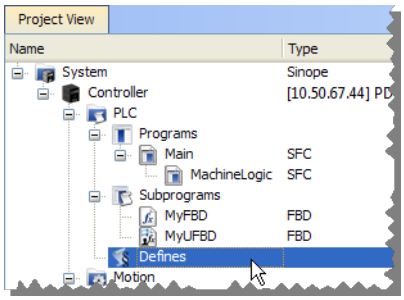


Figure 11-54: Global Defines

Double-click the **Defines** item to open the global definitions file (named: `appli.eqv`) in a text editor as:

```
#define DefReqTime    T#10ms          // 10 ms
#define BitMask       2#00100111      // binary
#define BitMaskHex    16#12AE         // hexadecimal

#define OFF           FALSE           (* redefinition of FALSE constant *)
#define PI            3.14            (* numerical constant *)
#define ALARM         (bLevel > 100)  (* complex expression *)
```

Figure 11-55: Edit the Global Definitions

Each definition must be entered on one line of text using this syntax:

```
#define   Identifier   Equivalence   (* comments *)
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.10.2.0.1 Definition Inside Another Definition

A definition within the contents of another definition is allowed.

The definition used in the second must be declared first.

Example:

```
#define PI      3.14
#define TWOPI   (PI * 2.0)
```

8.10.2.1 Local Definitions

Local definitions are user-created defines used within the corresponding program through an `ifdef` statement.

TIP

Using definitions disturbs the program monitoring and makes error reports more complex.
It is recommended to restrict the use of definitions to simple expressions.
This helps avoid misunderstandings when reading or debugging a program.

8.11 Map Input and Output to Variables

This procedure describes how to map Onboard I/O to PLC variables with these devices:

- Kollmorgen AKD Series Servo Drives
- PCMM2G
- EtherCAT motion bus I/O

IMPORTANT

This operation is disabled when the KAS-IDE is connected to the controller and an application is running.

See Also

- "Map I/O from the Project Explorer" (➔ p. 550)
- "PLC Variable Creation Wizard" (➔ p. 530)
- "PLC Variable Selector dialog box" (➔ p. 531)
- "Analog I/O Parameters" (➔ p. 533)

8.11.1 Map I/O from the Project Explorer

TIP

See the "PDO Selection/Mapping tab" (➔ p. 554) about mapping I/O with third-party devices.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. In the Project Explorer, expand the Controller node and the EtherCAT nodes to access the devices.

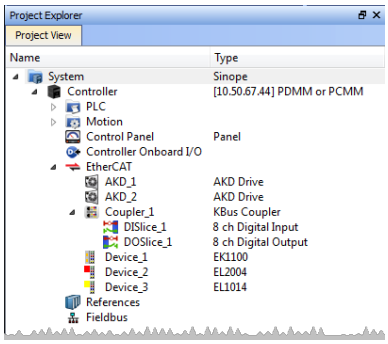


Figure 11-56: AKT (K-Bus) devices

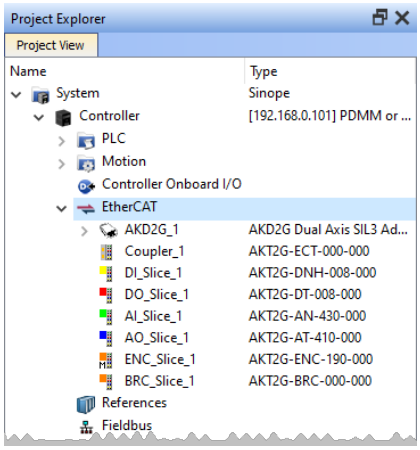


Figure 11-57: AKT2G (E-Bus) devices

2. Accessing the device properties is slightly different depending upon the device.
See "Access to Device Properties" (➔ p. 529).
3. Map the variables:
   - Drag and drop a variable from the Dictionary onto a table entry.
   - Use the "PLC Variable Creation Wizard" (➔ p. 530) for Kollmorgen devices.
   - Directly map/unmap the Inputs/Outputs to PLC variables using the "PLC Variable Selector dialog box" (➔ p. 531).
4. Select the channels to map.
Selection may be done by click-dragging or shift-clicking a range of entries.
The entire table may be selected by clicking on its border.
5. After the selection is made, right-click and select **Map** or press **<Enter>**. (Figure 11-58)
This opens the "PLC Variable Selector dialog box" (➔ p. 531).

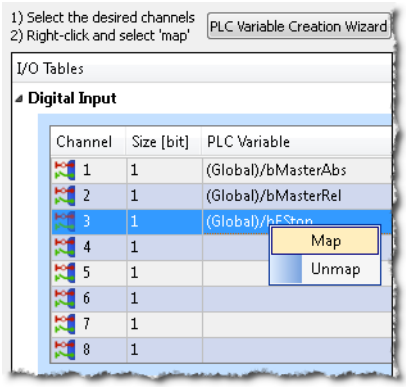


Figure 11-58: Map or Unmap options

TIP

The **Unmap** command in the contextual menu is used to remove the link between the variable and the associated channels.
Deleting a variable from the dictionary which is mapped to the channels also removes the links.

6. Choose the variable to link to the channels or PDO object. (Figure 11-59)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

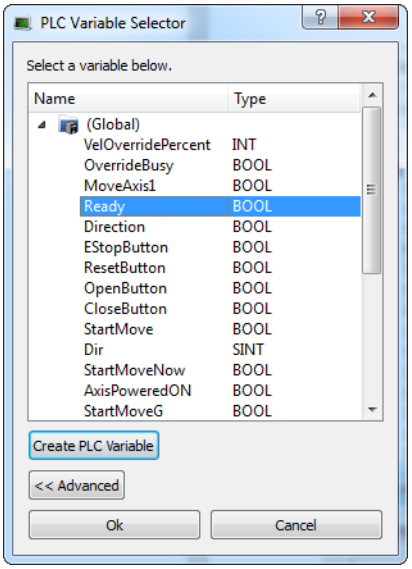


Figure 11-59: PLC Variable Selector dialog box

NOTE

The list of variables is filtered to display only those with relevant types.

- Mapping to variables within a structure instance is supported.
- Mapping to array variables and array variables in a structure instance is supported.
- Variables assigned to the Safe Digital Input and Safe Digital Output channels must be Read-only.

IMPORTANT

The limitations of PLC variable mapping are:

- Each PLC variable can be mapped to an EtherCAT I/O and exclusively to either:
  - A Controller Onboard I/O.
  - An external driver.
- Because a variable can only be mapped to one channel or PDO object, when you link a variable to a new channel or PDO object, the previous mapping is removed (even if linked to another slice or device).
- Individual bits within a variable can be mapped to multiple I/O channels on different devices (e.g., AKD, PCMM2G, or Slice).

Example: The same PLC variable cannot be mapped to both PROFINET and a Controller Onboard I/O but it is possible with a regular EtherCAT I/O.

What happens to variables that are mapped to a device I/O which was removed using the ECATDeviceAction function block?

When a device is removed using ECATDeviceAction:

- **Variable mapped to an Analog or Digital Input Type**
  - The value of the variable becomes '0' when the mapped device is removed.
  - The device I/O value is automatically reconnected to the variable when the device is reconnected to the network.
- **Variable mapped to an Analog or Digital Output**
  - The value is not sent to the device when the mapped device is removed.
  - When the device is reconnected, the value is sent to the I/O automatically.

What happens to the PLC variables which are mapped to the X22 Connector I/Os of an AKD2G if the drive is replaced with a single-axis AKD2G which does not have an X22 Connector?

All of the I/Os mapped to the X22 connector are grayed out in the IO tab: (Figure 11-60)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

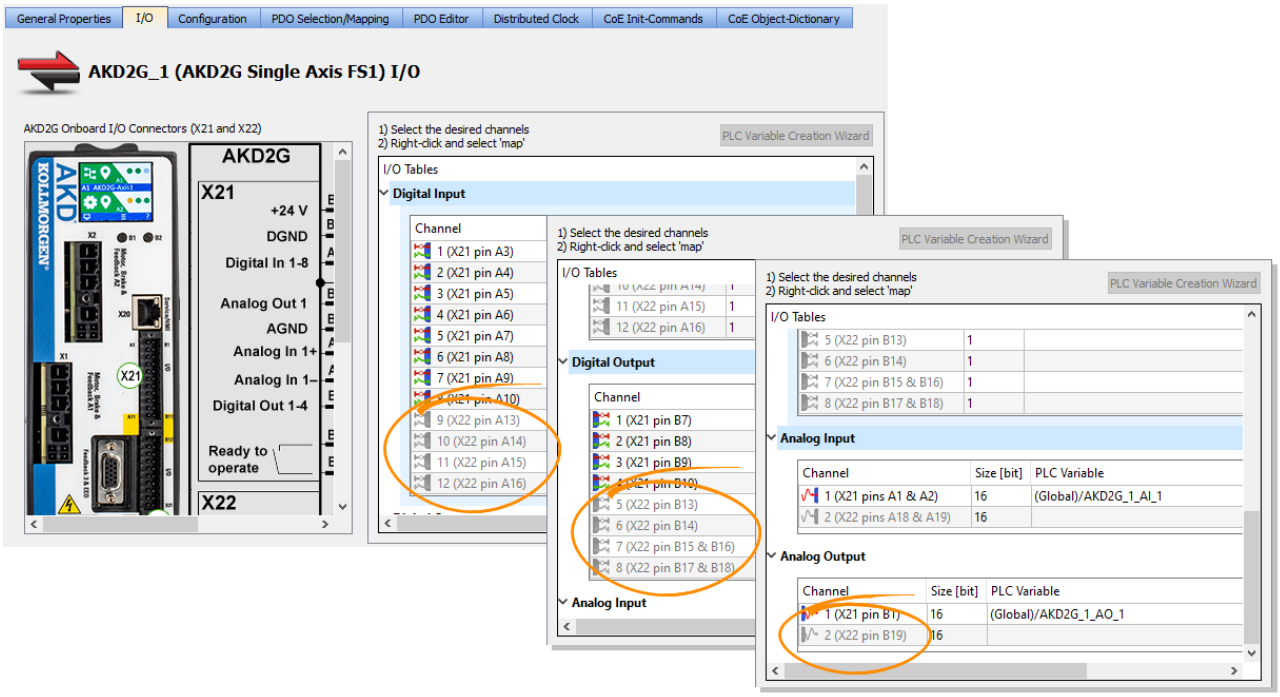


Figure 11-60: X22 I/O Map

NOTE

- AKD2G drives are available with various I/O options.
- An AKD2G drive of one variant can be replaced in the KAS project with a drive that has a different I/O option.
- See the **AKD2G Model Nomenclature** section in Automation and Motion Control Programmable Automation Solutions for available models and I/O options.

8.11.1.1 Access to Device Properties

Kollmorgen AKD Series Servo Drives

This includes Kollmorgen Essentials Drive, AKD, and AKD2G.

In the project tree, double-click the drive and select the **I/O** tab. (Figure 11-61)

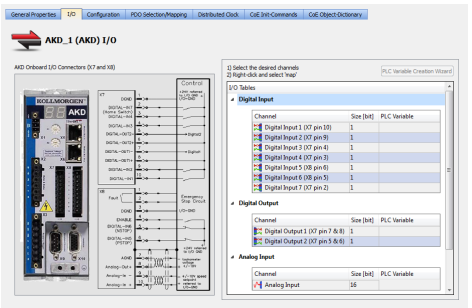


Figure 11-61: Example: AKD, AKD2G, or Kollmorgen Essentials Drive

KAS Controllers

In the project tree, double-click **Controller Onboard I/O** in the project tree. (Figure 11-62)

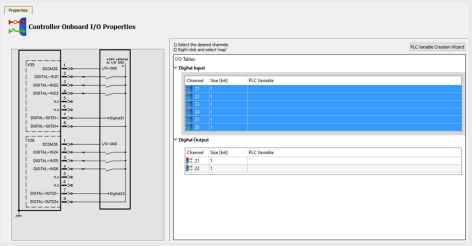


Figure 11-62: Example: Controller Onboard I/O

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

I/O Slice

In the project tree, double-click the I/O Slice under EtherCAT (e.g., AKT2G) or the Coupler entry (e.g., AKT) and click the **I/O** tab. (Figure 11-63)

For Slice Analog I/O and thermocouples, the offset and gain parameters must be defined. (Figure 11-64)

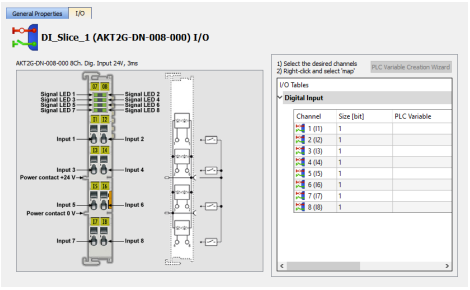


Figure 11-63: E-Bus Properties

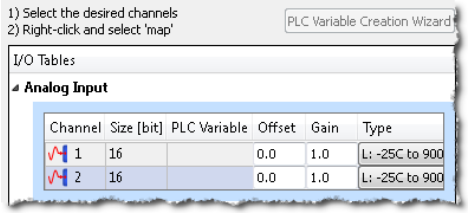


Figure 11-64: K-Bus

Safety I/O Slice

In the project tree, double-click the Safety I/O Slice under EtherCAT (E-Bus) and click the **I/O** tab. (Figure 11-65)

- Safe Digital Inputs and Output channels are clearly identifiable by a yellow table in the **I/O** tab.
- By default, the digital output logic on the safety slices / devices is disabled from the standard PLC.
- See the Description of Safety Parameters and the AKT2G-SDO-004-000Safety Terminal with Four Digital Fail-safe Outputs table.

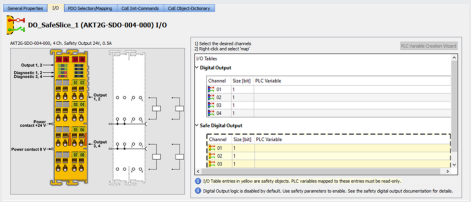


Figure 11-65: E-Bus Safety

See Also

- "Analog I/O Parameters" (→ p. 533)
- "Configure Onboard I/O" (→ p. 224)
- "Controller Onboard I/O" (→ p. 616)
- "PLC Variable Selector dialog box" (→ p. 531)

8.11.2 PLC Variable Creation Wizard

The wizard is used to automatically create a list of variables used for the mapping. (Figure 11-66)

The variable type is **Boolean** for digital I/Os and **UINT** for analog I/Os.

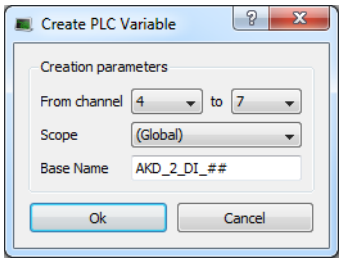


Figure 11-66: Wizard to Create PLC Variable - Parameters

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Field	Description
From channel	Defines the range of channels you want to map automatically. (Figure 11-67)
Scope	Defines where the variables are created. If you select the **Global** scope, the variables are created under the Global node in the Dictionary. (Figure 11-68)
Base Name	Pattern used for variable naming with the name of the object in the Project Tree, followed by a 2-letter acronym (2LA) representing the I/O type. The channel number replaces **##**.

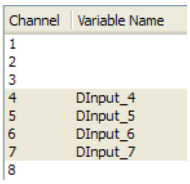


Figure 11-67: Wizard to Create PLC Variable - Mapped Channels

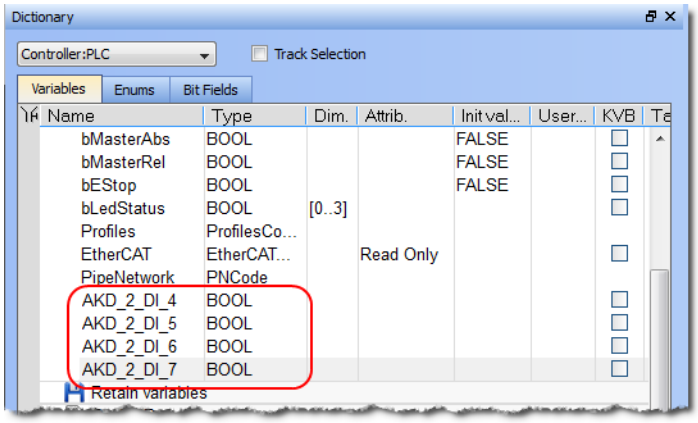


Figure 11-68: Wizard to Create PLC Variable - Variables in the Dictionary

8.11.3 PLC Variable Selector dialog box

The **PLC Variable Selector** dialog box is used to filter the variables to show only those variables compatible with the size of the selected PDO object.

- Mapping to variables within a structure instance is supported.
- Mapping to array variables and array variables in a structure instance is supported.

8.11.3.1 Create PLC Variable Button

This button is used to create a new variable to map to the selected items. (Figure 11-69)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

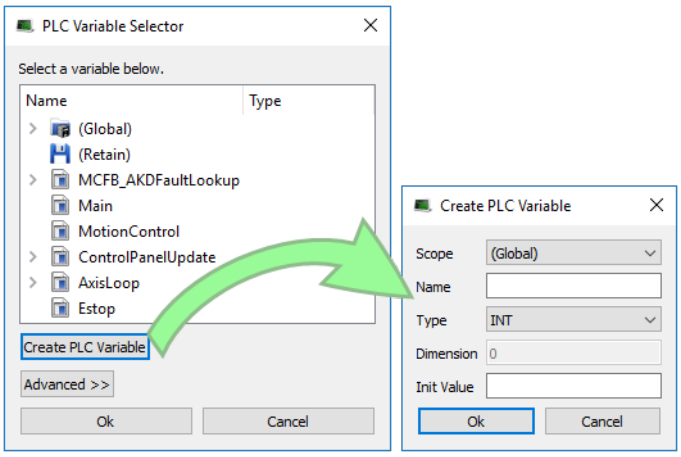


Figure 11-69: PLC Variable Selector dialog box > Create PLC Variable dialog box

Field	Description
Scope	Defines where the variable is created.
Name	See "Name a Variable" (→ p. 110).
Type	Defines the variable type and its Dimension if the variable is an array. See: • "Arrays" (→ p. 287) • "Data Types" (→ p. 286)
Dimension	The dimension of the variable (read-only). The dimension is set automatically based on the necessary size and the selected variable type.
Init Value	See "Variable Initial Value" (→ p. 115).

Based on the variable **Type** we select, the KAS-IDE automatically sets the **Dimension** (the size of the array) and assign the bits across the objects.

See the online help for a video that demonstrates mapping a variable to several PDO objects.

TIP

- Safety PDOs are highlighted in yellow with a border.
- Any variable assigned to a FSoE PDO must be Read Only.
  - See "Safety, PDOs, and Variables" (→ p. 568).
- See "Set Up a Variable to Monitor Safety Data" (→ p. 558).

See also "Create Variables" (→ p. 515)

8.11.3.2 Advanced Button

For integer variables with types stored on several bits, the Advanced button gives access to the **Start bit** definition. (Figure 11-70)

This allows you to link a set of channels to a specific range of bits within an integer variable.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

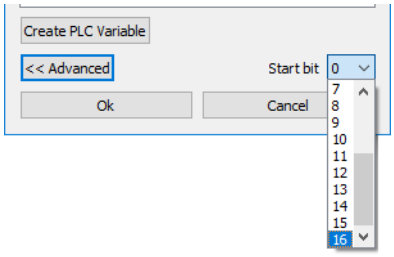


Figure 11-70: Advanced button > Start bit

Example

Select three channels ranging from 1 to 3 and map them to a DINT variable (stored on 32-bits ranging from 0 to 31).

- The first channel can be linked to position ranging from 0 to 29.
- The three channels are mapped to the last three bits ranging from 29 to 31. (Figure 11-71)

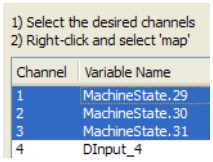


Figure 11-71: Mapped Channels

NOTE

When the selected variable is an array, the **Start bit** is disabled.
An offset is not allowed when mapping to an array.

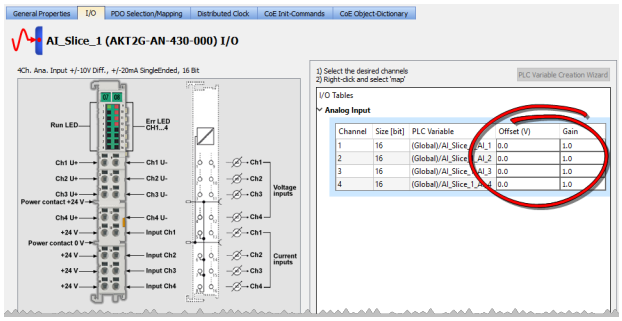
8.11.4 Analog I/O Parameters

8.11.4.1 AKT2G Input Terminals

The offset and gain is applied to calculate the value converted from the analog input:

```
PLC Variable Value = Analog to Digital Converter Value x Gain + Offset
```

Where the Gain and Offset are configured from the terminal's I/O tab.



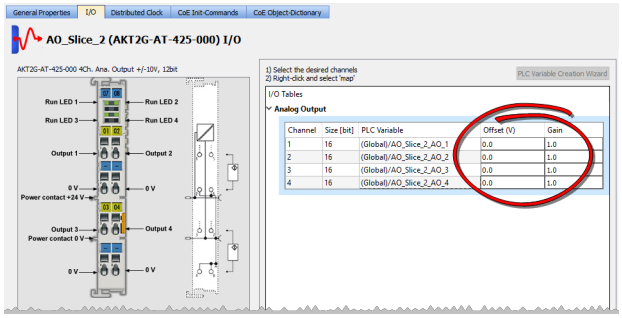
See E-Bus Analog I/O Data Representation for more information.

8.11.4.2 AKT2G Output Terminals

The offset and gain is applied to calculate the value to be converted to the analog output:

```
Digital to Analog Converter Value = PLC Variable Value x Gain + Offset
```

Where the Gain and Offset are configured from the terminal's I/O tab.



See E-Bus Analog I/O Data Representation for more information.

8.11.4.3 AKT Input Terminals

The process data that are transferred to the Bus Coupler are calculated using these equations:

$$Y_a = (B_a + X_{ADC}) \times A_a$$

$$Y_{out} = B_w + ((A_w + A_h) \times Y_a)$$

With these parameters:

Parameter	Description
X_{ADC}	Output values of the Analog Input Modules A/D converter.
Y_{out}	Process data to the controller.
B_a , A_a	Manufacturer offset and gain compensation ‡.
A_h	Manufacturer scaling: default gain ‡.
B_w , A_w	User scaling: Offset and Gain as set in the "Analog Input Properties" (➔ p. 534).

‡ For the thermocouple input terminals, AKT-AN-200-000 and AKT-AN-400-000, the manufacturer default gain is 160.
For all other supported terminals, the manufacturer default gain is 1.
The manufacturer default offset is 0 (zero) for all supported terminals.

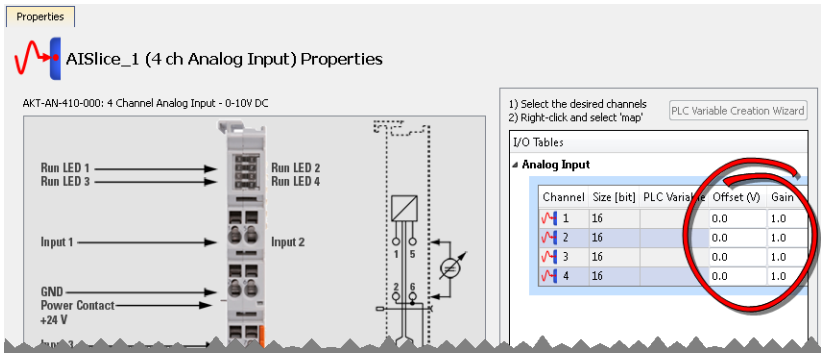


Figure 11-72: Analog Input Properties

8.11.4.4 AKT Output Terminals

The process data that are transferred to the Bus Coupler from the controller are calculated using these equations:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

$$Y_2 = B_w + ((A_w \times A_h) \times X)$$

$$Y_{dac} = Y_2 \times A_a + B_a$$

With these parameters:

Parameter	Description
X	Controller Process data.
Y_{dac}	Controller data to analog output module D/A converter.
B_a , A_a	Manufacturer offset and gain compensation ‡.
A_h	Manufacturer scaling: default gain ‡.
B_w , A_w	User scaling: offset and gain as set in the "Analog Output Properties" (➔ p. 535).

‡ The manufacturer default offset is zero for all supported terminals.
The manufacturer default gain is 1 for all supported terminals.

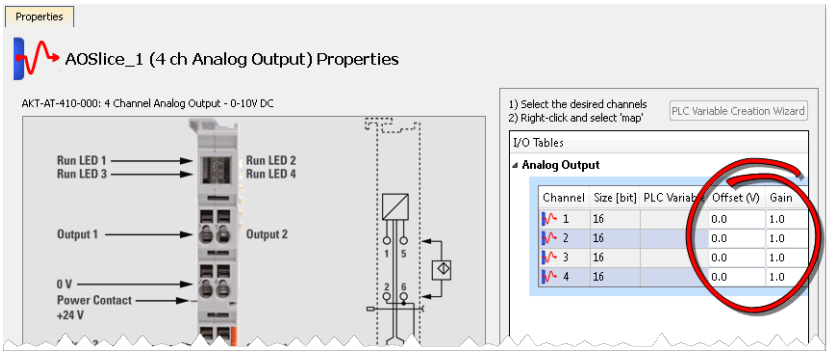


Figure 11-73: Analog Output Properties

8.12 Configure an EtherCAT

This section provides information for the EtherCAT entry in the Project Tree.

- Specifically this includes these tabs:
  - "EtherCAT Devices tab" (➔ p. 536)
  - "EtherCAT Master Settings tab" (➔ p. 539)
  - "ENI File tab" (➔ p. 541)
  - "ESI Files tab" (➔ p. 544)
  - "FSoE / PDO Connections tab" (➔ p. 546)

Double-click an EtherCAT item in the Project Tree or right-click and select **Properties** to access these tabs.

Reference information for devices in the project can be found in "EtherCAT Devices" (➔ p. 548).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

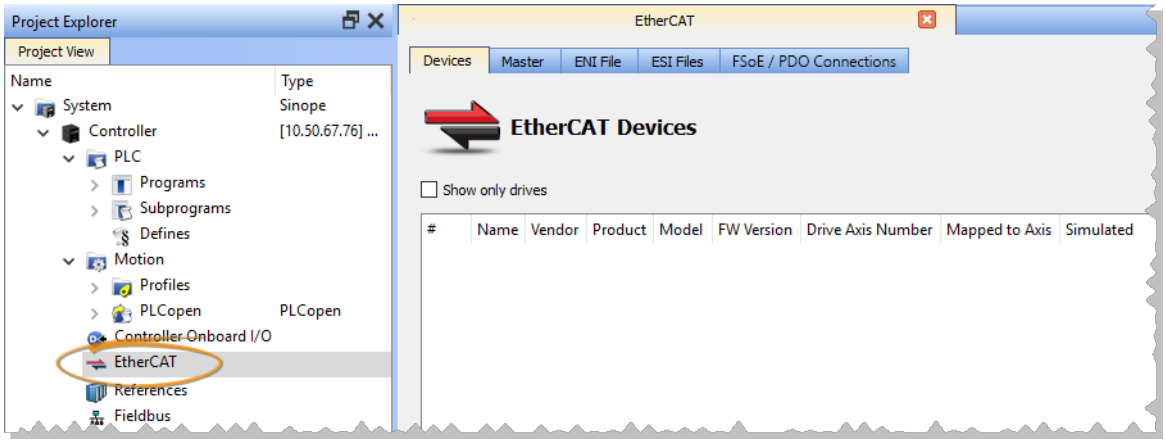


Figure 11-74: EtherCAT area of the KAS-IDE

8.12.1 EtherCAT Devices tab

The EtherCAT Devices tab lists all the EtherCAT devices and provides the ability to discover and map their use. (Figure 11-75)

- The interface is slightly different for AKD and AKD2G drives.
- The AKD2G prompts you to select the mapping by the drive axis.

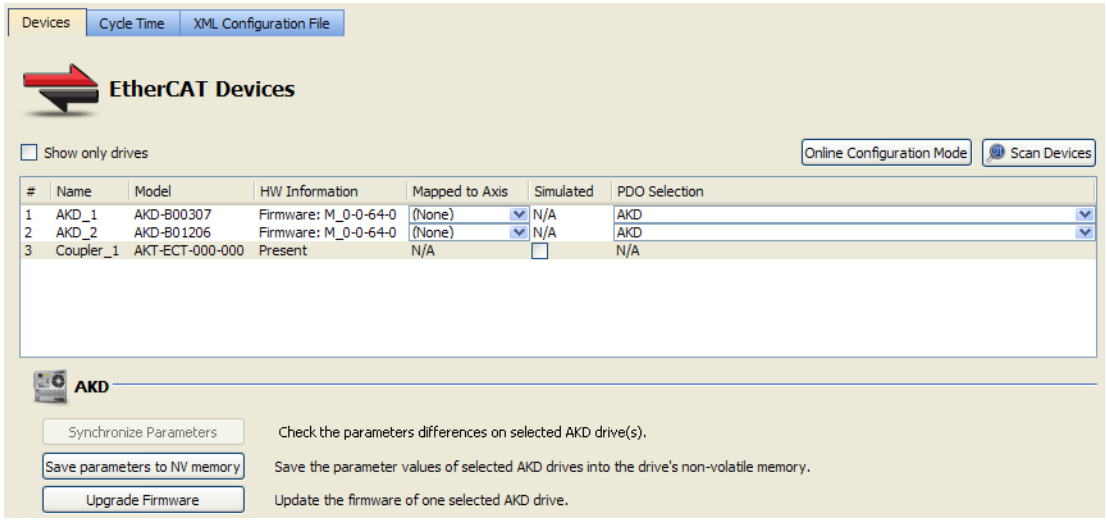


Figure 11-75: EtherCAT Devices Summary Form

IMPORTANT

First generation AKDs and Kollmorgen Essentials Drive support one axis per drive.
AKD2G drives support two axes per drive.
When configuring drives in the "EtherCAT Devices tab" (→ p. 536), you must designate which motion axis applies to each drive axis, which is the actual axis the drive controls.

8.12.1.1 EtherCAT Devices

Item	Description
Show only drives	This option hides all EtherCAT slaves from the list that are not drives.
Scan Devices	The KAS Runtime sends EtherCAT messages to discover the physical devices and modules present in the network. See "Scan Devices" (→ p. 538).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>Name, Vendor, Product, and Model</td><td>

The project device name, vendor name, product name (or code number), and model number for each device is displayed and ordered by the position in the tree.
The model (when available) includes the extension and connectivity options (e.g., <code>NBEC</code>).

</td></tr>
<tr><td>FW Version</td><td>

For Kollmorgen drives, the firmware version is displayed on the conditions that:

The Scan Devices routine was run successfully on the EtherCAT network (for drives with version 01-08-000-00 firmware or later).
Firmware was downloaded to the drive(s) while the KAS project was open in the KAS-IDE.

For other devices, the firmware version appears if the Scan Devices routine was successfully run and the device supports reading this information over EtherCAT (i.e., it uses CoE object 0x100A:0).
Otherwise (Unknown) is displayed.

NOTE

If a Kollmorgen drive is showing the firmware version as Unknown, the drive has valid resident firmware (to boot from), but does not have valid operational firmware.
Download and install the latest operational firmware and reboot the AKD / AKD2G.

</td></tr>
<tr><td>Drive Axis Number</td><td>

This number designates which axis, of those available on a drive, is being defined.
For most drives, this number is a 1.
For AKD2Gs, there are two entries:

Drive Axis Number 1.
Drive Axis Number 2.

</td></tr>
<tr><td>Mapped to Axis</td><td>
For each drive, it is displayed if it is:

Unassigned: From the drop-down menu, choose an axis that has not been assigned.

It is applicable for either PLCopen or Pipe Network motion engines.

Already mapped to a physical device: The mapping operation is done using the Scan Devices command.

See details in "Scan Devices" (➔ p. 538).

If a drive node is removed from the EtherCAT network using ECATDeviceAction, then the mapped axis (if any) is automatically forced to simulation.

</td></tr>
<tr><td>Simulated</td><td>
Used to simulate the device, which means that the device is not used and no communication to this device is performed through the fieldbus.
When can drives can be simulated?
<table>
<tr><th>Mapped to an Axis?</th><th>Simulated State</th></tr>
<tr><td>No</td><td>Simulation is not applicable</td></tr>
<tr><td>Yes</td><td>

If the Drive is mapped to a physical drive, then the simulation is Enabled.

Set the state to Yes/No --> Display check box.

If Drive is not mapped to a physical drive, the Simulation is forced to Yes.

</td></tr>
</table>
</td></tr>
</table>

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Item	Description
Save parameters to NV memory	• Save the drives' parameters to the "NVRAM" (➔ p. 771) of each drive currently selected in the list. • This action is enabled only when the Online Mode is activated. • You are automatically prompted to save modified drive parameters if this action has not been performed prior to these circumstances: • Exiting "Online Configuration Mode" (➔ p. 471). • Disconnecting from the controller. • Closing the project. • Exiting the KAS-IDE.
Upgrade Firmware	• This command triggers a firmware upgrade for the selected drives. • Use **Ctrl+A** shortcut to select all drives. • For more details, see the FAQ "Setup" (➔ p. 794) section.

8.12.1.2 Scan Devices

The scan process allows these tasks:

- Discover the devices and modules physically present in the fieldbus network.
  - See "EtherCAT Network - Physical View" (➔ p. 538).
- Map them to items under the EtherCAT node of the Project Explorer
  - See "EtherCAT Network - Logical View" (➔ p. 538).
- The order of the devices in the tree is the same as in the real fieldbus network.

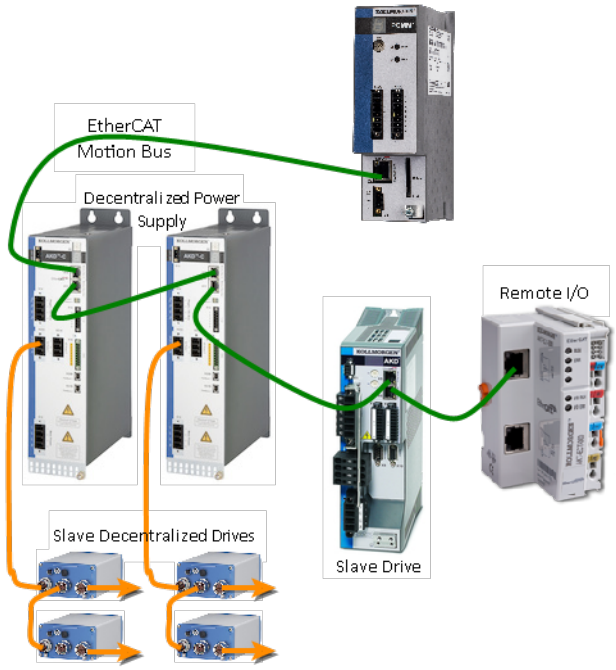


Figure 11-76: EtherCAT Network - Physical View

8.12.1.3 EtherCAT Network - Logical View

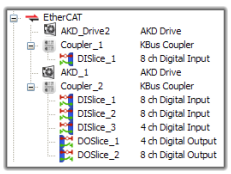


Figure 11-77: K-Bus

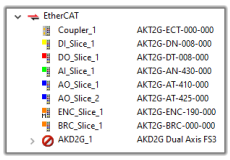


Figure 11-78: E-Bus

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

During the scan operation, all physical devices connected to the EtherCAT network are discovered and identified.

- The KAS-IDE lists the devices in the order they are connected.
- Kollmorgen devices (e.g., AKD and S300 / S700 drives, Standard I/O Couplers, and remote I/O terminals) include detailed information for each device.
  - See "Remote Input/Output Terminals" (➔ p. 777) for a list.

8.12.1.3.1 EtherCAT Device Icons

The status of devices can be determined by their icon in the logical view.

Icon	Meaning	Description
(device icon)	Normal	• The device has been added by scanning the system. • The associated ESI file has been found. • The icon varies by device, but is the icon set by the manufacturer in the ESI file.
	Excluded	The device has been added manually or has been disassociated from a discovered device.
	ESI missing	The ESI file is missing.
	Error	For AKD, shown when there is an error.

8.12.1.3.2 Scan Limitations

- I/O slices for Standard I/O Coupler do not reveal their Device IDs.
- The discovery feature does not differentiate between:
  - AKT-DN-004-000 and AKT-DNH-004-000 I/O terminals.
  - AKT-DN-008-000 and AKT-DNH-008-000.
- Devices other than those made by Kollmorgen are identified by the Vendor Name (or ID number) and Product Name (or ID number).
  - If the device is missing an ESI file, the ESI file supplied by the device Vendor must be imported.
  - The ESI file is required by the KAS-IDE to decode and display the Product Name, Device Description, and other details.
- ESI files for any MDP devices connected to the network should be added to the KAS-IDE ESI library before starting a scan.
  - If the ESI file for an MDP device is not available, the scan cannot identify the device and cannot discover any physical or logical modules under the device.

8.12.2 EtherCAT Master Settings tab

This tab includes configurations for the EtherCAT bus master.

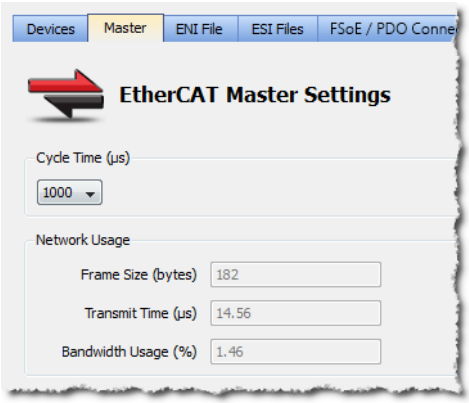


Figure 11-79: EtherCAT Master Settings tab

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Item	Description
Cycle Time (µs)	Duration of one cycle in microseconds to define the time base period for scheduling the motion and the PLC programs. Available times are: • 250 • 500 • 1000 • 2000 • 4000 • See "Tasking Model / Scheduling" (➔ p. 450) for more information about scheduling.
Frame Size	The total size (in bytes) of the EtherCAT frame sent cyclically. • The size is proportional to the number of EtherCAT slaves (and consequently the PcDO data) on the network. • The EtherCAT frame size is between 84 to 1500 bytes.
Transmit Time	The time (in microseconds) that it takes to send a frame.
Bandwidth Usage	An estimation of the percentage of the cycle time used to transmit a frame of data. Bandwidth value goes up when cycle time decreases (see calculation below)
Working Counter Error Limit	A configurable threshold for generating an E30 error and shutdown of EtherCAT communication. • The default setting is 3 Errors in 1000 frames. • Other options are: • Trigger on 1st Error • 2 Errors in 1000 frames • 10 Errors in 1000 frames • 100 Errors in 1000 frames • Disabled • If the Disabled option is selected, the Master keeps the EtherCAT network running in op mode. • It is up to the application to decide if it is appropriate to shut down the Master's EtherCAT communication (MLMotionStop). • The slave device state for EtherCAT communication errors is determined by the manufacturer. • AKD drives transition to Init state when 7 consecutive packet failures are encountered and the drive executes a controlled stop. See "EtherCAT Error Messages" (➔ p. 808) for more information about working counter.

The read-only fields show **(unknown)** when the **Use imported file** check box is selected.

Otherwise, they are recalculated and refreshed each time:

- A device is added or removed.
- A device simulation state changes.
  - See "EtherCAT Devices tab" (➔ p. 536) for more information.
- The **Use imported file** check box is cleared.

See "ENI File tab" (➔ p. 541).

8.12.2.1 Bandwidth Calculation Algorithm

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The Bandwidth (BW) usage calculation takes into account the calculated frame size and the Ethernet speed (100 Megabits per second).

```
BW% = Transmission time / Cycle Time
```

With Transmission time (µsec) = (Frame Size in bytes * 8) bits / 100 * 10^6 bps

Example

```
If Frame Size = 100 bytes
then Transmission Time = 100*8 / (100*10^6 ) = 8 µsec
```

```
If cycle time = 1000 µsec
then BW% = 8/1000 = 0.8 %
```

8.12.3 ENI File tab

During the compilation, the KAS-IDE generates the EtherCAT Network Information (ENI) file based on the EtherCAT devices defined in the project.



Figure 11-80: ENI File tab

See "Compile the Project" (➔ p. 261).

8.12.3.1 Import or Export an EtherCAT ENI File

Item	Description
Import ENI File	Used to browse and select an ENI file to be imported. • If the file is successfully imported into the project, the **Use imported file** option is automatically selected. • Once imported, the ENI file is added to the project. • This enables you to include EtherCAT devices in the project that are not natively supported by KAS. • See "Add an Unsupported EtherCAT Device" (➔ p. 889).
Use imported ENI file	Used to specify whether or not to use the imported ENI file.
Export ENI File	Used to export the ENI file generated by the KAS-IDE. • Specify the name and directory for the file. • Only the logical devices in the project tree that are mapped to a physical device (and **not** simulated) are taken into account when generating the ENI file. • This export can be useful if you want to use the file in another context or with another program.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

IMPORTANT

Importing an external ENI file overrides all EtherCAT project device information and configuration settings in the KAS-IDE.

These views and configurations are **not** applicable when using an imported ENI file:

- Project View: All devices located under the EtherCAT node.
- EtherCAT Device View tabs:
  - General Properties
  - PDO Selection/Mapping
  - Distributed Clock
  - CoE Init-Commands
- Slice I/O Properties
- Mapping PLC Variables to Slice I/O or PDO objects.

Information displayed in the views may not match the imported ENI file.

8.12.3.2 Use an Imported ENI File

- The KAS-IDE works in a degraded mode when using an imported ENI file, and the Mapped to Axis settings are disabled.
  - This is because the information about the devices in the project tree and the EtherCAT widget table is no longer relevant.
- When using an imported configuration file these parameters must be manually set for each axis:
  - the type of motion bus.
  - its address on the fieldbus ring.

  This is done by right-clicking on the Axis Pipe Block and selecting the **Properties** command.
- Scan Devices must be run from the "EtherCAT Devices tab" (➔ p. 536) before downloading the application to the controller.

8.12.4 ESI File Management

ESI files are referenced from a project's EtherCAT devices are stored in the project.

8.12.4.1 Opening a Project

When opening a project, the ESI files are copied to the local folder. (Figure 11-81)

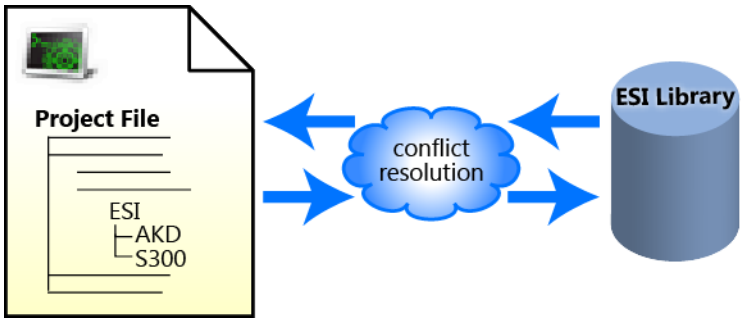


Figure 11-81: Opening

When opening a KAS project, the project's ESI files are compared to the internal library.
If there are conflicts, you are prompted to resolve them.

8.12.4.2 Adding or Deleting ESI files

Adding or deleting ESI files from KAS-IDE affects the internal library. (Figure 11-82)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

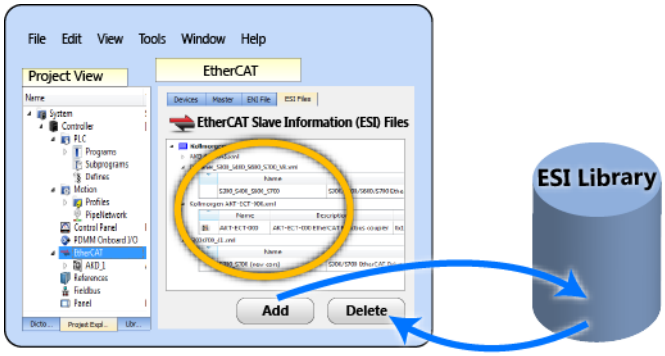


Figure 11-82: Adding/Deleting

Adding or deleting an ESI file from the KAS-IDE affects KAS's internal library of ESI files.

8.12.4.3 Saving a Project

When saving a project, the ESI files are copied to (project folder)\Controller\ESI. (Figure 11-83)

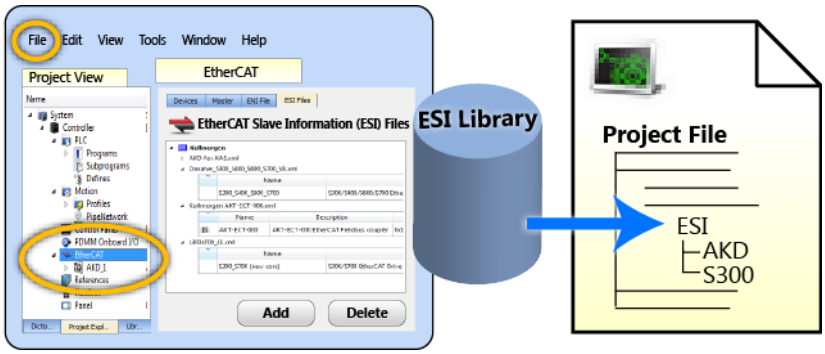


Figure 11-83: Saving

When a KAS project is saved, a copy of the ESI file is included in the project file.

TIP

The ESI files are stored along with the project.
This allows the project to be moved to another location and still compile.
It still compiles because the ESI files are copied back to the local folder.

NOTE

There is the possibility that the names and/or contents of files can be conflicting. The system checks the files before adding them. You will be prompted to chose the correct file should any conflicts be found.

⚠WARNING

Replacing an ESI file changes the configuration of any associated device to the new file.
You are prompted to confirm or cancel replacing an ESI file and the associated device configurations.

ESI files are replaced when:

- Opening a project file that contains an ESI file that is different from the file in the KAS Internal Library of ESI files stored on the PC.
  - You are prompted to select either:
    - **term_A** (use the version in the project).
      Selecting **term_A** overwrites the ESI Internal Library version.
    - **term_B** (use the ESI Internal Library version).

- Click the **Add** button in the ESI Files tab view and import an ESI file that either:
  - Already exists in the KAS Internal Library of ESI Files stored on the PC.
  - Contains duplicate device information that already exists in the KAS Internal Library of ESI Files stored on the PC.

IMPORTANT

It is strongly recommended that you do not add or remove ESI files while multiple instances of the KAS-IDE are open.
Doing so modifies a global cache of ESI files.
Changes to the global ESI file cache from one instance of the KAS-IDE could create unexpected consequences in another instance.

8.12.5 ESI Files tab

This tab lists the available EtherCAT Slave Information (ESI) files and provides the ability to add and remove files.

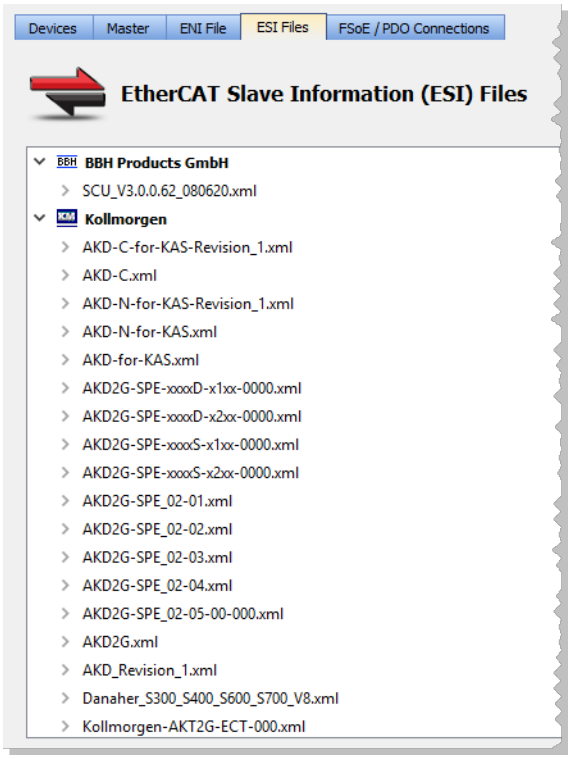


Figure 11-84: The ESI Files tab

ESI files are sorted by vendor and then by the files provided by the vendor.

Each file lists the device it supports including:

- Device name
- Device description
- Device revision number

ESI files are added to the list by clicking the **Add** button and navigating to where the file is saved.

The **Delete** button removes a file from the list but not the hard drive.

TIP

KAS-IDE manages Kollmorgen devices and automatically installs and/or upgrade their ESI files to the latest version.
The KAS-IDE allows Kollmorgen ESI files to be upgraded, but downgrading is not possible.
Example: The Kollmorgen ESI files **AKD-for-KAS.xml** or **AKD-N-for-KAS.xml** should not be manually

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

imported.

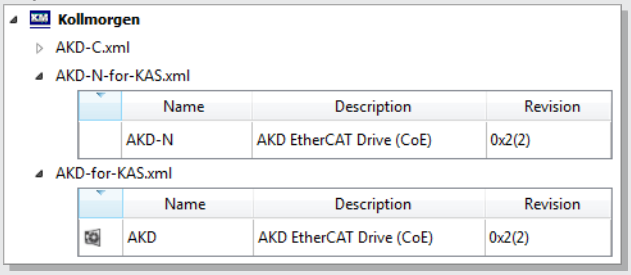


Figure 11-85: Do not overwrite these files.

Under special circumstances, Kollmorgen ESI files may be manually upgraded.
We recommend working with a Kollmorgen representative when doing so.

See "ESI File Management" (→ p. 542) about working with ESI files,

⚠WARNING

Replacing an ESI file changes the configuration of any associated device to the new file.
You are prompted to confirm or cancel replacing an ESI file and the associated device configurations.

ESI files are replaced when:

- Opening a project file that contains an ESI file that is different from the file in the KAS Internal Library of ESI files stored on the PC.
  - You are prompted to select either:
    - **term_A** (use the version in the project).
      Selecting **term_A** overwrites the ESI Internal Library version.
    - **term_B** (use the ESI Internal Library version).
- Click the **Add** button in the ESI Files tab view and import an ESI file that either:
  - Already exists in the KAS Internal Library of ESI Files stored on the PC.
  - Contains duplicate device information that already exists in the KAS Internal Library of ESI Files stored on the PC.

ⓘIMPORTANT

It is strongly recommended that you do not add or remove ESI files while multiple instances of the KAS-IDE are open.
Doing so modifies a global cache of ESI files.
Changes to the global ESI file cache from one instance of the KAS-IDE could create unexpected consequences in another instance.

8.12.6 ESI Files with References

ESI files may contain references to other ESI files or to EDS files.

- The referenced files may be located in the same directory or sub-directories.
  - ESI files with references are common with Modular Device Profile (MDP) devices.
  - This is because the ESI files for the modules are shared between several device Products and/or Revision Numbers.
- When adding an ESI file with references to the KAS-IDE's ESI library, add **only** the top-level file that contains the references to the other files and sub-directories.
  - The KAS-IDE checks the selected ESI file for reference files and automatically adds them.

Example: When adding the Murrelektronik Cube67 MDP (modular) device ESI file, select the top-level ESI file. (Figure 11-86)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

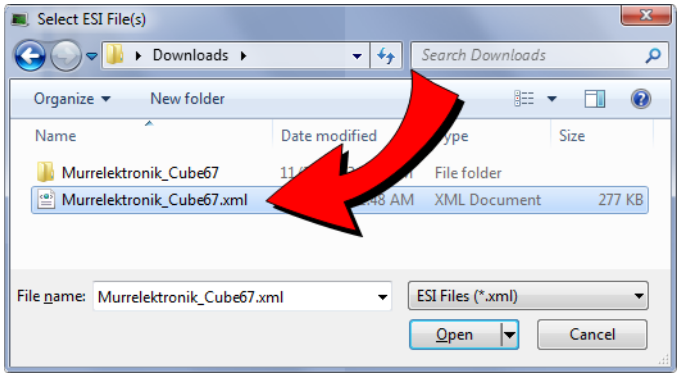


Figure 11-86: Select the top-level ESI file

Do not select the lower-level module ESI files found in a sub-folder. (Figure 11-87)

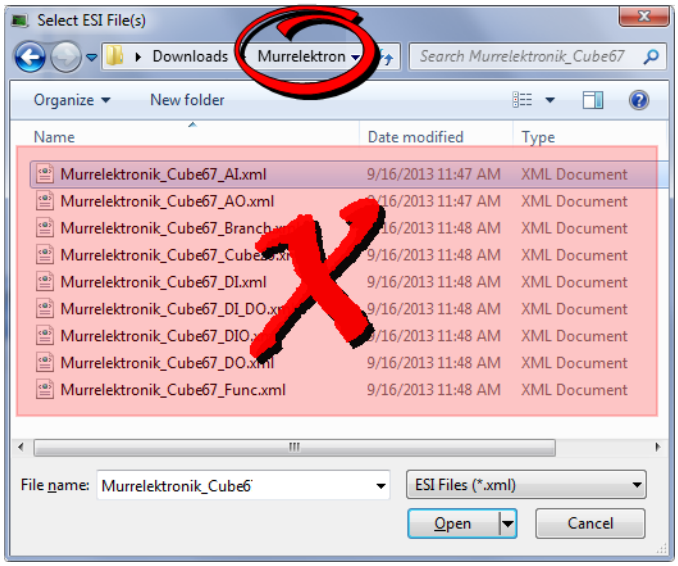


Figure 11-87: Do not select the lower-level module ESI files

IMPORTANT

WARNING: Always add the top-level ESI file to the KAS-IDE's ESI library.
Do not add, delete, or upgrade lower-level module files to the KAS-IDE's ESI library.
Removing or replacing lower-level files may cause unexpected or unknown behavior.
If any reference files are missing when the top-level file is added to the KAS-IDE's ESI library, an error message appears to show which sub-files are missing.

8.12.6.1 EtherCAT Device Configuration - MDP Devices

The KAS-IDE can configure and manage EtherCAT modular device profile (MDP)devices.

- Devices which meet the ETG.5001 standard are be added to the EtherCAT network in the Project Explorer and modules are mapped to them.
- This includes devices such as I/O device modules, safety device modules, or fieldbus gateway modules.

8.12.7 FSoE / PDO Connections tab

The Fail Safe over EtherCAT (FSoE) tab displays devices discovered on the EtherCAT network, listing their Input (Tx) and Output (Rx) PDOs.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- This provides a way to connect an Input PDO to an Output PDO.
- The KAS Runtime copies the PDO data between the connected PDOs when EtherCAT is in operational mode. This feature makes it possible to transfer safety-critical process data via the EtherCAT frames. The FSoE protocol, communication, and state machine are handled by the EtherCAT Safety devices.

TIP

Connected PDOs are intended for FSoE operation.
This feature is not limited to FSoE.
It can be useful for any application that needs to copy PDO data between EtherCAT devices.

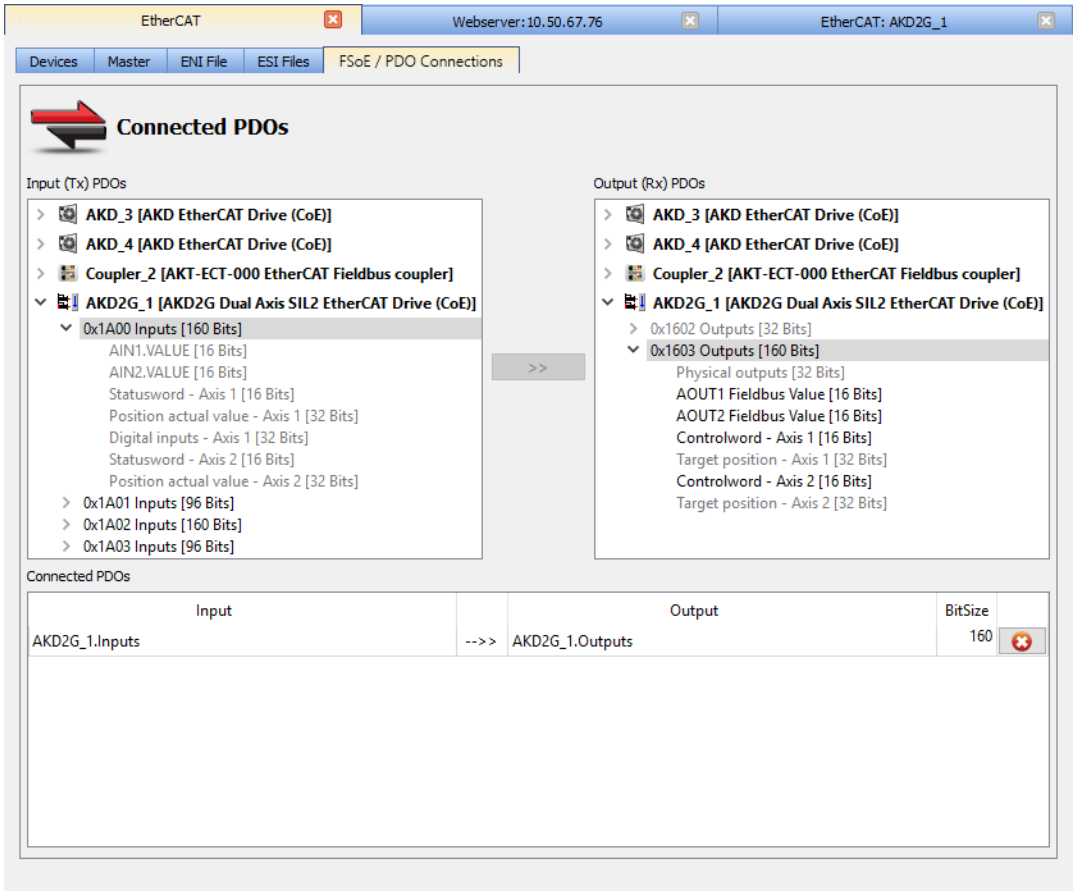


Figure 11-88: FSoE / PDO Connections tab

How to Map FSoE PDOs

These instructions assume you have already scanned the EtherCAT network.

1. Double-click (or right-click) on EtherCAT in the Project Explorer, and select the FSoE tab.
2. Select the desired Input PDO.

   Output PDOs which may be mapped (or paired) to the input are shown in black. Outputs of a different size are grayed out.
3. Select the desired Output PDO.
4. Click on the Connect PDOs button.

Connected PDOs are listed at the bottom of the tab.

TIP

See "Set Up FSoE Master and an AKD2G with SafeMotion Monitor" (➔ p. 891) for an example of mapping PDOs.

How to Unmap FSoE PDOs

1. Double-click (or right-click) on EtherCAT in the Project Explorer, and select the FSoE tab.
2. Select the desired pair of Connected PDOs from the list at the bottom of the tab.
3. Click the Delete Connection button.

What happens to the connected PDO(s) when a device is removed using the ECATDeviceAction function block?

The PDO's connection is removed internally when the device is removed using the ECATDeviceAction function block and has its PDO connected to another device.

The PDOs are automatically reconnected when the device is connected back to the EtherCAT network.

NOTE

Inputs and Outputs must have identical Bit sizes in order to be paired.

See Also

"Safety over EtherCAT" (➔ p. 421)

8.13 EtherCAT Devices

This section provides reference for defining EtherCAT devices. (Figure 11-89)

- Devices are added by right-clicking the **EtherCAT**entry in the Project Explorer.
- Devices are seen as branches of the EtherCAT item.
- Double-clicking a device opens a tab labeled **EtherCAT: [device name]**.
- See "Configure an EtherCAT" (➔ p. 535) about the tree entry.

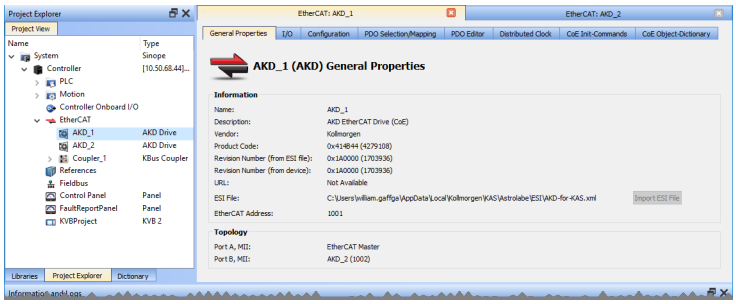


Figure 11-89: Selected EtherCAT device - General Properties tab

8.13.1 General Properties tab - EtherCAT Device

This tab provides:

- Information about the selected EtherCAT device and the associated ESI file.
- The ability to import an ESI file if one is not present.

Access this tab by:

- Double-click an EtherCAT device.
- Right-click on an EtherCAT device and select Properties. (Figure 11-90)

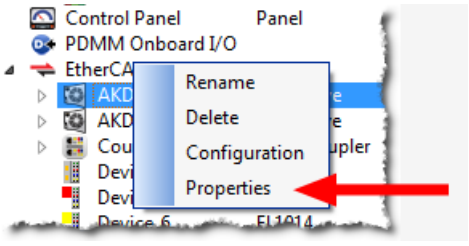


Figure 11-90: Properties option

These are the main areas in the tab: Information and Topology.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.13.1.1 Information area

This area details the device's basic device and vendor information. (Figure 11-91)

This area is not editable.

- This area shows the path to the ESI file associated with the device.
- The **Import ESI File** button is available if the ESI file is missing.
  - This lets you select a file to associate with the device.
- This area lists the **EtherCAT Address** for devices and **FSoE Address** for discovered safety devices.

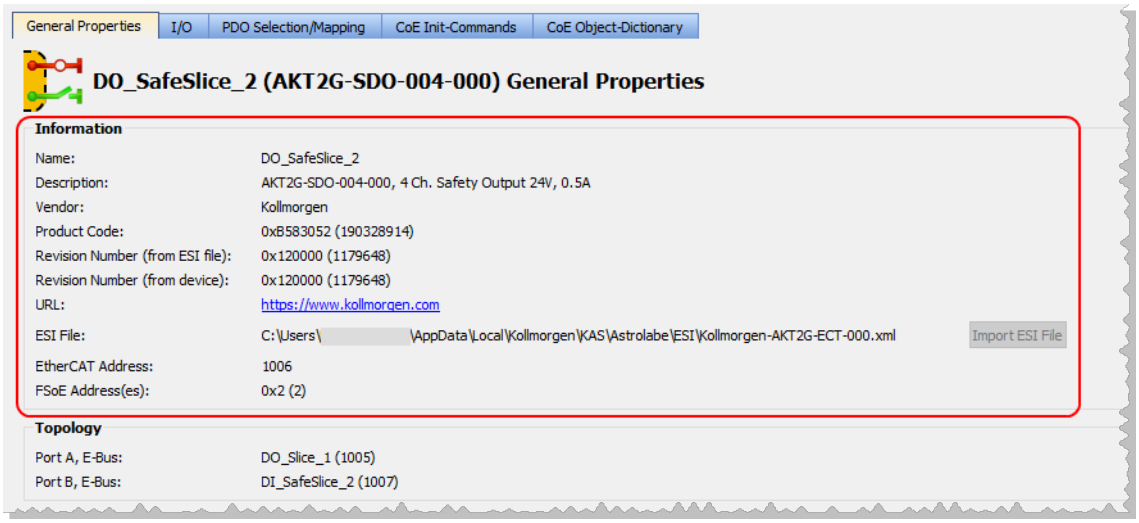


Figure 11-91: Information area - General Properties tab

Why are the Revision Numbers different?

It is possible there is a mismatch in the revision of firmware on the device and the revision referenced by the ESI file when the device was scanned. (Figure 11-92)

If this occurs, the device may not have access to the full capabilities of the latest firmware.

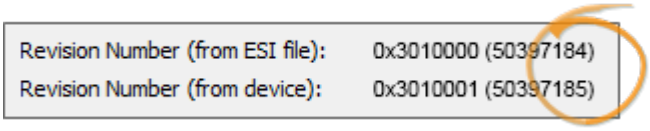


Figure 11-92: Mismatched firmware

8.13.1.2 Topology area

This section lists the device's ports, assignments, and possible states.

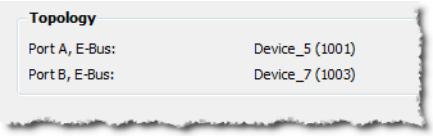
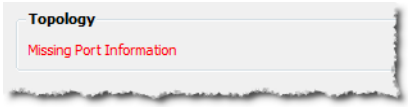
Condition	Appearance
Physical Devices	

Figure 11-93: Physical Devices

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Condition	Appearance
Virtual Device	 **Figure 11-94:** Virtual Device
Missing ESI file	 **Figure 11-95:** Missing ESI file

8.13.2 Map I/O from the Project Explorer

TIP

See the "PDO Selection/Mapping tab" (→ p. 554) about mapping I/O with third-party devices.

1. In the Project Explorer, expand the Controller node and the EtherCAT nodes to access the devices.

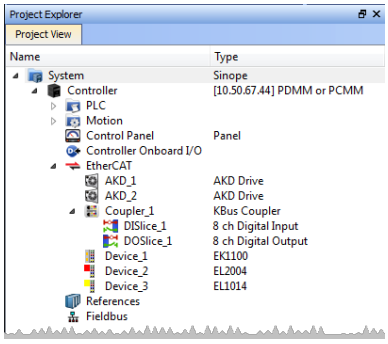


Figure 11-96: AKT (K-Bus) devices

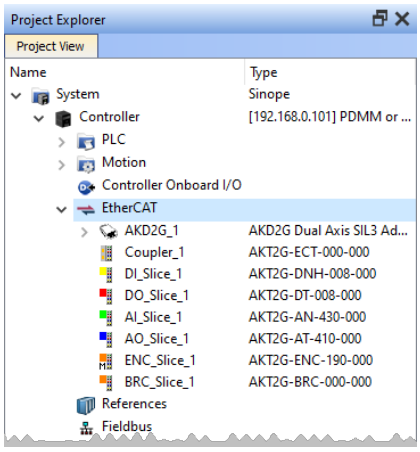


Figure 11-97: AKT2G (E-Bus) devices

2. Accessing the device properties is slightly different depending upon the device.
   See "Access to Device Properties" (→ p. 552).
3. Map the variables:
   - Drag and drop a variable from the Dictionary onto a table entry.
   - Use the "PLC Variable Creation Wizard" (→ p. 530) for Kollmorgen devices.
   - Directly map/unmap the Inputs/Outputs to PLC variables using the "PLC Variable Selector dialog box" (→ p. 531).
4. Select the channels to map.
   Selection may be done by click-dragging or shift-clicking a range of entries.
   The entire table may be selected by clicking on its border.
5. After the selection is made, right-click and select **Map** or press **<Enter>**. (Figure 11-98)
   This opens the "PLC Variable Selector dialog box" (→ p. 531).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

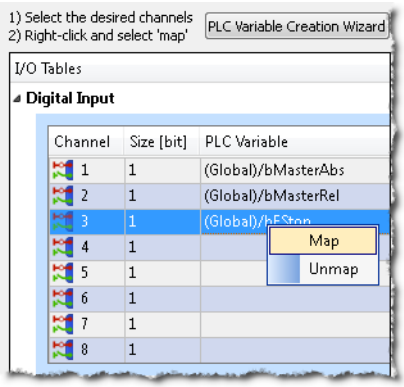


Figure 11-98: Map or Unmap options

TIP

The **Unmap** command in the contextual menu is used to remove the link between the variable and the associated channels.
Deleting a variable from the dictionary which is mapped to the channels also removes the links.

6. Choose the variable to link to the channels or PDO object. (Figure 11-99)

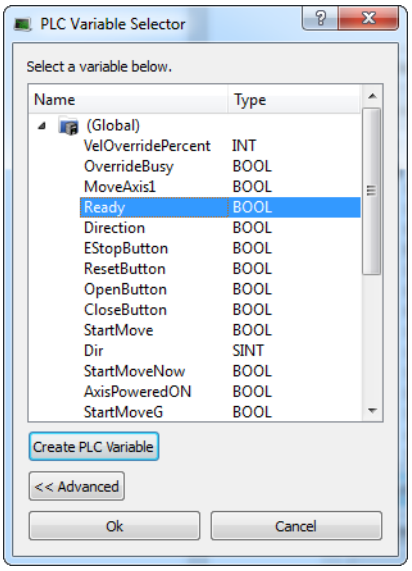


Figure 11-99: PLC Variable Selector dialog box

NOTE

The list of variables is filtered to display only those with relevant types.
- Mapping to variables within a structure instance is supported.
- Mapping to array variables and array variables in a structure instance is supported.
- Variables assigned to the Safe Digital Input and Safe Digital Output channels must be Read-only.

IMPORTANT

The limitations of PLC variable mapping are:
- Each PLC variable can be mapped to an EtherCAT I/O and exclusively to either:
  - A Controller Onboard I/O.
  - An external driver.
- Because a variable can only be mapped to one channel or PDO object, when you link a variable to a new channel or PDO object, the previous mapping is removed (even if linked to another slice or device).
- Individual bits within a variable can be mapped to multiple I/O channels on different devices (e.g., AKD, PCMM2G, or Slice).

Example: The same PLC variable cannot be mapped to both PROFINET and a Controller Onboard I/O but it is possible with a regular EtherCAT I/O.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

What happens to variables that are mapped to a device I/O which was removed using the ECATDeviceAction function block?

When a device is removed using ECATDeviceAction:

- **Variable mapped to an Analog or Digital Input Type**
  - The value of the variable becomes '0' when the mapped device is removed.
  - The device I/O value is automatically reconnected to the variable when the device is reconnected to the network.
- **Variable mapped to an Analog or Digital Output**
  - The value is not sent to the device when the mapped device is removed.
  - When the device is reconnected, the value is sent to the I/O automatically.

What happens to the PLC variables which are mapped to the X22 Connector I/Os of an AKD2G if the drive is replaced with a single-axis AKD2G which does not have an X22 Connector?

All of the I/Os mapped to the X22 connector are grayed out in the IO tab: (Figure 11-100)

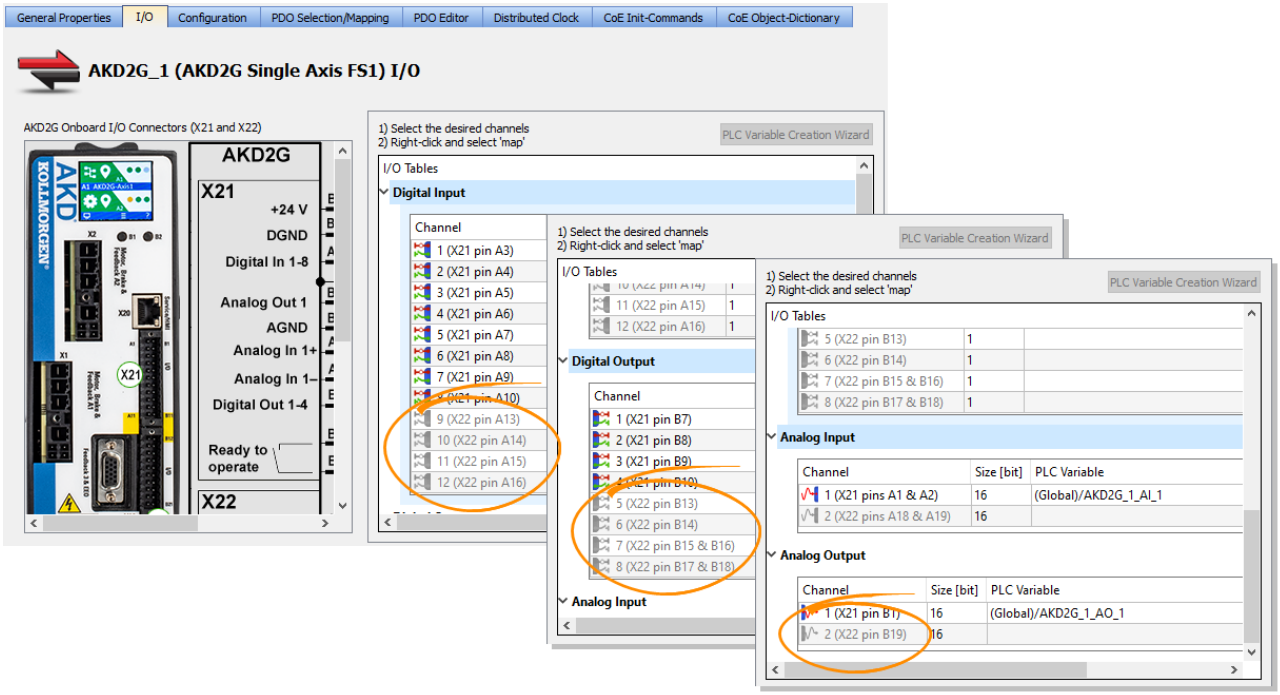


Figure 11-100: X22 I/O Map

NOTE

- AKD2G drives are available with various I/O options.
- An AKD2G drive of one variant can be replaced in the KAS project with a drive that has a different I/O option.
- See the **AKD2G Model Nomenclature** section in Automation and Motion Control Programmable Automation Solutions for available models and I/O options.

8.13.2.1 Access to Device Properties

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Kollmorgen AKD Series Servo Drives

This includes Kollmorgen Essentials Drive, AKD, and AKD2G.

In the project tree, double-click the drive and select the **I/O** tab. (Figure 11-101)

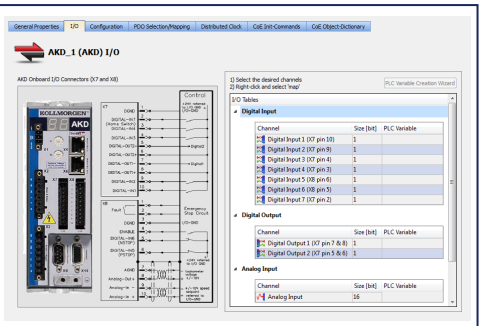


Figure 11-101: Example: AKD, AKD2G, or Kollmorgen Essentials Drive

KAS Controllers

In the project tree, double-click **Controller Onboard I/O** in the project tree. (Figure 11-102)

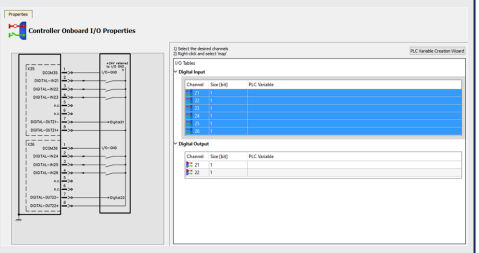


Figure 11-102: Example: Controller Onboard I/O

I/O Slice

In the project tree, double-click the I/O Slice under EtherCAT (e.g., AKT2G) or the Coupler entry (e.g., AKT) and click the **I/O** tab. (Figure 11-103)

For Slice Analog I/O and thermocouples, the offset and gain parameters must be defined. (Figure 11-104)

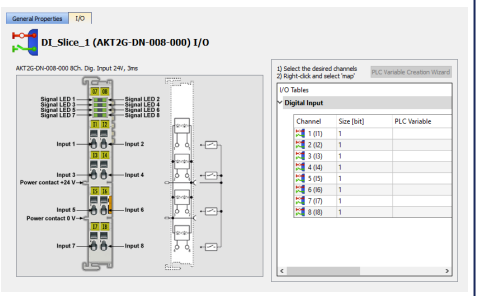


Figure 11-103: E-Bus Properties

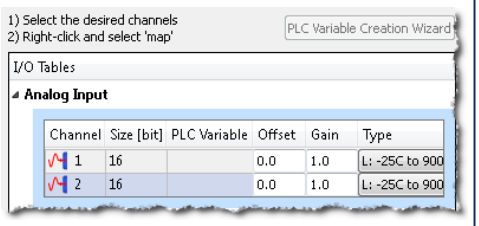


Figure 11-104: K-Bus

Safety I/O Slice

In the project tree, double-click the Safety I/O Slice under EtherCAT (E-Bus) and click the **I/O** tab. (Figure 11-105)

- Safe Digital Inputs and Output channels are clearly identifiable by a yellow table in the **I/O** tab.
- By default, the digital output logic on the safety slices / devices is disabled from the standard PLC.
- See the Description of Safety Parameters and the AKT2G-SDO-004-000Safety Terminal with Four Digital Fail-safe Outputs table.

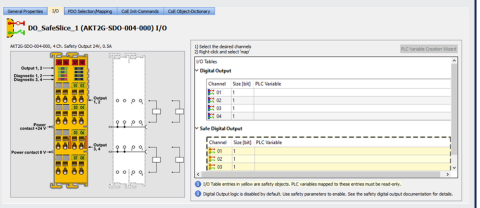


Figure 11-105: E-Bus Safety

See Also

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Analog I/O Parameters" (➔ p. 533)
- "Configure Onboard I/O" (➔ p. 224)
- "Controller Onboard I/O" (➔ p. 616)
- "PLC Variable Selector dialog box" (➔ p. 531)

8.13.3 Configuration tab

Parameters for AKD2Gs, AKDs, and AKT2G-IO stepper modules can be set by either:

- Double-click the device in the project tree.
- Right-click the device and selecting **Configuration**.

This opens a Configuration tab to set the parameters for the device.

See Also

- "Configure the Kollmorgen Servo Drive" (➔ p. 221)
- "AKT2G-IO-SM-Lxx Stepper Drive Configuration tab" (➔ p. 226)

8.13.4 PDO Selection/Mapping tab

Use this tab to:

- "Map PLC Variable to PDO Object" (➔ p. 555)
- "Select Input and Output PDOs" (➔ p. 556)
- "View the Contents of a PDO" (➔ p. 557)

This tab includes the PDO configurations for an EtherCAT device. (Figure 11-106)

- Assigned PDOs and their objects are viewable for the Inputs (Tx) and Outputs (Rx).
- The PDOs are active when the EtherCAT network is initialized to operation mode.

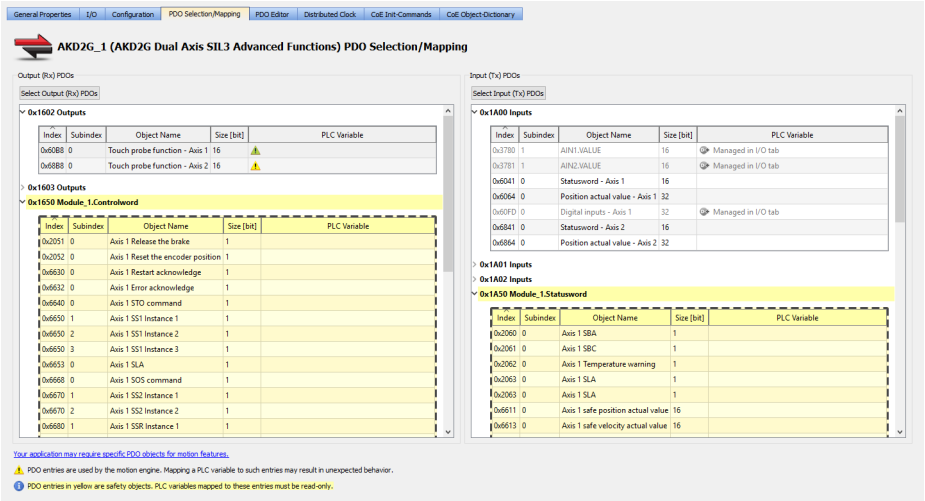


Figure 11-106: PDO Selection/Mapping tab with FSoE objects (in yellow).

TIP

- Safety PDOs are highlighted in yellow with a border.
- Any variable assigned to a FSoE PDO must be Read Only.
  - See "Safety, PDOs, and Variables" (➔ p. 568).
- See "Set Up a Variable to Monitor Safety Data" (➔ p. 558).

Each assigned PDO is listed by its **Index** (hex) and **Name**. (Figure 11-107)

- The objects associated with each PDO are listed below the name.
- The objects are identified by their object dictionary **Index** and **Subindex**.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The **Object Name** provides a simple description.
- The **Size** determines the data length.
- The **PLC Variable** mapped to the PDO objects is listed.

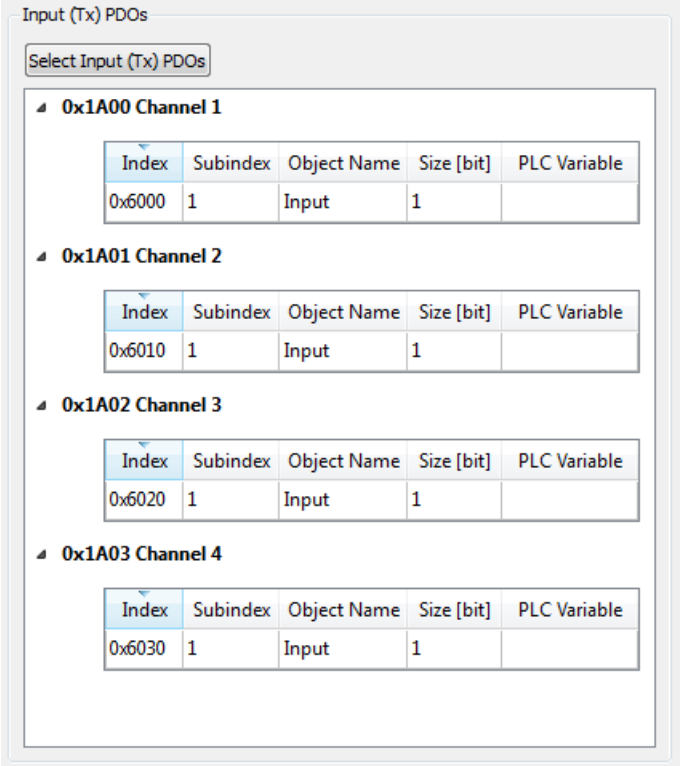


Figure 11-107: Example: Listed Input (Tx) PDOs

8.13.4.1 Map PLC Variable to PDO Object

PLC variables can be mapped to PDO objects by:

- Double-click in a PLC variable cell to open the "PLC Variable Selector dialog box" (➔ p. 531).
  - The **PLC Variable Selector** dialog box is used to filter the variables to show only those variables compatible with the size of the selected PDO object.
  - Mapping to variables within a structure instance is supported.
  - Mapping to array variables and array variables in a structure instance is supported.
- Right-click in a PLC variable cell and select **Map**. (Figure 11-108)

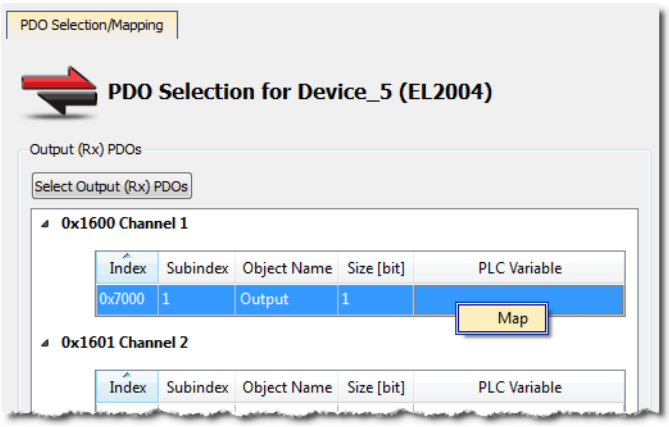


Figure 11-108: Map option

- Drag-and-drop a variable from the Dictionary to a PLC variable cell.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Multiple objects may be selected by:
  - <Shift-click> a contiguous range of objects.
  - <Ctrl-click> non-contiguous objects.
- **Do not** map PLC variables to third-party drive PDO objects.
  - They may conflict with the KAS Runtime's internal motion engine read/write operations.

Special information regarding mapping PLC variables to safety objects

PLC variables may be mapped to safety objects in exactly the same way as any other object, but the variable must be set as Read-only.

- The PLC Variable Selector shows only Read-only variables. (Figure 11-109)

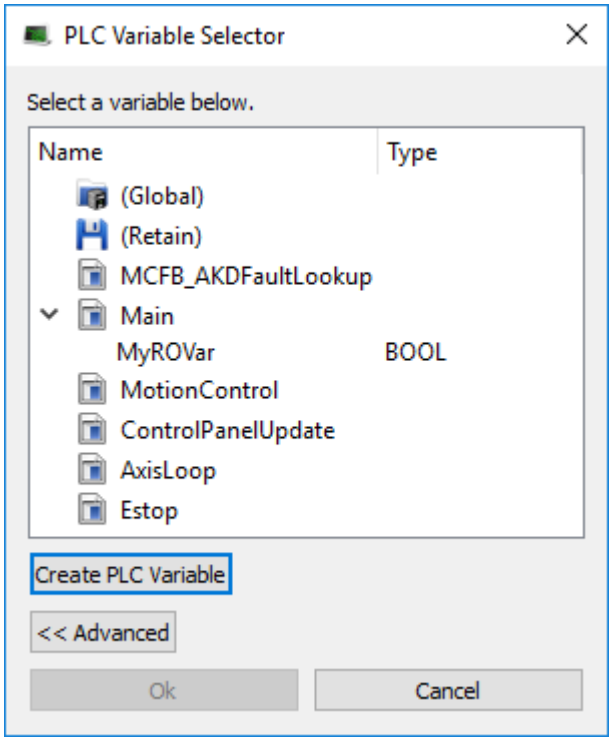


Figure 11-109: PLC Variable Selector dialog box

- Variables created by clicking the Create PLC Variable button, when accessed from a safety PDO, are automatically set to Read-only.
- There is no warning if an assigned variable is changed to Read / Write.
  - An error is reported when the project is compiled.

What happens to variables that are mapped to a device PDO which was removed using the ECATDeviceAction function block?

When an device is removed using ECATDeviceAction, what happens to variables depends on what they are mapped to.

- **Variable mapped to an Input (Tx) PDO object**
  - The value of the variable becomes 0 (zero) when the mapped device is removed.
  - The device PDO object is automatically reconnected to the variable when the device is reconnected to the network.
- **Variable mapped to an Output (Rx) object**
  - The link to the device PDO is removed when the mapped device is removed.
  - When the device is reconnected, the PDO object is automatically reconnected to the variable.

See "Map Input and Output to Variables" (→ p. 526).

8.13.4.2 Select Input and Output PDOs

Press the **Select Input (Tx) PDOs** or **Select Output (Rx) PDOs** button to choose the Input or Output PDOs.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The appropriate dialog box opens. (Figure 11-110)

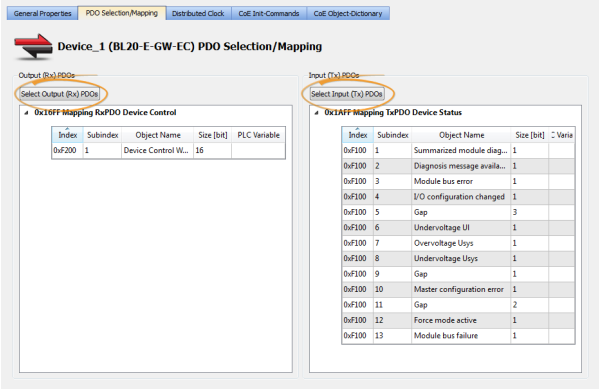


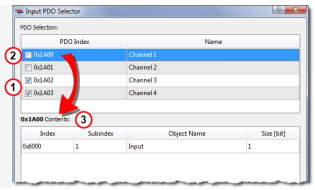
Figure 11-110: Example: PDO Selection/Mapping tab

- The upper portion contains check boxes to select specific PDOs for the EtherCAT network cyclic data.
- The lower portion contains the list of objects included within a specific PDO.

8.13.4.3 View the Contents of a PDO

Click a row in the upper portion of the PDO Selector dialog box to view the contents of that PDO.

- The lower portion is updated to list the associated objects.
- Example: The PDOs 0x1A02 and 0x1A03 are selected for the input objects but the contents for PDO 0x1A00 are listed because that row is selected. (Figure 11-111)



1. Selected PDOs.
2. Selected row.
3. Contents of selected row's PDO.

Figure 11-111: Example: Selected row

Notes

- Some EtherCAT devices may not have selectable input and/or output PDOs.
- Some PDOs allow you to select more than one PDO at a time while others are exclusive.
- Exclusive PDOs prevent simultaneously selecting certain other PDOs.
  - Example: Choosing PDO 0x1A00 excludes selecting PDO 0x1A04: (Figure 11-112)

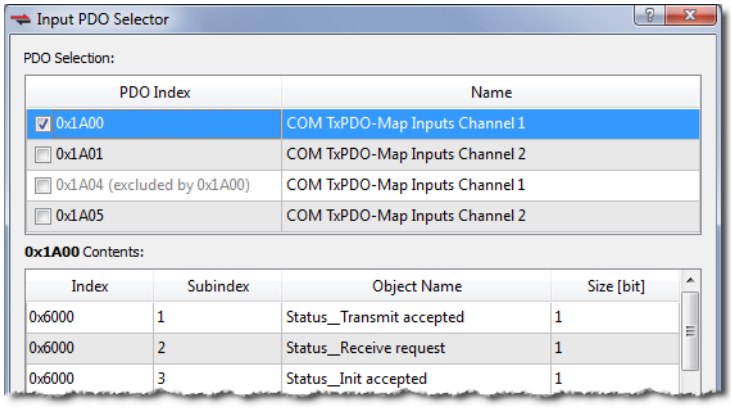


Figure 11-112: Example: PDO 0x1A00 selected

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Device vendors determine the PDOs, content and possible selection exclusivity.
  - This information is defined inside the device vendor's ESI file.
  - Contact your local Kollmorgen representative for more information about the drive.

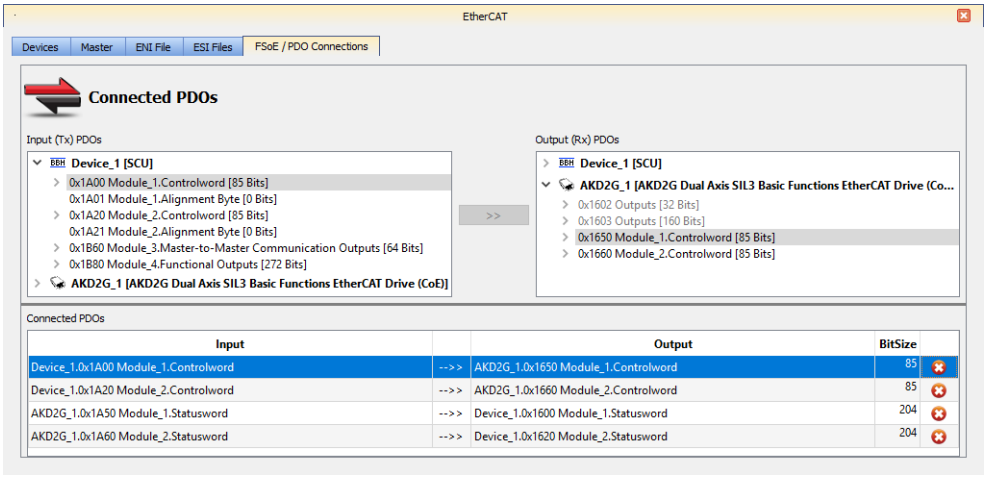
8.13.4.4 Set Up a Variable to Monitor Safety Data

This example creates a variable used to monitor data from a safety device.

The project for this example has a BBH SCU and an AKD2G.

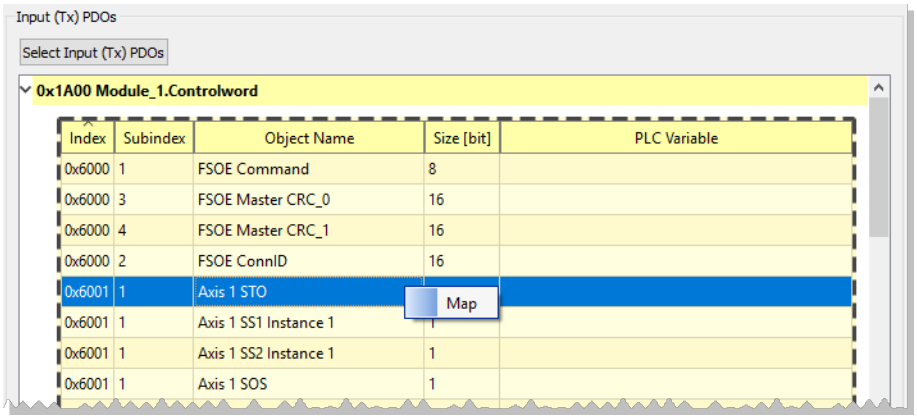
1. Connect the SCU and drive PDOs.
   See "FSoE / PDO Connections tab" (→ p. 546).

   While this step is optional, it is quite useful and will save you time and effort.



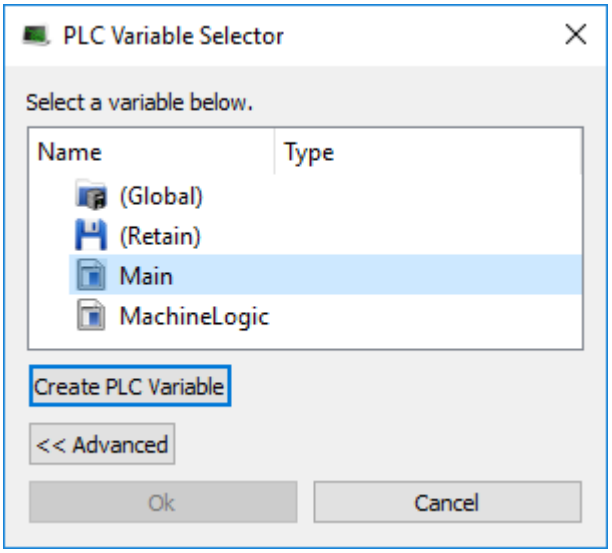
2. Right-click (or double-click) an **Input (Tx) PDOs** object that will be monitored and select **Map**.

   It is recommended that you use Tx PDOs as the source for a variable as they update every cycle, where Rx PDOs may not be current.

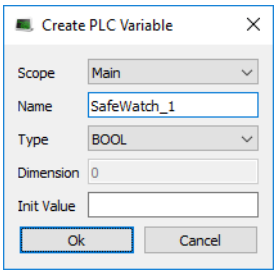


3. Read-Only variables are the only ones that will be shown in the **PLC Variable Selector**.
   Click **Create PLC Variable**.
   In this case we are creating the variable inside of **Main**.

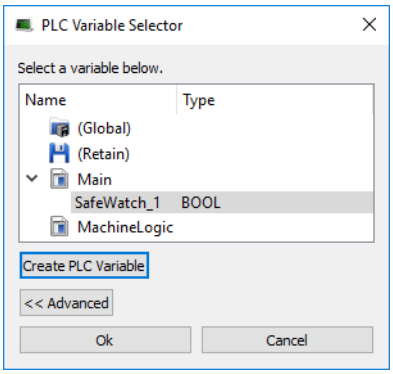
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



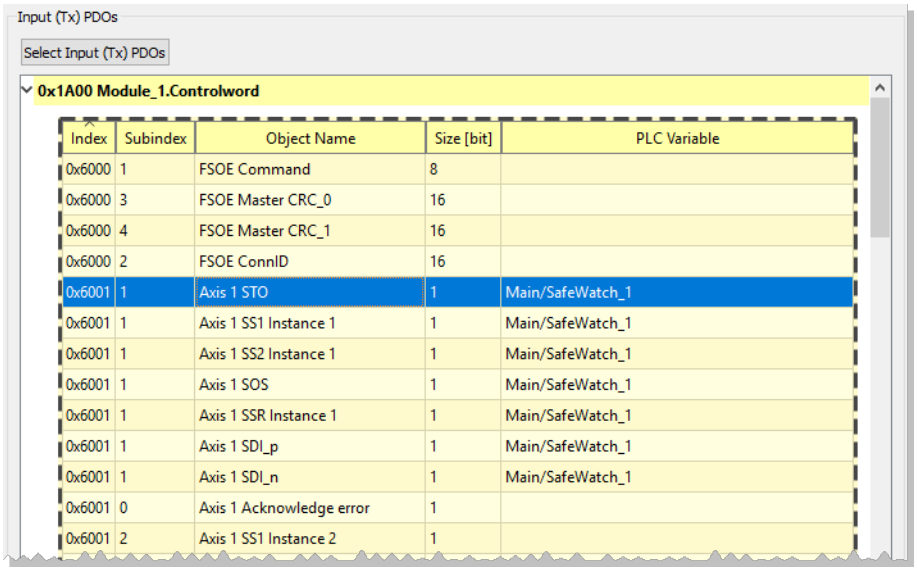
4. Define a variable by giving it a name, type, and optionally an initial value and click **OK**.



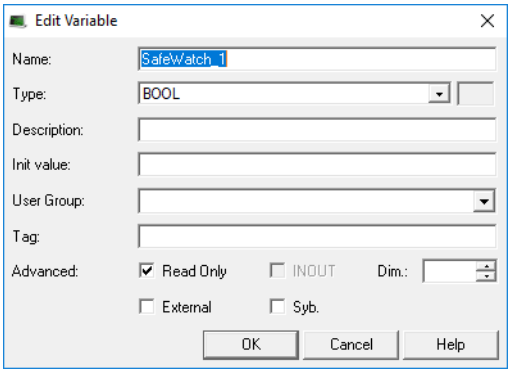
5. The **PLC Variable Selector** now displays variable.
Click **OK**.



The variable is now assigned and available to monitor any object in the same Index:Subindex.



6. **OPTIONAL:** The variable may be further defined, such as giving it a description.
   Looking at it, we can see that it was created as **Read Only**.
   It is recommended that the variable is not changed to Read/Write.
   If it is changed an error will be generated when the project is compiled.



8.13.5 PDO Editor tab

Use this tab to edit PDOs.

This tab has these sections:

- "Editable PDOs" (→ p. 561)
- "PDO Content" (→ p. 563)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

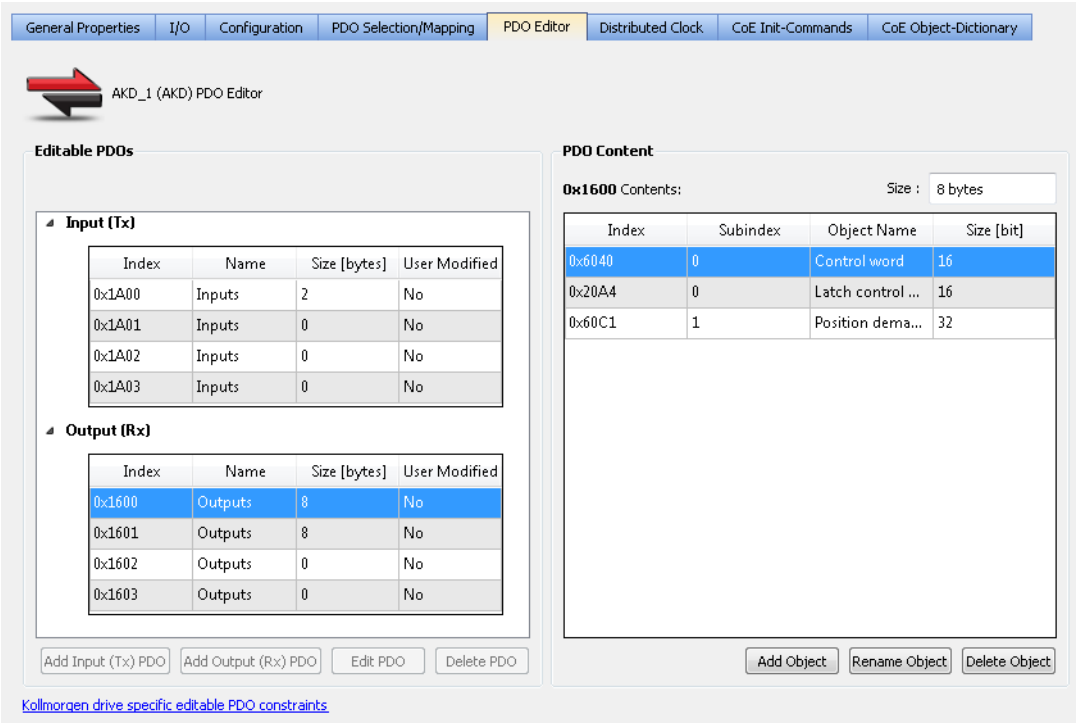


Figure 11-113: PDO Editor tab

NOTE

See "PDO Restrictions and Compiler Errors" (➔ p. 565) about compiler errors and certain constraints.

8.13.5.1 Editable PDOs

This section lists all of the PDOs which can be edited. (Figure 11-114)

- PDOs can be created (Add PDO), modified (Edit PDO), or removed from the project (Delete PDO).
- See the EtherCAT device-specific documentation for more information.

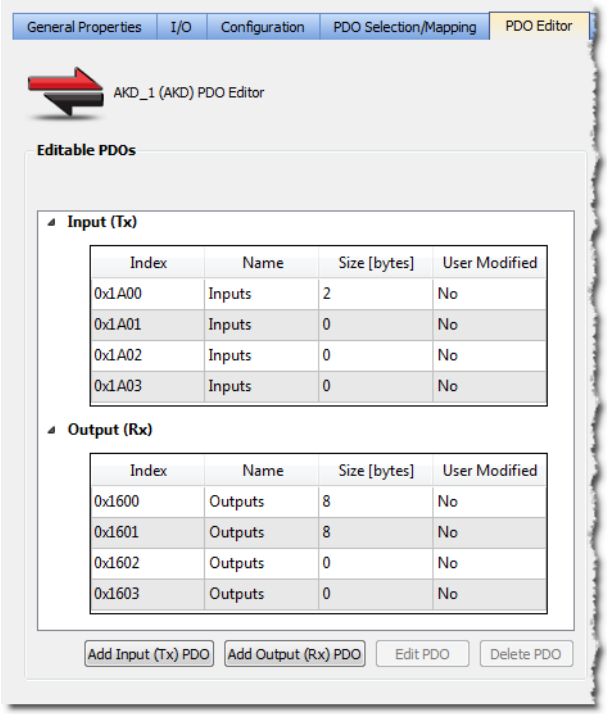


Figure 11-114: PDO Editor tab - Editable PDOs area

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The Input (Tx) and Output (Rx) PDOs are grouped separately.

Each PDO table contains this information:

Name	Description
Index	Identification value for the PDO. The range is defined by the EtherCAT specification. • Rx PDOs: 0x1600 to 0x17FF • Tx PDOs: 0x1A00 to 0x1BFF
Name	• Text label to identify the PDO. • The name is provided either by the device vendor or user. • It can be changed by the user.
Size [bytes]	Total length for the PDO object(s) data.
User Modified	• Yes: Indicates the user has changed the PDO or its content. • No: Indicates the PDO is the default provided by the device vendor.

- The KAS-IDE compiler enforces restrictions on the sizes of individual PDOs and the sum of the sizes of PDOs.
- See "PDO Restrictions and Compiler Errors" (➔ p. 565).

8.13.5.1.1 Add a Flexible PDO

1. Click the **Add Input (Tx) PDO** or **Add Output (Rx) PDO** button.
   The **Add Input** or **Add Output** dialog box opens.
2. Enter a **Name** and select an **Index** value for the PDO.
   The dialog performs a live check for valid entries. (Figure 11-115)
3. Click **OK** to close the dialog box.

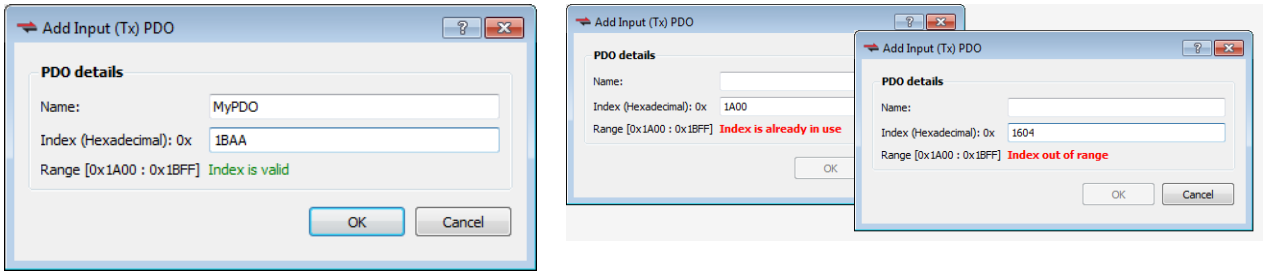


Figure 11-115: Examples of live checking for valid entries.

- Objects may be added once the PDO has been created.
  - See "Add a CoE Object" (➔ p. 564).
- EtherCAT device support for flexible PDOs is optional.
  - See the manufacturer's device-specific manual about support for flexible PDOs and supported PDO index values.

8.13.5.1.2 Edit a Flexible PDO

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

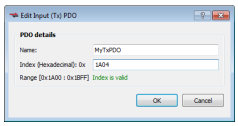


Figure 11-116: Example: Edit Input (Tx) PDO

1. Select a flexible PDO from the list of Editable PDOs.
2. Click the **Edit PDO** button.
   The Edit Input (Rx / Tx) PDO dialog box opens.
3. Make the changes.
   The dialog performs a live check.
   The **OK** button is activated if the entry is valid.
4. Click **OK** to save the changes or selections and close the dialog box.

8.13.5.1.3 Delete a Flexible PDO

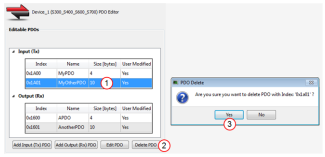


Figure 11-117: Editable PDOs and Delete confirmation mess

1. Select a flexible PDO from the list of Editable PDOs.
2. Click the **Delete PDO** button.
   A confirmation message appears.
3. Click **Yes** to confirm the deletion.

8.13.5.2 PDO Content

This are lists the objects in the currently selected PDO.

- Objects can be created (Add Object), modified (Edit Object), or removed from the project (Delete Object).
- Select a PDO in the **Editable PDOs** section (left side of dialog box) to view the objects in a PDO.
- The list of objects appear in the **PDO Content** section (right side of dialog box). (Figure 11-118)

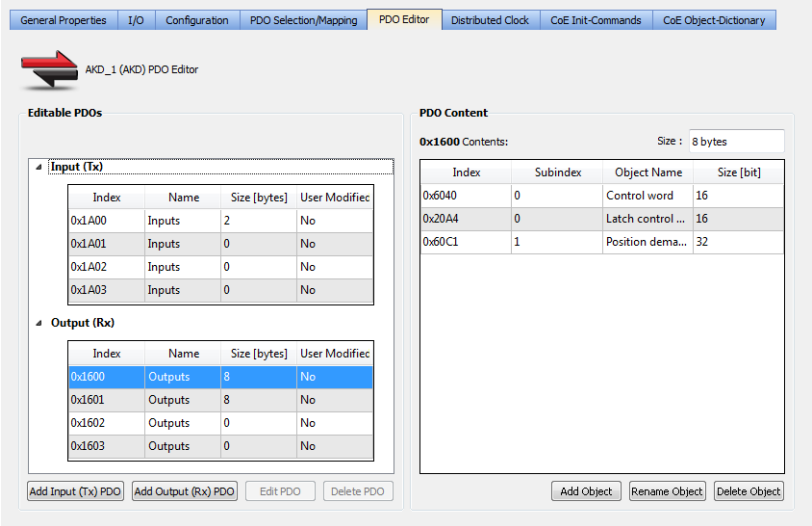


Figure 11-118: Example: PDO Content area

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Name	Description
Index	Identification value for the PDO. The range is defined by the EtherCAT specification. • Rx PDOs: 0x1600 to 0x17FF • Tx PDOs: 0x1A00 to 0x1BFF
Name	• Text label to identify the PDO. • The name is provided either by the device vendor or user. • It can be changed by the user.
Size [bytes]	Total length for the PDO object(s) data.
User Modified	• Yes: Indicates the user has changed the PDO or its content. • No: Indicates the PDO is the default provided by the device vendor.

NOTE

Constraints for the Kollmorgen AKD device family:
The PDO content **Size** field is highlighted in red if the Editable PDO content size is greater than 8 bytes.

8.13.5.2.1 Add a CoE Object

1. Select a PDO in the **Editable PDOs** section.
2. Click the **Add Object** button.
   The **Add CoE Object to PDO** dialog box opens. (Figure 11-119)

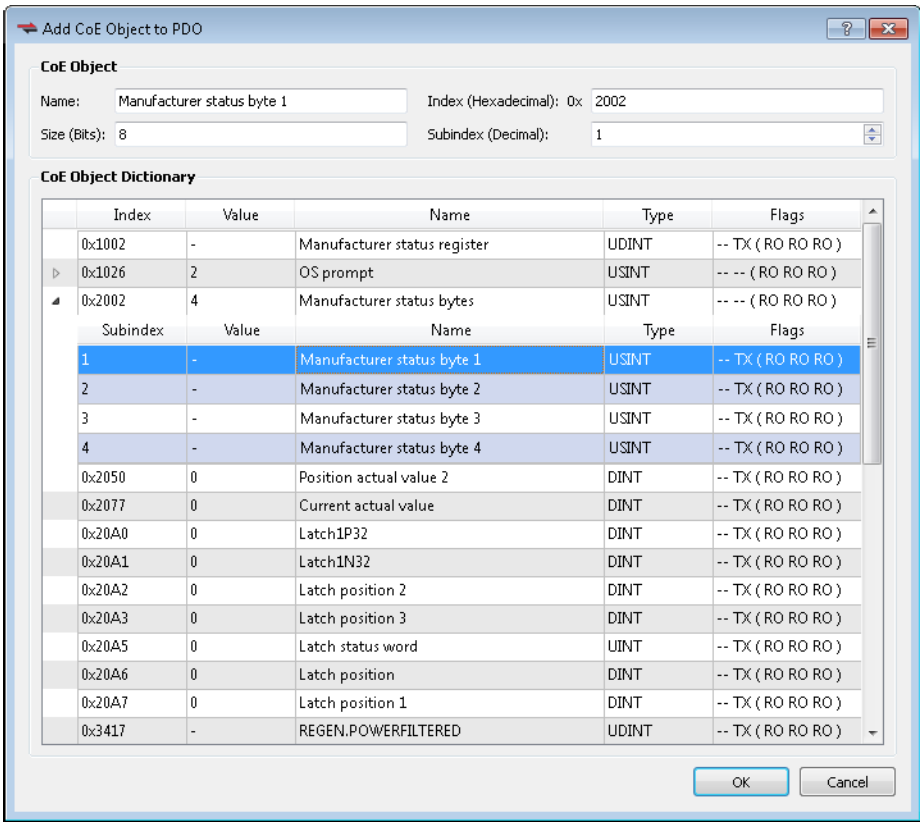


Figure 11-119: Add CoE Object to PDO dialog box

3. Either:
   - Select an object from the **CoE Object Dictionary** table.
   - Manually enter the **Name**, **Index**, **Size**, and **Subindex** data.
4. Click **OK** to save the changes or selections and close the dialog box.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

- The Flags indicate whether an object can be selected for an Input (Tx) or an Output (Rx) PDO.
- If the Flags are "-- --", the object is not eligible for the PDOs.
- Selecting the wrong object type generates an error.

8.13.5.2.2 Rename a CoE Object

1. Select a PDO from the **Editable PDOs** list.
2. Select the object to be renamed from the PDO Content.
3. Click the **Rename Object** button.
   The **Rename** dialog box opens. (Figure 11-120)
4. In the **Rename PDO object** text box, enter the new name.
5. Click **OK** to save the changes or selections and close the dialog box.

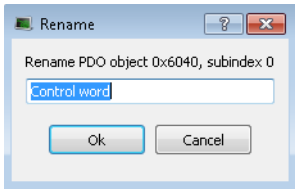


Figure 11-120: Rename dialog box

8.13.5.2.3 Delete a CoE Object

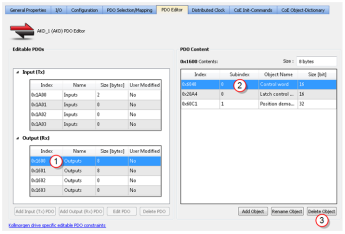


Figure 11-121: PDO Editor - Editable PDOs and PDO Content areas

1. In the **Editable PDOs** list, select a PDO.
2. In the PDO Content list, select the object to delete.
3. Click the **Delete Object** button.
4. Confirm the deletion.

8.13.5.3 PDO Restrictions and Compiler Errors

- "Restrictions for AKD Devices" (➔ p. 565)
- "Restrictions for AKD2G Devices" (➔ p. 566)
- "Restrictions for Kollmorgen Essentials Drive Devices" (➔ p. 566)
- "Restrictions for All Other Devices" (➔ p. 567)

8.13.5.3.1 Restrictions for AKD Devices

The KAS-IDE compiler enforces these restrictions to the editable PDOs for AKD devices:

- The maximum byte count for the content of an individual PDO (both selected and unselected) is 8 bytes.
  - The PDO content **Size** field is highlighted in red if the Editable PDO content size is greater than 8 bytes.
- The total maximum byte count for the content of all Tx PDOs (both selected and unselected) is 32 bytes.
- The total maximum byte count for the content of all Rx PDOs (both selected and unselected) depends upon the firmware.

Firmware Version	Byte Limit	Notes
v1-17 or later	32 bytes	This is the byte limit when the EtherCAT parameter ECAT.LEGACYREV is set to 0 (zero).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Firmware Version	Byte Limit	Notes
v1-16 or later	20 bytes	This is the byte limit when the EtherCAT parameter ECAT.LEGACYREV is set to 1.
v1-15 or earlier	20 bytes	

- The total byte count for the content of all selected Tx PDOs must be **even**.
  - This does not apply to AKD2G devices.
- The total byte count for the content of all selected Rx PDOs must be **even**.
  - This does not apply to AKD2G devices.

TIP

The object 0x00 can be used to pad the PDO to get the total byte count to an even value.

IMPORTANT

Compiler errors are generated if these restrictions are not met.

These are example compiler errors:

```
In AKD_1, the size of Rx PDO 0x1600 is greater than the limit: 9 > 8
```

```
In AKD_1, the total byte count for all editable Rx PDOs is greater than the
limit: 39 > 32
```

```
In AKD_1, the total byte count for all editable Rx PDOs that are assigned is
not even: 29
```

8.13.5.3.2 Restrictions for AKD2G Devices

The KAS-IDE compiler enforces these restrictions to the editable PDOs for AKD2G devices:

- PDOs have a size constraint of 128 bytes per PDO.
- The total maximum byte count for the content of all Rx PDOs (both selected and unselected) is 512 bytes.
  - [(128 per PDO) x 4 = 512 bytes].
- The total maximum byte count for the content of all Tx PDOs (both selected and unselected) is 512 bytes.
  - [(128 per PDO) x 4 = 512 bytes].
- Based on the Sample Rate / Cycle Time, there is a limitation to the number of PDOobjects:

Sample Rate	Cycle Time	Max. PDOs
4kHz	250µs	14 Rx, 14 Tx
2kHz	>500µs	30 Rx, 30 Tx

8.13.5.3.3 Restrictions for Kollmorgen Essentials Drive Devices

The KAS-IDE compiler enforces these restrictions to the editable PDOs for Kollmorgen Essentials Drive devices:

- PDOs have a size constraint of 128 bytes per PDO.
- Rx PDOs
  - The maximum number of Rx PDOs supported is 14.
  - The total maximum byte count for the content of all Rx PDOs (both selected and unselected) is 512 bytes.
    - [(128 per PDO) x 4 = 512 bytes].

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Tx PDOs
  - The maximum number of Tx PDOs supported is 14.
  - The total maximum byte count for the content of all Tx PDOs (both selected and unselected) is 512 bytes.
    - [(128 per PDO) x 4 = 512 bytes].

8.13.5.3.4 Restrictions for All Other Devices

The KAS-IDE compiler enforces the following restriction to the editable PDOs for all devices:

- The maximum byte count for the content of an individual PDO (both selected and unselected) is 64 bytes.

Compiler errors will be generated if these restrictions are not met. Example compiler errors:

```
In Device_1, the size of Tx PDO 0x1A16 is greater than the limit: 66 > 64
```

8.13.5.4 Redundant PDO Entries

It is possible in the KAS-IDE to have the same PDO entry occur in multiple PDOs (or even in a single PDO).

- A PDO entry is uniquely defined by an Index and a Subindex.
- The name of the entry is irrelevant for identification.

In Figure 11-122:

- The entry defined by Index `0x6040`, Subindex`0`, appears in the Rx PDOs `0x1600` and `0x1604`.
- The entry `0x3470` and subindex `4` appear in two different PDOs.

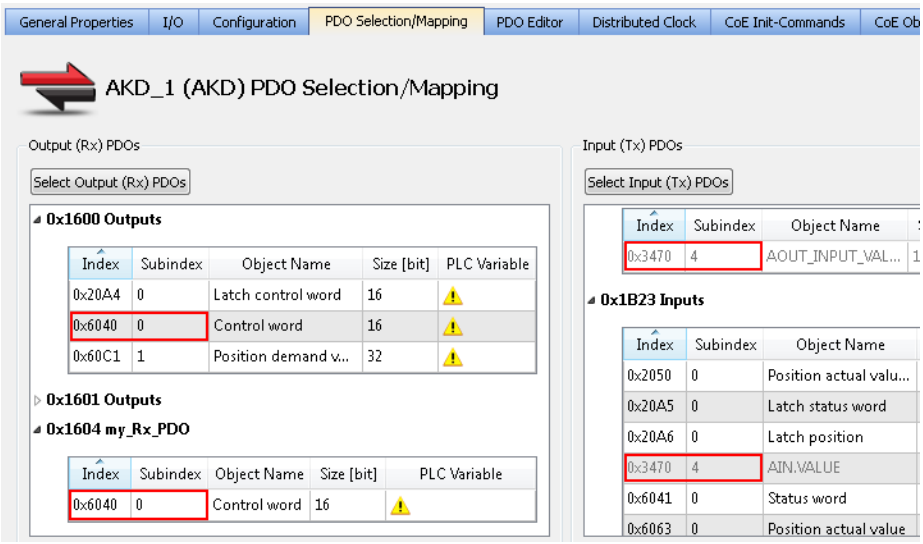


Figure 11-122: Example: PDO redundant entries

8.13.5.4.1 Redundant Rx PDO

The redundant Rx PDO would generate this compiler error:

```
EtherCAT: ERROR: In AKD_1, PDO object index 0x6040, subIndex 0 is
redundant. It is in the following PDO(s): 0x1600, 0x1604
```

Attempting to export the ENI file fails with this error:

```
Failed to export ENI file, check the logs for more details.
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8.13.5.4.2 Redundant Tx PDO

A redundant Tx PDO entry does not prevent a successful compilation nor generate an error when exporting the ENI file.

A redundant Tx PDO entry generates a compiler warning:

```
EtherCAT: Warning : In AKD_1, PDO object index 0x3470, subIndex 4 is
redundant. It is in the following PDO(s): 0x1A04, 0x1B23
```

8.13.5.5 Safety, PDOs, and Variables

This topic is about PDOs, safety objects, and variable mapping.

- Within the PDO Selection / Mapping tab any safety-related objects are clearly identifiable by being in a yellow table. (Figure 11-123)
- Mapping a variable to an object is easily done by any of these methods:
  - Double-click the object and select or create a variable.
  - Right-click the object, select Map, and select or create a variable.
  - Drag an existing variable from the Dictionary to the object.

Variables assigned to a safety object must be Read Only.

Variables that are Read/Write do not appear in dialogs and drag-and-drop does not work.

NOTE

- Once a variable is assigned, there is no further validation until the project is compiled.
- That means a variable could be changed to Read/Write and still be assigned.
- When you compile the project an error is generated.

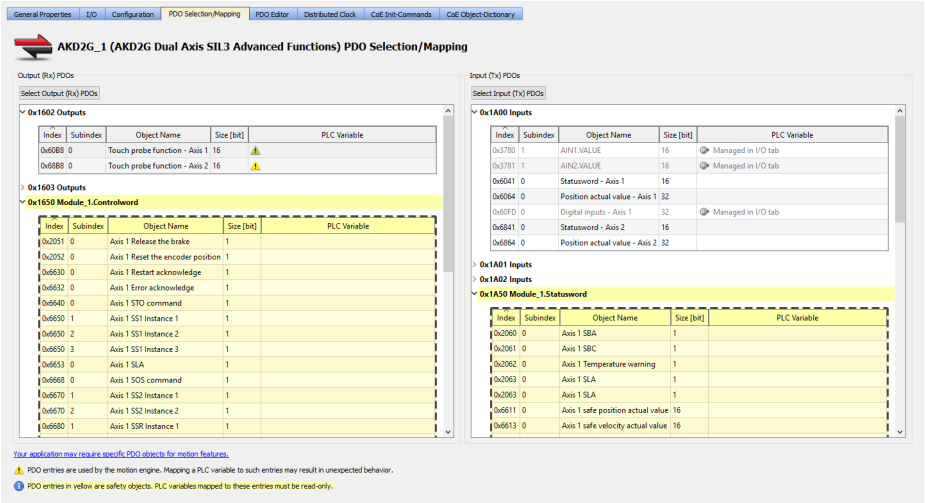


Figure 11-123: PDO Selection/Mapping tab with FSoE objects (in yellow).

See "Set Up a Variable to Monitor Safety Data" (➔ p. 558).

8.13.6 Distributed Clock tab

This tab is used to change settings related to the Distributed Clock for all Kollmorgen and third-party devices (both discovered and manually added).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

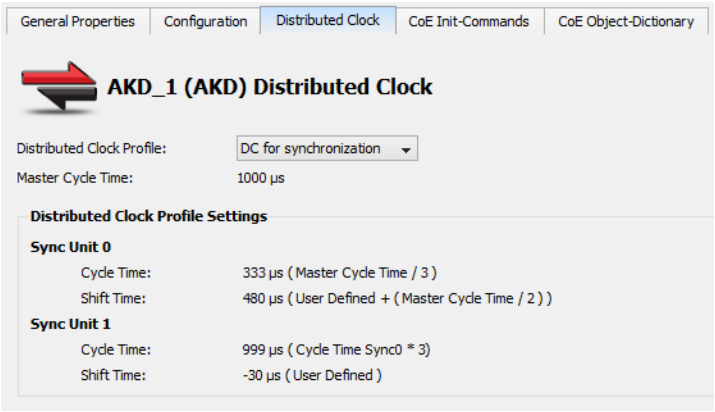


Figure 11-124: Distributed Clock tab

Element	Description
Distributed Clock Profile	Select the Distributed Clock (DC) operation mode. These modes cannot be edited. **FreeRun, DC unused, SM-Synchronous** • Device cyclic operation is triggered by the EtherCAT master's SyncManager via the cyclic frame. • Devices are **not** synchronized by the reference clock. **DC-Synchron, DC for synchronization, DC-Synchronous** • Device cyclic operation is triggered by the local distributed clock. • Devices are synchronized by the reference clock. **DC-Synchron (input based)** • As DC-Synchron mode, with the cyclic start time so it coincides with other input slaves in the EtherCAT system.
Master Cycle Time	Base interval in microseconds and is used by the master. This is changed and automatically updated by changing the **Cycle Time** value on the "EtherCAT Master Settings tab" (→ p. 539) tab.
Sync Unit 0	**Cycle Time** • Sync Unit Cycle: Unit is synchronized relative to the Master Cycle Time • User defined: Unit has its own interval **Shift Time** • Unit is adjusted by the shift time
Sync Unit 1	**Cycle Time** • Sync Unit Cycle: Unit is synchronized relative to the Master Cycle Time • Sync 1 Cycle: Unit is synchronized relative to the First Sync Unit • User defined: Unit has its own interval **Shift Time** • Unit is adjusted by the shift time

NOTE

Some or none of the content is available in these scenarios:

- The Sync 0 or Sync 1 parameter is not present.
- Distributed Clock is not supported by the device.
- The ESI file is missing.

8.13.6.1 Oversampling Devices

Some EtherCAT devices have oversampling features.

- An oversampling device is able to record (input) or provide (output) signals at a higher rate than the EtherCAT cycle time.
- This rate is called the **oversampling factor**.
- Example: Using an oversampling factor of 10 and an EtherCAT cycle time of 1ms (1Khz), an input device can record values every 100µs (1000 divided by 10).

Oversampling devices have as many PDO objects in their cyclic frames as the oversampling factor to achieve the higher rate.

- Each of these PDO objects corresponds to one sample.
- Example: An output device with an oversampling factor of 4 has four PDO objects:
  - Output 1, Output 2, Output 3 and Output 4.

The oversampling factor is tied to the Distributed Clock Profile.

A warning appears next to the DC profile selection box in the Distributed Clocks tab when a device has oversampling features. (Figure 11-125)

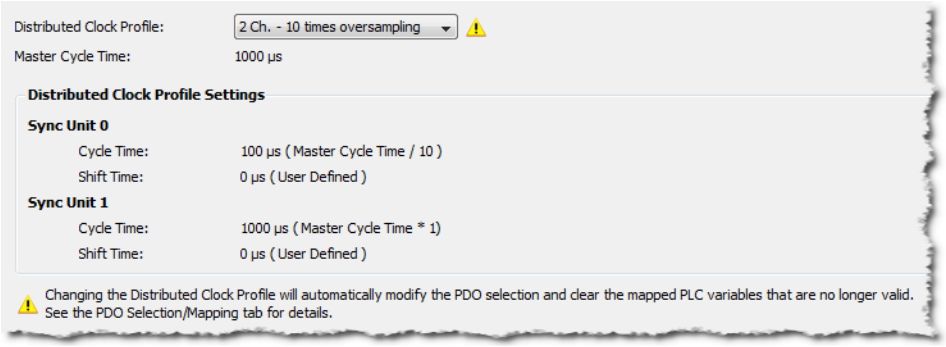


Figure 11-125: Warning message of device with oversampling.

Changing the Distributed Clock Profile automatically changes the PDO selection.

- All PDOs corresponding to the selected Distributed Clock Profile and its according oversampling factor are selected.
- These maps are discarded if one or several PLC variables were mapped to a PDO that is no longer selected.

8.13.7 CoE Init Commands tab

This tab lists the EtherCAT device's CoE Init Commands.

- The Init commands are grouped based on the EtherCAT transition state. (Figure 11-126)
  - The sequence within each group defines the order in which the commands are executed on the device.
- The order of CoE Init Commands can be changed to define the correct command sequence required for device operation.
- Selecting a command enables buttons to move a command Up, Down, to the Top of the table, and the Bottom of the table.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

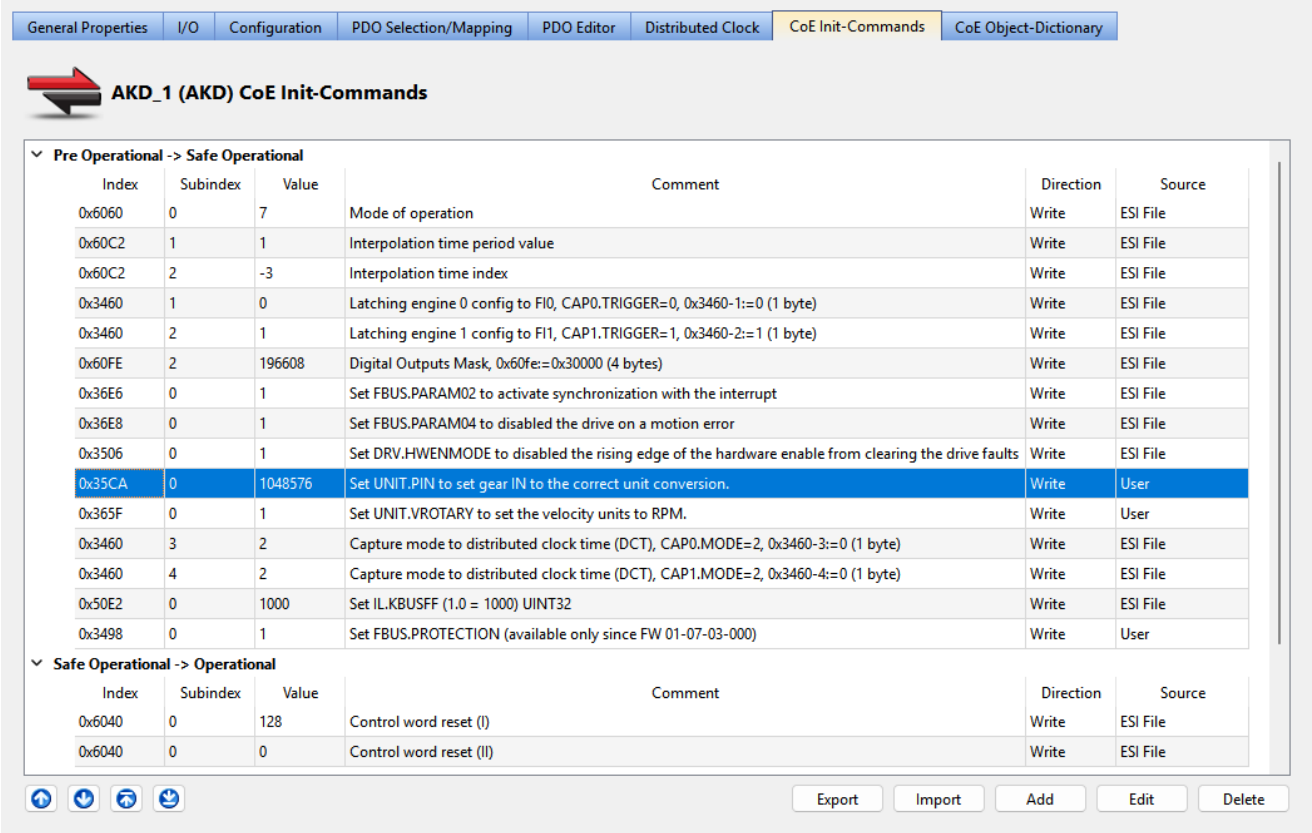


Figure 11-126: CoE Init Commands tab

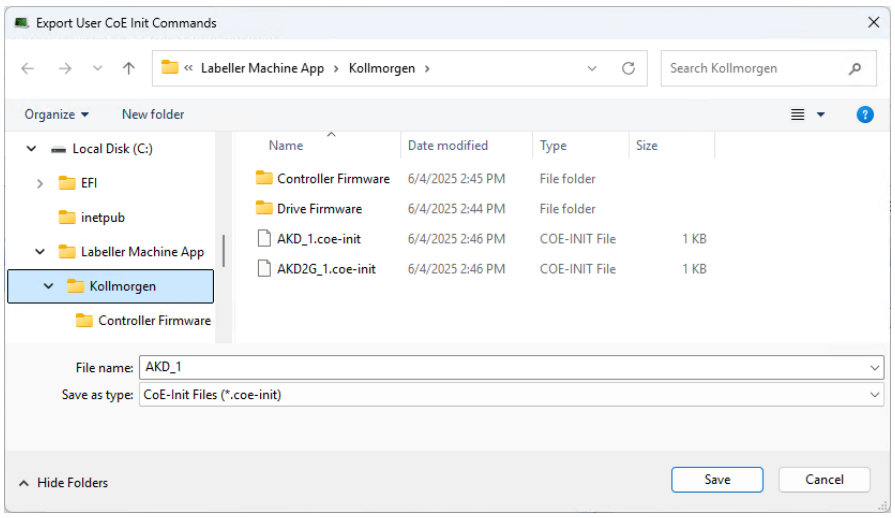
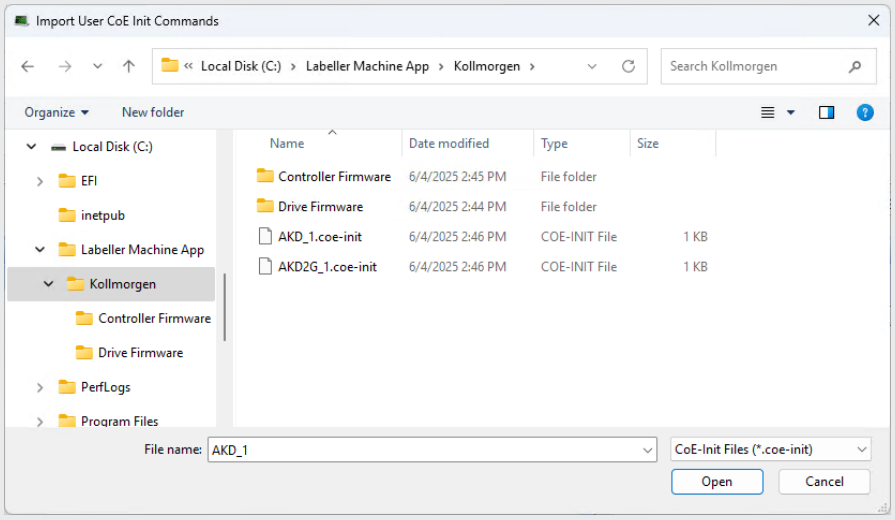
NOTE

- This tab is grayed out if the device does not support the CoE Protocol or the ESI file is unavailable.
- Fixed Init-commands have been deprecated and are not listed.

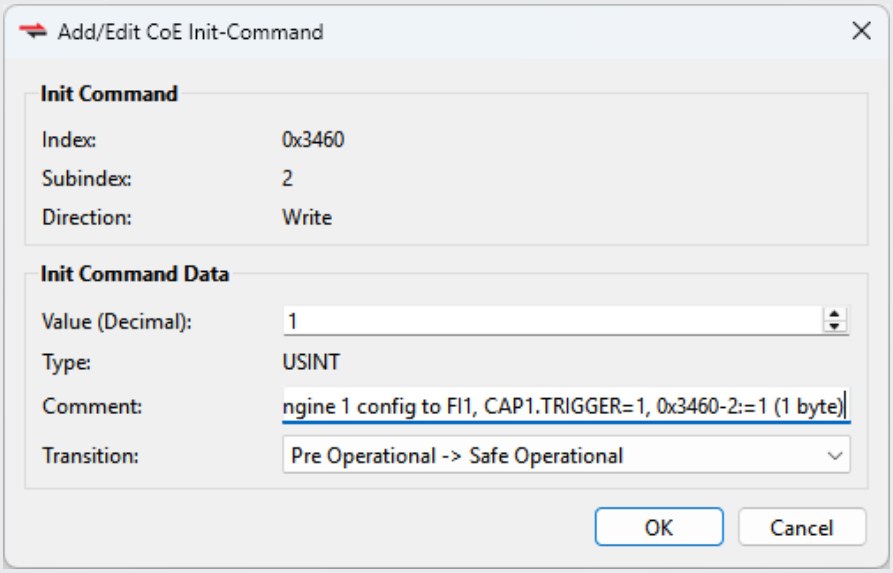
Column	Description
Index	The hex value of the CoE-Index.
Subindex	The CoE-Subindex.
Value	Value of the init command.
Comment	Description of the init command.
Direction	Specifies if the command is **Read** or **Write**.
Source	The source of the init command. These are the values: • **ESI File** denotes that the command comes from the ESI file. • **User** denotes that the command is created by the user.

8.13.7.0.1 Functions

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Button	Description
Export	Click to open the **Export User CoE Init Commands** dialog box. (Figure 11-127)  **Figure 11-127: Export User CoE Init Commands** dialog box. • Use the dialog box to export the user-configured CoE init commands to a file. • This file can be used to import new commands into other EtherCAT devices. • User-configured commands are those that show up in the Source column as User.
Import	Click to open the **Import User CoE Init Commands** dialog box. (Figure 11-128)  **Figure 11-128: Import User CoE Init Commands** dialog box • Use the dialog box to import CoE init commands from a file. • If the init command already exists for the device, it is overwritten with values from the imported file. • New and overwritten commands are marked as user-configured. • In the **Source** column of the CoE Init Commands tab, these commands appear as **User**.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Button	Description
Add	Click to open the **Add/Edit CoE Init-Command** dialog box. (Figure 11-129) • Use the dialog box to create a new Init Command. • See "Add CoE Commands" (→ p. 573).
Edit	• Select an existing Init Command and click the **Edit** button or double-click the selection to edit the Init Command's attributes. • The dialog box is used to edit the command data including the Value, Comment, and Transition.  **Figure 11-129: Add/Edit CoE Init-Command** dialog box
Delete	Click to remove the selected Init Command row.

8.13.7.1 Add CoE Commands

This command is used to construct a new Init Command by supplying Index, Subindex, Value, Comment, Transition, and Direction attributes.

- The dialog lists the device's object dictionary.
- The objects are filtered based on the selected in the **Direction** drop-down.
  - Example: If Write is selected, the Read-only objects are not shown. (Figure 11-130)

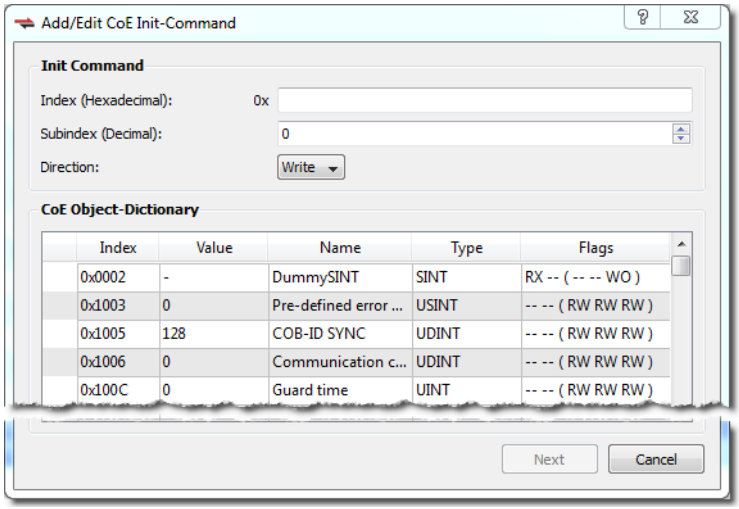


Figure 11-130: Direction drop-down selection: Write

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

- The **Index** field accepts four characters at most.
- The **Subindex** has a range of 0 - 255.
- When setting the Direction you may select **Read** or **Write**.
- Selecting an item in the CoE Object-Dictionary auto-populates the **Index** and **Subindex** entries.

The **Next** button is available once the **Index** and **Subindex** fields have values.

If the Init Command is already present in the dictionary:

If the Init Command is already in the dictionary (i.e., the Index and Subindex specified are already used), the dialog updates to show the **Init Command Data** area.

Use this area to specify the **Value**, a **Comment**, and the **Transition**. (Figure 11-131)

- If the **Type** is STRING (20) or STRING (50), only 20 or 50 characters can be entered, respectively.
- The **Transition** options are:
  - **Pre Operational -> Safe Operational**.
  - **Safe Operational -> Operational**.

Click **OK** to add the command to the appropriate transition group in the "CoE Init Commands tab" (→ p. 570).

This sets the Source attribute to **User**.

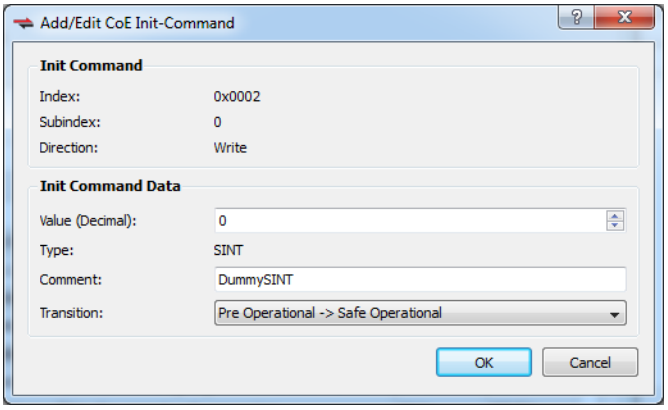


Figure 11-131: Init Command Data area

If the Init Command data type is an array of bytes, all the bytes can be entered in HexBinary format in the **Value** text box.

The size of the array determines the maximum data that can be entered. (Figure 11-132)

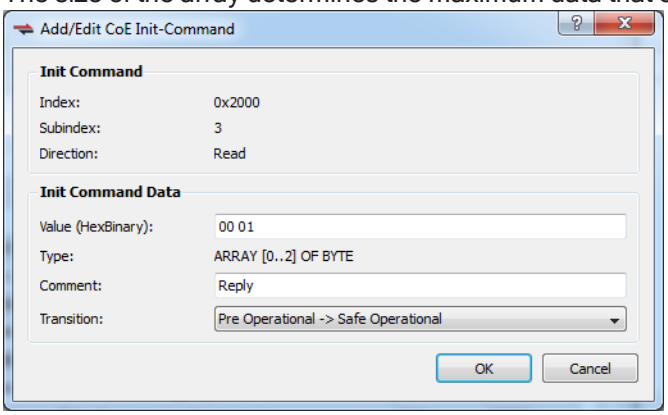


Figure 11-132: Value text box

If the Init Command is not present in the dictionary:

Adding a new command (i.e., the Index and Subindex are not in the dictionary), the dialog updates to show the **Init Command Data** area.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Use this area to specify the **Value**, a **Comment**, and the **Transition**.

- The **Value** must be entered in HexBinary format. (Figure 11-133)
- The **Transition** options are:
  - **Pre Operational -> Safe Operational**.
  - **Safe Operational -> Operational**.

Click **OK** to add the command to the appropriate transition group in the "CoE Init Commands tab" (→ p. 570).

This sets the Source attribute to **User**.

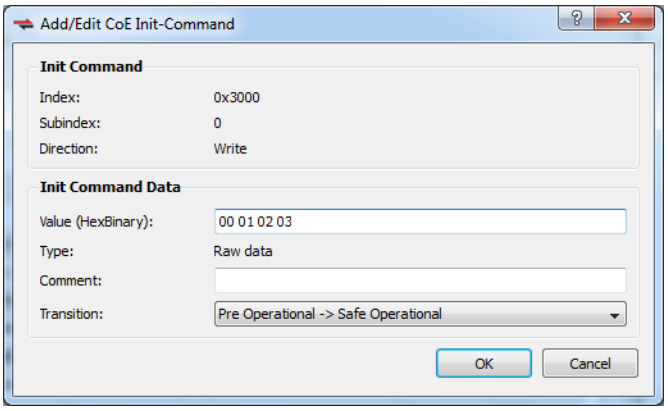


Figure 11-133: Value text box

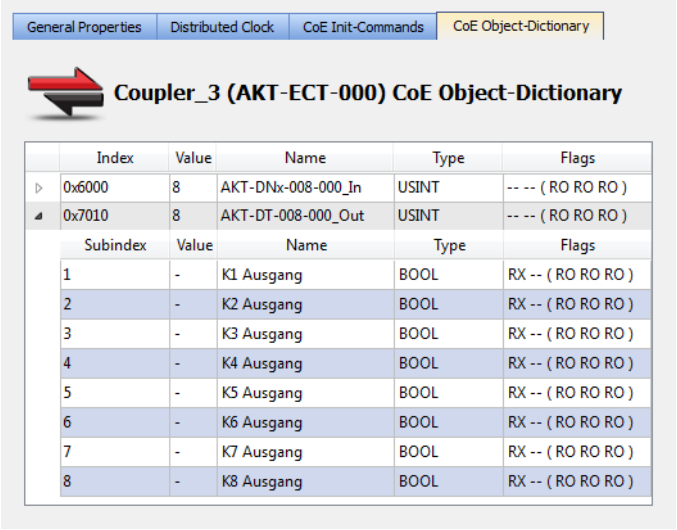
8.13.8 CoE Object Dictionary tab

This tab displays all the CoE (CAN over EtherCAT) objects associated with the EtherCAT device.

- It is used as reference to add new CoE Init Commands for the EtherCAT device.
- The CoE objects can be used for three different actions, depending upon the access flags.
  - PDO mapping for cyclic communication during application execution.
  - CoE Init commands during application start-up.
  - SDO communication during application execution.

IMPORTANT

This tab is grayed out if the ESI file does not have CoE object information or if the file is unavailable.



Index	Value	Name	Type	Flags
0x6000	8	AKT-DNx-008-000_In	USINT	-- -- (RO RO RO)
0x7010	8	AKT-DT-008-000_Out	USINT	-- -- (RO RO RO)
Subindex	Value	Name	Type	Flags
1	-	K1 Ausgang	BOOL	RX -- (RO RO RO)
2	-	K2 Ausgang	BOOL	RX -- (RO RO RO)
3	-	K3 Ausgang	BOOL	RX -- (RO RO RO)
4	-	K4 Ausgang	BOOL	RX -- (RO RO RO)
5	-	K5 Ausgang	BOOL	RX -- (RO RO RO)
6	-	K6 Ausgang	BOOL	RX -- (RO RO RO)
7	-	K7 Ausgang	BOOL	RX -- (RO RO RO)
8	-	K8 Ausgang	BOOL	RX -- (RO RO RO)

Figure 11-134: CoE Object Dictionary tab

CoE objects can be composed of a simple data type or a complex type.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- In case of a complex data type, each simple data type composing it has a Subindex.
- Figure 11-135 shows the CoE object can be expanded to show all the Subindex's details that are the same as the details for a simple type CoE object.

	0x10F3	Diagnostics: History buffer	5	USINT	-- -- (RO RO RO)
	Subindex	Name	Value	Type	Flags
	1	Max. number of messages	32	USINT	-- -- (RO RO RO)
	2	Latest message	0	USINT	-- -- (RO RO RO)
	3	Latest acknowledged message	0	USINT	-- -- (RW RW RW)
	4	New active message	0	USINT	-- TX (RO RO RO)
	5	Control bits	0	UINT	-- -- (RW RW RW)

Figure 11-135: CoE table - Subindex column

8.13.8.0.1 CoE Table Descriptions

Field	Description
Index / Subindex	CoE Index or Subindex number.
Name	Name of the CoE object.
Value	Default value of the CoE object.
Type	Data type of the CoE object.
Flags	This column contains five values in the format **XX YY (AA BB CC)**, each representing an access option. The values of XX and YY provide PDO mapping options: XX Specifies if the CoE object can be mapped as RxPDO. It is represented as **RX** if it can be mapped. YY Specifies if the CoE objecct can be mapped as TxPDO. It is represented as **TX** if it can be mapped. The values for AA, BB, and CC provide the CoE Access type. The values can be read only (**RO**), read-write (**RW**), or write only (**WO**): AA Access type of the object in PREOP state. BB Access type of the object in SAFEOP state. CC Access type of the object in OP state.
Module	Name of the module that is associated with the CoE object.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

9 Controller Error and Alarm Descriptions

When either "Errors" (→ p. 577) or "Alarms" (→ p. 583) occur, always check the controller log messages.

- The controller continuously displays any error or alarm codes after booting and not in recovery mode.
- Errors have a priority over Alarms.
  - Only one error or alarm code is displayed at a time.
  - The code with the highest priority is displayed until it is cleared.
- The log messages provide more details about the failure and the history of events leading up to the failure.
  - Use the log messages to determine the specifics about the cause of the failure and correct the underlying problem.

9.1 Errors

NOTE

- **In the **How to Clear** column:
  - **HW** indicates a reboot is required to clear the error or alarm.
  - **SW** indicates the error or alarm can be cleared from the web server.
  - **APP** indicates the error or alarm is automatically cleared when the application is restarted.

Code	Description	Cause	Remedy	How to Clear**
E01	PCMM2G: • After 20 seconds, the PLC application is stopped. • The CPU is throttled. • Critical temperature reached. • Critical temperature limit is 83 °C.	Controller temperature exceeds the safe operating temperature limit.	• Power-off. • Check airflow and operating environment are within hardware specifications. • Allow unit to cool before power-on.	• SW: • PCMM2G

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Code	Description	Cause	Remedy	How to Clear**
E02	Out of memory. KAS Runtime is stopping.	Memory leak, memory corrupted, or hardware memory failure.	• Power-off/on. • If the problem is recurrent, check the Release Notes for firmware updates. • If the problem persists, return the hardware for repair.	HW
E03	Fan failure.	CPU cooling fan is not able to operate properly.	• Check temperature and monitor for High temp alarm (see A01). • Return the hardware for fan replacement.	• SW: • PCMM2G
E10	Firmware is corrupted.	Flash memory corrupted during firmware download or flash hardware failure.	• Re-download firmware. • Boot into recovery mode, download firmware, and power-off/on. See "Recover Using the Web server" (→ p. 672) for more information. • If the problem persists, return the hardware for repair.	SW
E12	Not enough flash memory available.	Flash memory is full, unable to write to flash.	Clean-up the flash memory by removing log files, application programs, recipes, or other data files.	SW
E13	Out of NVRAM space to "Retain Variables" (→ p. 292).	NVRAM is full.	Change application to reduce the amount of retained variables.	SW
E15	Cannot read / write files from/to a USB flash drive.	• PLC function failures do **not** cause this error. • USB flash drive is not plugged in. • The file system is corrupt and cannot be mounted.	1. Plug-in a valid USB flash drive. 2. Reformat the USB flash drive using the Settings > USB Flash Drive > Format button.	SW
E16	Not enough space available on the USB flash drive.	• USB flash drive is full, unable to write to the USB flash drive. • PLC function failures.	• Clean-up the USB flash drive space by deleting files. • Reformat the USB flash drive using the Settings > USB Flash Drive > Format button.	SW

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Code	Description	Cause	Remedy	How to Clear**
E17	Cannot connect to a shared directory.	• Improper shared directory configuration on remote computer or the controller. • Improper permissions, access control, or security policy for computer's share configuration. • Computer is not accessible via Ethernet network.	• Verify the computer's shared directory is accessible. • Check and Apply the KAS Application > Shared Directory configuration. • Power-off/on the controller.	SW
E20	Runtime plug-in, process, thread or application failed to start.	• KAS Runtime or application code failed to auto-start at boot. • Caused by either a high controller temperature at power up or a system error.	1. Check the web server for other alarms or errors (e.g., temperature alerts) and resolve them. 2. Power cycle the device. 3. If E20 is still reported, reset to factory defaults. 4. If the problem is recurrent, check the Release Notes for firmware updates. 5. Download firmware.	HW
E21	Runtime process, thread, or driver failed to respond during operation.	• KAS Runtime code failed during normal operation. • In extreme cases, the display may indicate that the controller failed at an early stage of the "Boot Sequence" (➔ p. 596).	• Power-off/on. • If the problem is recurrent, check the Release Notes for firmware updates.	HW
E22	Fatal error in PLC program, application stopped.	Virtual machine failed to execute an instruction.	• Check the PLC application code for programming errors that could cause a fatal execution error (e.g., divide by zero, array out of bound, etc.) • Check that the IDE and controller Runtime versions are compatible. • Re-compile the application, download, and re-start.	SW
E23	CPU is overloaded.	Either: • The motion engine did not complete. • The PLC program did not complete within the timeout period due to excessive CPU load.	• Stop the application. • Power-off/on. • Reduce the sample rate, simplify the application, or reduce the application cycles and restart the application. • See "CPU Overload (E23)" (➔ p. 585).	SW

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Code	Description	Cause	Remedy	How to Clear**
E24	PLC application cannot be started	PLC application cannot be started because of an existing condition. Possible reasons: • Maintenance operation is in progress. • Controller is in "Online Configuration Mode" (➔ p. 471). • AKD Restore failed. • The IDE version of the compiled PLC code and controller runtime version do not match. • Previous download failed.	1. Check the controller web-server home page for any maintenance operation in-progress. Wait for the operation to finish. 2. Connect to the controller with the IDE and disable "Online Configuration Mode" (➔ p. 471). 3. Check the EtherCAT network topology by using the Scan network button in the web-server's Restore tab. Correct the physical topology and re-execute an AKD-x restore. 4. The IDE version (only major.minor.micro) should match with runtime version. To correct, install the correct version of IDE or Runtime. 5. Connect the IDE and download application.	SW
E30	EtherCAT communication failure during operational mode.	Network communication error. • This is commonly caused by a loose or bad EtherCAT connection occurring after the network successfully starts up. • See the IDE Controller log for more information.	• Read the AKD-x parameter ECAT.DIAG for more information on where in the EtherCAT network the issue may be occurring. • See Debugging Intermittent EtherCAT Communication Issues on KDN for more information. • Check the EtherCAT network wiring and devices state. • Re-start the application.	SW
E31	EtherCAT communication failure during pre-op mode.	EtherCAT network operation failed due to a network communication error.	• Check the EtherCAT network wiring and devices state. • Re-start the application.	SW
E32	EtherCAT communication failure during bootstrap mode.	EtherCAT network operation failed due to a network communication error.	• Check the EtherCAT network wiring and devices state. • Re-start the application.	SW

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Code	Description	Cause	Remedy	How to Clear**
E33	EtherCAT failed to initialize into operational mode.	EtherCAT network initialization failed. • This is commonly caused by the EtherCAT configuration in the project not matching the actual hardware. • See the IDE Controller log for more information.	• Check the EtherCAT network wiring and devices state. • Re-start the application. • See "Why am I getting an E33 error?" (➔ p. 788).	SW
E34	EtherCAT failed to initialize into pre-op mode.	EtherCAT network initialization failed due to a network communication error.	• Check the EtherCAT network wiring and devices state. • Re-start the application.	SW
E35	EtherCAT failed to initialize into bootstrap mode.	EtherCAT network initialization failed due to a network communication error.	• Check the EtherCAT network wiring and devices state. • Re-start the application.	SW
E36	EtherCAT failed to discover the expected devices.	EtherCAT network discovery failed due to a mismatch between the discovered and expected devices.	• Check the EtherCAT devices and wiring order. • Correct the device order wiring or re-scan the network, re-compile, and download the updated application. • Re-start the application.	SW
E37	EtherCAT failed to return to init state.	EtherCAT network initialization failed due to a network communication error.	• Check the EtherCAT network wiring and devices state. • Re-start the application.	SW
E44	Software features license data is corrupted.	• The detected software feature license stored on the controller is corrupted. • See the Controller logs for more information.	If Step 1 is not successful continue to Step 2. 1. Power-off/on. 2. If the problem is recurrent, check the Release Notes for firmware updates. 3. If the problem persists, return the hardware for repair.	HW

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Code	Description	Cause	Remedy	How to Clear**
E45	Axes count exceeded license limit.	The number of axes defined in the application is greater than the maximum number permitted by the controller model.	• Review the system configuration. • Remove any axes beyond the licensed limit. • If additional axes are required, consider upgrading the controller to a higher-tier model (e.g., 8 axes to 24). • See "Controllers" (→ p. 767) for available models.	APP
E50	Backup to USB flash drive failed.	An unrecoverable error occurred during the backup operation.	• Repeat the backup to USB flash drive operation. • If it fails again, replace the USB flash drive.	SW
E51	Restore from USB flash drive failed.	An unrecoverable error occurred during the restore operation.	Do not reboot the Controller! • Repeat the restore operation. • If it fails again, reset the Controller to factory defaults. • If the problem persists, return the hardware for repair.	SW
E52	USB flash drive Backup files are missing or corrupt.	The restore operation failed due to missing, incomplete, or corrupt files on the USB flash drive.	• Perform a backup operation before the restore. • Use a USB flash drive with valid backup files.	SW
E60	Inactive partition is corrupted.	An unrecoverable error occurred when starting KAS Runtime on the inactive partition.	These are the recovery steps. If Step 1 is not successful, go to Step 2 and continue to Step 3 if needed. 1. Re-download firmware. 2. Boot into recovery mode, download firmware, and power-off/on. See "Recover Using the Web server" (→ p. 672) for more information. 3. If the problem persists, return the hardware for repair. See "Booting the PCMM2G Controller" (→ p. 593) for more information about partitions.	SW

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

9.2 Alarms

NOTE

- **In the **How to Clear** column:
  - **HW** indicates a reboot is required to clear the error or alarm.
  - **SW** indicates the error or alarm can be cleared from the web server.
  - **APP** indicates the error or alarm is automatically cleared when the application is restarted.

Code	Description	Cause	Remedy	How to Clear**
A01	High temperature exceeded.	CPU temperature near the safe operating temperature limit for: • PCMM2G = 78 °C	Check airflow and operating environment are within hardware specifications.	SW
A02	Low on memory.	Memory leak or corruption.	• Power-off/on. • If the problem is recurrent, check the Release Notes for firmware updates. • If the problem persists, return the hardware for repair.	SW
A04	Low input voltage.	+24 volt input power is +19 volts or less.	Check power supply voltage and connection to the controller.	SW
A05	PCMM2G only. The system time was not set on startup.	PCMM2G only. • Network Time Protocol (NTP) is disabled. • The time could not be retrieved at startup. • This is most likely caused by a dead or missing battery.	PCMM2G only. Either: • Insert a good battery and set the time. • Enable NTP. • See "Date/Time tab" (→ p. 662) for more information.	SW
A12	Flash memory is low on free space.	Flash memory is almost full.	• Clean-up the flash memory by removing log files, application programs, recipes, or other data files. • Reset to factory defaults.	SW

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Code	Description	Cause	Remedy	How to Clear**
A21	Recoverable process or thread failed to respond during operation.	KAS non-runtime code failed during normal operation and was automatically restarted.	• Power-off/on. • If the problem is recurrent, check the Release Notes for firmware updates.	SW
A23	CPU is heavily loaded.	CPU usage is too high for 5 (or more) seconds.	Reduce the sample rate, simplify the application, or reduce the application cycles.	SW
A30	EtherCAT missed a send frame during operation mode.	The EtherCAT master was unable to send a frame for one or more cycles.	Reduce the controller CPU load so it has enough Real-Time margin to send EtherCAT frames every cycle.	SW
A31	• EtherCAT received a frame with a wrong Working Counter value. • See "Actual Position Interpolation" (→ p. 584).	• The slave EtherCAT device does not exist or cannot be physically reached due to connector and/or cable failure. • Slave hardware failure • Slave is still busy with a previous command and is not ready.	• Check the EtherCAT cables and the connectors. • Check the EtherCAT devices for any device errors.	
A38	EtherCAT missed a receive frame during operation mode. See "Actual Position Interpolation" (→ p. 584).	The EtherCAT master did not receive, or received too late, a frame for one or more cycles.	• Read the AKD-x parameter ECAT.DIAG for more information on where in the EtherCAT network the issue may be occurring. • See Debugging Intermittent EtherCAT Communication Issues on KDN for more information. • Check the EtherCAT network wiring and devices state. • Decrease the EtherCAT cycle rate.	SW
A40	Local digital IO missed a cyclic update.	Local digital IO was not updated during a cycle or the updates are no longer synchronous.	Reduce the sample rate, simplify the application, or reduce the application cycles.	SW

9.3 Actual Position Interpolation

- If EtherCAT actual position data is missing due to a corrupted, late, or missing EtherCAT frame ("A31" (→ p. 584) or "A38" (→ p. 584)), the controller interpolates the actual position data for servo axes, digitizing axes, and pipe network sampler blocks.
- The interpolation reduces velocity discontinuities when the actual position data is used for camming and gearing.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

9.4 CPU Overload (E23)

If the Motion Engine or PLC program execution (VM) do not complete a full cycle within their respective timeout periods, an "E23" (➔ p. 579) error is flashed on the 7-segment display.

Process	Timeout
Motion Engine	200 milliseconds
PLC Program (VM)	10 seconds

The Real-Time operation for EtherCAT and the Motion Engine have the highest priority in the controller.

- The PLC VM has the second highest priority in the controller.
- These processes continue to execute, even if their timeout values are exceeded.

If the CPU overload is severe, there may not be enough CPU time to execute the background operations.

- The background operations include the 7-segment display update, monitoring push-buttons, web-server, Modbus, and communications with the KAS-IDE.
- The 7-segment display indicates a CPU overload or frozen software task using one of these patterns:

Display	Meaning	Applicable Controller
	CPU overload is extreme.	• PCMM2G

To recover from an E23, stop the application from the KAS-IDE or web-browser (KAS Application view).

- If the CPU overload is severe, the controller may not have enough CPU time to respond to the IDE or web-browser.
  - If so, power off/on the controller.
- If the controller is configured for Auto-start, press and hold the B3 menu button at boot-time to prevent the application from automatically re-starting.
  - Then, connect to the controller with the KAS-IDE.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

10 Using the KAS Simulator

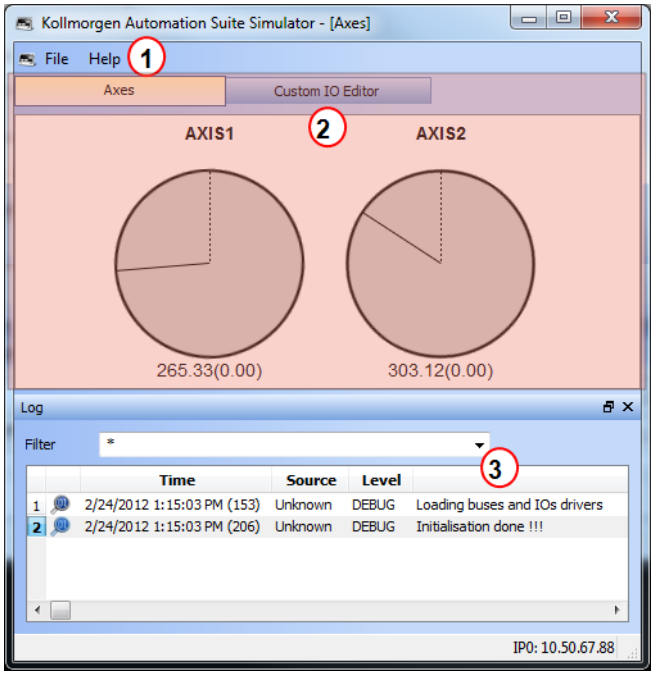


10.1 KAS Simulator Graphical User Interface

10.1.1 Main window

KAS Simulator main window contains:

1. The menu bar
2. The workspace
3. The "KAS Runtime Log window" (→ p. 590)



10.1.2 Log window

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This Log window shows all log messages related to the KAS Simulator.

Error and warning messages issued from the operating system, and "Printf Function" (➔ p. 272) instructions, are in this window.

Every log message includes:

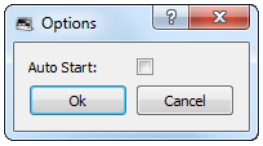
- Timestamp
- Source
- Logging Level
- Message

10.1.3 KAS Simulator - Menus

10.1.3.1 File Menu

Command	Description
Start	Start the application with the "Retain Variables" (➔ p. 292).
Cold Start	Start the application with the initial settings.
Stop	Stop the application.
Option	Set parameters for the KAS Simulator application (see explanations below).
Exit	Leave KAS Simulator application.

Option



Option	Description
Autostart	Autostart the application when KAS Simulator is launched. Autostart mode is recommended when a system is in production.

NOTE

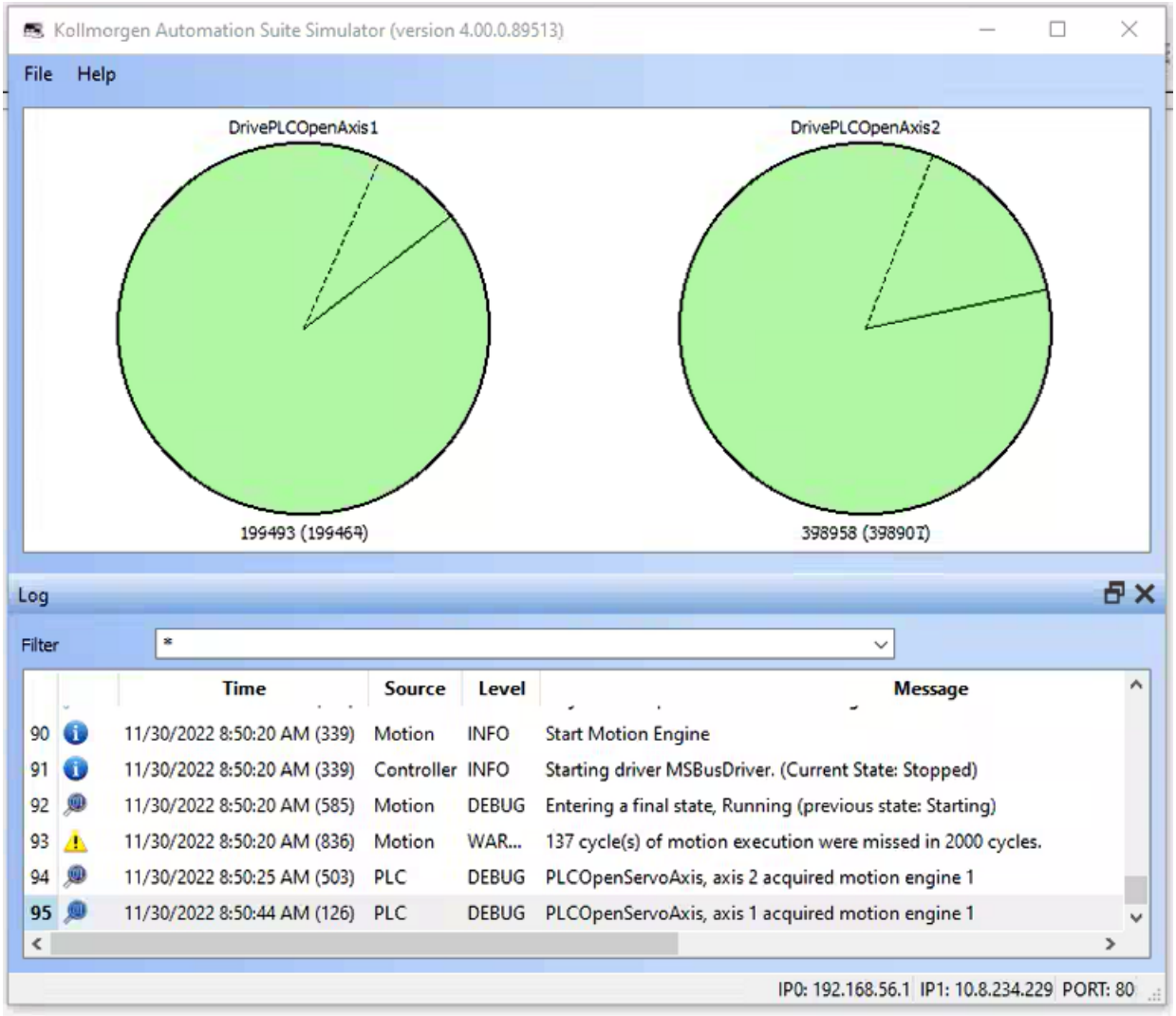
Parameters are saved in the **Options.bin** file located under:
<user>\AppData\Local\Kollmorgen\KAS\Sinope Simulator\Resources\

10.1.3.2 Help Menu

Command	Description
About	Show version numbers and other "Find the KAS-IDE Version" (➔ p. 151) about KAS Simulator

10.1.4 KAS Simulator window

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



- The dashed line (or dotted line) represents the Reference Position in User units.
- When the is solid line (or normal line) visible, it represents the Actual Position in User units.

Below the disk, the reference position for the associated axis is represented in this format:

```
Range value (Modulo value according to the periodicity)
```

In this image, the **Error** command (in the contextual menu of the axis tab) is used to simulate an error on an axis. Notice the impact on the HMI and implement counter-measures if necessary.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

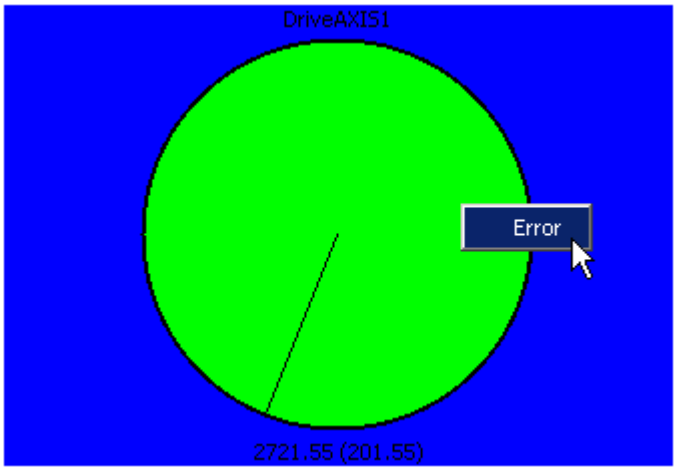


Figure 13-1: Set Axis in **Error** Mode

The drive becomes Red when it is set to **Error** .

See also see "Motion State Machine" on page 323 for more information.

To deselect an axis already selected (blue rectangle), click in the white surrounded outside border of the axis tab.

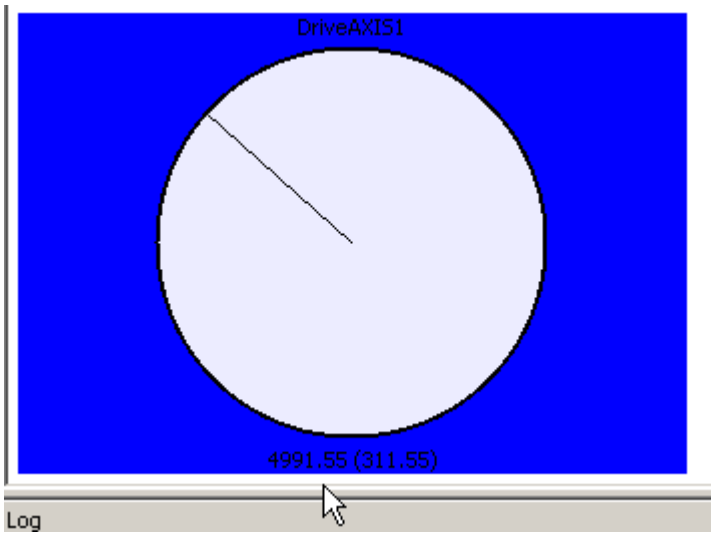
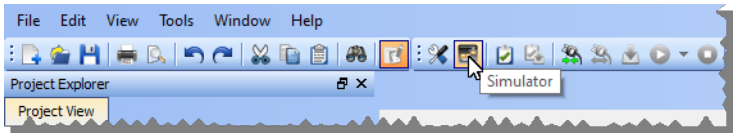


Figure 13-2: Deselect an Axis

10.2 Start the KAS Simulator

The **KAS Simulator** is launched from within the KAS-IDE.

Click the KAS Simulator button in the toolbar.



Click the button to:

- Change the KAS project to use KAS Simulator.
- Connect the KAS-IDE to the KAS Simulator.
- Launch the KAS Simulator.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TIP

Each version of KAS-IDE has a matching version of Simulator which it opens.
Only one instance of KAS Simulator may be run at a time.

The first time KAS Simulator is run, it attempts to open some TCP/IP ports to allow communication.

The system's firewall detects this and prompts for an action.

The KAS Simulator normally uses port 80 for the web server.

- If another service is using port 80, then KAS Simulator automatically assigns a free port available to the webserver to listen to the incoming HTTP requests.
  - The port number assigned for the Simulator web server is shown in the status bar.
- If an alternate port is assigned to the web server in the KAS Simulator for HTTP communication, the HTTP URL should also include the port number to access the web server.
  - Example: In Figure 13-3, the web server is listening to port 52526.
    To connect to the Simulator from the web browser, use the URL: http://127.0.0.1:52526.

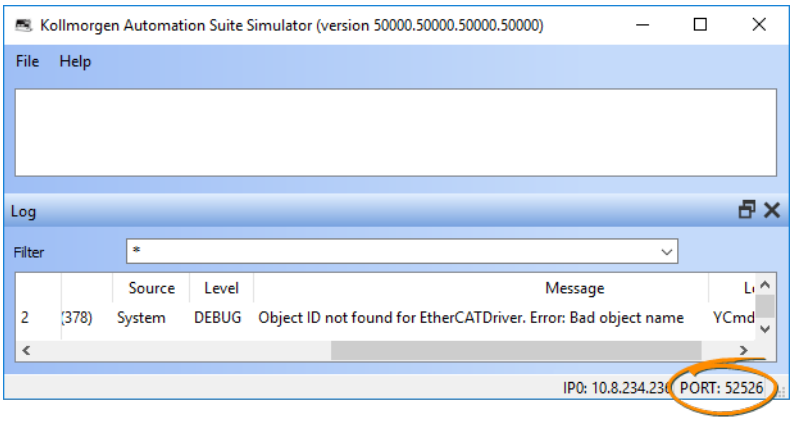


Figure 13-3: Example: KAS Simulator Port

- Depending on the applications running on the host computer, the alternate port may vary every time the simulator is started.

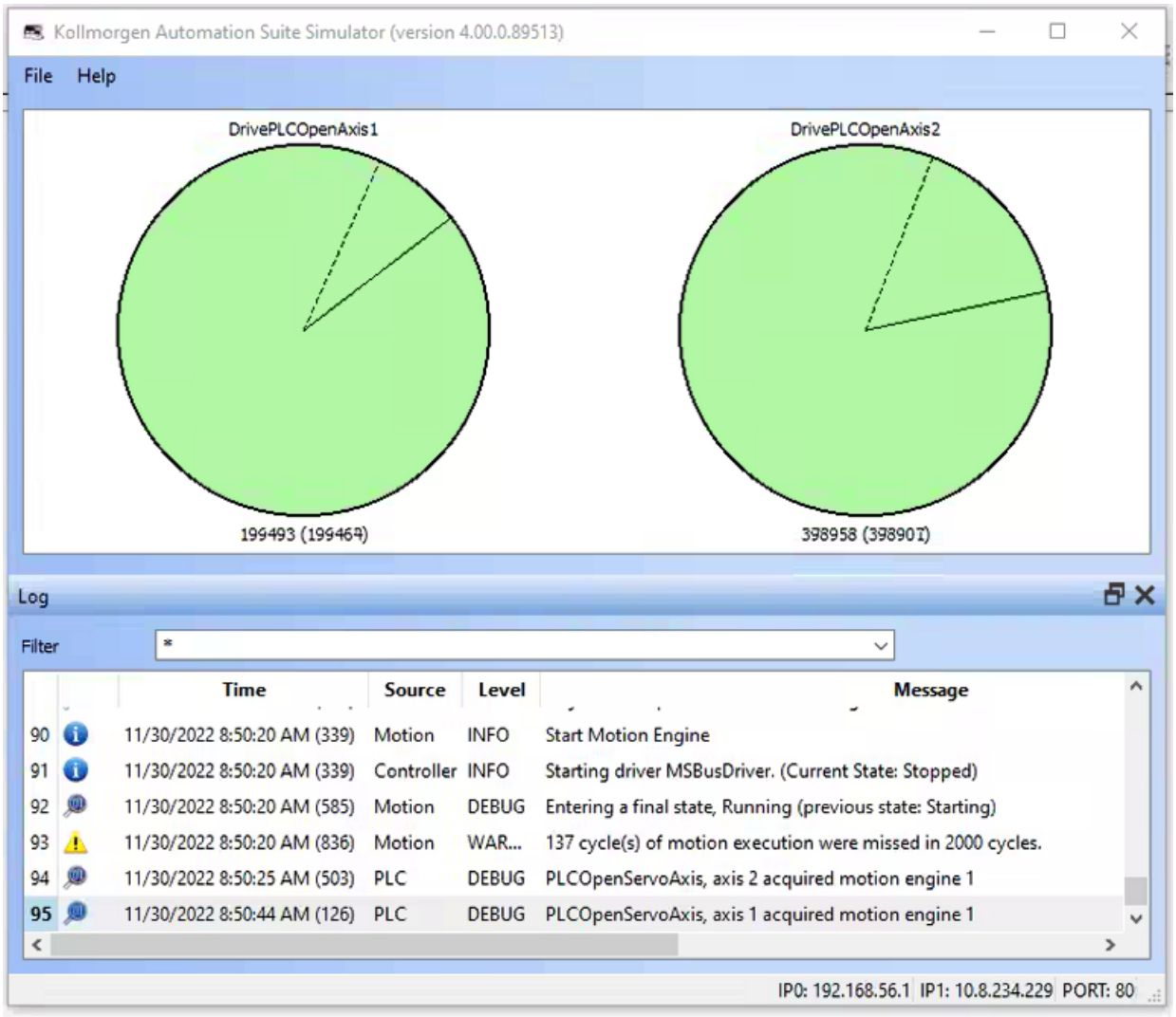
TIP

Only one instance of Simulator may be used at a time, but multiple instances of the KAS-IDE may be used with the Simulator.
A simple rule to remember is whatever instance is connected to Simulator, owns Simulator.

10.2.1 KAS Runtime Log window

The KAS Runtime Log window provides a running display of activity related to the execution of the application. Items displayed include application startup and initialization information.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



NOTE

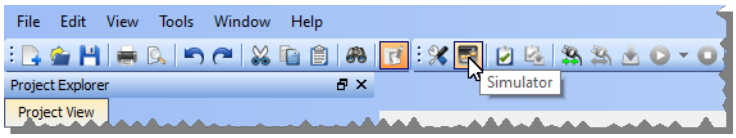
Some of the steps performed during the initialization process can be specified in the **initscript.bin** file located under:
C:\Program Files (x86)\Kollmorgen\Kollmorgen Automation Suite\Sinope Simulator\Resources

See also "Log window" (→ p. 586).

10.3 Stop the KAS Simulator

The **KAS Simulator** can be closed in several ways:

- From KAS-IDE by clicking the KAS Simulatorbutton in the toolbar.
- Selecting File > Exit.
- Click the Close button at the top right of the KAS Simulator.



Click the KAS Simulator button to perform these actions:

1. Stop the project if it is running.
2. Disconnect the KAS-IDE from the KAS Simulator.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

3. Close the KAS Simulator.
4. Change the project back to using a PxMM as the controller.

Selecting File > Exit or the Close button stops the project and disconnect the KAS-IDE from the KAS Simulator but the project does not reset the controller.

TIP

After the project is debugged using KAS Simulator, it can be downloaded to the real controller in production. This operation can be done simply by modifying the IP address of the device and then "Download the Project" (→ p. 265).

TIP

Only one instance of Simulator may be used at a time, but multiple instances of the KAS-IDE may be used with the Simulator.
A simple rule to remember is whatever instance is connected to Simulator, owns Simulator.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

11 Using the Controllers



Tasks related to the Kollmorgen controllers are:

- Configure parameters.
- Start and stop the KAS application.
- Update the firmware.
- Reset to factory settings.
- Review of "Controller Error and Alarm Descriptions" (➔ p. 577).

Rebooting or resetting the controller and recovering the firmware can be performed from the device or using the Web server.

11.1 Booting the AKD PDMM or PCMM Controller

This topic explains the boot sequence for the controller, which is based on the RAM and the Flash memory.

The flash memory contains two images:

- Recovery image (4 Mb) contains QNX operating system and the KAS web server.
- Regular image (9 Mb) contains QNX operating system, the KAS web server, and the KAS Runtime.

See Also

- "Boot from Recovery Image (B2 Button)" (➔ p. 602)
- "Controller Boot Startup Script" (➔ p. 598)

11.2 Booting the PCMM2G Controller

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This topic explains the boot sequence for the PCMM2G controller.

The PCMM2G controller contains these partitions: **Active**, **Boot**, **Data**, and **Inactive**.

- "Active Partition" (→ p. 594)
- "Boot Partition" (→ p. 594)
- "Data Partition" (→ p. 594)
- "Inactive Partition" (→ p. 594)

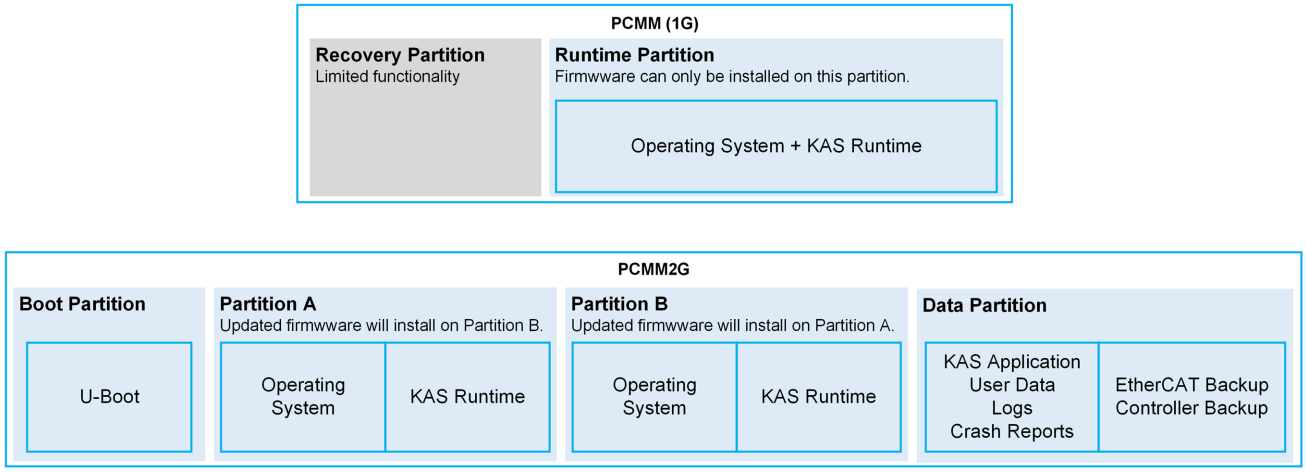


Figure 14-1: PCMM2G Partitions

11.2.1 Active Partition

- The **Active** partition:
  - Contains the entire operating system and the KAS Runtime.
  - Is where the controller is currently booted from.
    - It contains the current version of the operating system and KAS Runtime.
    - When the user downloads a new version of firmware, it is loaded onto the **Inactive** partition.
    - When the loading is complete, the controller reboots to the **Inactive** partition, making it the new **Active** partition.
- If the **Active** partition cannot boot up to an operable state, the controller can boot up a USB flash drive containing a recovery image.

11.2.2 Boot Partition

- The **Boot** partition contains the U-Boot, a universal bootloader.
  - The bootloader initializes the hardware and loads the operating system.

11.2.3 Data Partition

- The **Data** partition contains the:
  - KAS application, user data, logs, and crash reports.
  - EtherCAT and controller backup files.

11.2.4 Inactive Partition

- The **Inactive** partition:
  - Contains the entire operating system and the KAS Runtime.
  - Serves as a fallback.
    - If booting up the **Active** partition fails, the controller attempts to boot up from the **Inactive** partition.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- If the **Inactive** partition cannot boot up to an operable state, the controller can boot up from a USB flash drive containing a recovery image.
- The user can force booting up from the inactive partition by holding the B3 button in while powering on the controller.
    - This is one recourse to use if the controller does not become operational when booting off from the active partition.

See "Boot from Inactive Partition Image - PCMM2G" (➔ p. 595) for more information.

See Also

- "Boot from Recovery Image (B2 Button)" (➔ p. 602)
- "Controller Boot Startup Script" (➔ p. 598)
- "Boot Sequence" (➔ p. 596)

11.3 Boot from Inactive Partition Image - PCMM2G

The PCMM2G controller contains two versions of the firmware stored in the non-volatile storage (flash memory).

11.3.0.1 Operational Image

This is the firmware version that was last downloaded is stored in the primary partition.

11.3.0.2 Fallback Image

This is the firmware version downloaded prior to that is stored in the inactive partition.

NOTE

See "Booting the PCMM2G Controller" (➔ p. 593) about the partitions in the PCMM2G non-volatile storage.

11.3.1 Boot Failure

In rare situations, the controller might be unable to boot into operational firmware.

Examples: A failure could be caused by:

- Corruption of the flash memory containing the firmware image.
- A critical error early in the boot process.

Although Kollmorgen tries to ensure these failures never happen, the possibility requires a recovery mechanism to allow a working firmware image to be programmed.

11.3.2 Fallback Mode

Drastic failures of the operational firmware (i.e., those that cause the firmware to crash) trigger an automatic reboot into the fallback firmware image.

- An automatic reboot into the Fallback Mode does **not** happen when a critical error occurs early in the boot sequence.
    - This is a rare occurrence.
- The 7-segment display is stuck showing either 2 or 3.
    - It is possible for users to explicitly boot the controller into the Fallback Mode.

11.3.2.1 Boot into Fallback Mode

To explicitly boot into **Fallback Mode**, press and hold B3 button when powering up the controller.

- The goal of running a **Fallback** image is to recover the controller by downloading working operational firmware.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The 7-segment display on the controller reports an "E60" (→ p. 582) () error when the controller has booted up in **Fallback Mode**.

11.3.3 Limitations

There are some limitations to the system when the controller is running from the **Inactive** partition.

- The auto-start option for the KAS application is not supported.
  - The user must start the KAS application manually every time after the controller is powered up.
- An "E60" (→ p. 582) error is reported to indicate the active partition is corrupted.
  - This error is cleared when a new firmware is downloaded or restored using the previously taken controller backup.
  - See either the "Firmware tab" (→ p. 667) or "Restore tab" (→ p. 686).

Notes

IMPORTANT

It is strongly recommended to upgrade firmware again to have the same firmware version on the **Active** and **Inactive** partitions.

The KAS application remains compatible when the controller is operating with the same firmware version on the Active and Inactive partition.

- If the **Active** partition becomes corrupted, the controller runs from the **Inactive** partition.
  - If the firmware is **the same** on the **Active** and **Inactive** partitions, the KAS application runs correctly.
    - This avoids the need for the users to compile and download the KAS application to keep the machine running.
    - The user has time to download new firmware for resolving the corruption in the **Active** partition.
  - If the firmware is **not the same** on the **Active** and **Inactive** partitions, the KAS application does not work.
    - Failure Correction: On the **Inactive** partition, download and compile the firmware that is on the **Active** partition.

11.3.4 Recovery Mode

In the rare event the **Fallback** image stored in the flash memory does not work either, the controller can boot into Recovery Mode using the USB flash drive.

A binary image that boots the controller from the USB flash drive is provided with the firmware release.

See "PCMM2G Web server Recovery Mode" (→ p. 672) about how to recover the controller from the USB flash drive.

11.4 Boot Sequence

NOTE

The controller may be booted with or without an Ethernet cable attached.

These are the controller boot sequences:

- "PCMM2G Boot Sequence" (→ p. 597)
  - "Internal Flash" (→ p. 597)
  - "Recovery USB Drive" (→ p. 597)

When the controller is booted with a cable attached, the configured IP address (depending upon the current position of the rotary switch) appears in the 7-segment display.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- See "View the Controller IP Address" (→ p. 602) for more information.
- If the controller is started without a network connection then the IP address is not displayed.
- After the boot sequence is successful, the controller is in one of two modes:

	Normal operation.
	Recovery Mode (if firmware download is permitted).

11.4.1 PCMM2G Boot Sequence

These are the boot sequences of the PCMM2G:

- "Internal Flash" (→ p. 597)
- "Recovery USB Drive" (→ p. 597)

11.4.1.1 Internal Flash

NOTE

- If the controller is stuck in Stages 1 to 4, then recovery could be made by using the B3 button to boot off of the inactive partition.
- See "Boot from Inactive Partition Image - PCMM2G" (→ p. 595).

State	Display	Description
Hardware Power ON		PCMM2G has power.
Stage 1		Bootloader completed. Starting U-Boot.
Stage 2		U-Boot succeeded. Starting Linux Kernel.
Stage 3		Linux Kernel is up. PCMM2G processes Initializing.
Stage 4		Starting: • KAS Runtime process monitor. • KAS Runtime.
Final Stage		Runtime started.

11.4.1.2 Recovery USB Drive

See "PCMM2G Web server Recovery Mode" (→ p. 672).

NOTE

- If the controller is stuck in Stages 1 to 4, then recovery could be made by using the B2 button to boot off of the recovery USB drive.
- See "Boot from Recovery Image (B2 Button)" (→ p. 602).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

State	Display	Description
Hardware Power ON		PCMM2G has power.
Stage 1		Starting U-Boot from recovery USB drive.
Stage 2		U-Boot succeeded. Starting Linux Kernel.
Stage 3		Linux Kernel is up. PCMM2G processes Initializing.
Stage 4		Starting: • KAS Runtime process monitor. • KAS Runtime.
Final Stage		Recovery runtime started.

11.5 Boot from Recovery Image (B2 Button)

NOTE

- For PCMM2G:
  - Insert the Recovery USB flash drive into X37 USB Port of the PCMM2G before holding the B2 button during boot sequence.
  - See "PCMM2G Web server Recovery Mode" (→ p. 672).

IMPORTANT

- The controller displays a backward C on the 7-segment display if the wrong firmware is installed.
- You must enter recovery mode to load the correct firmware file.

See Also

- "Booting the PCMM2G Controller" (→ p. 593)

11.6 Controller Boot Startup Script

- After all the previous steps, the startup script starts automatically.
  - The script puts the 7-segment display into Stage 4.
- Before the controller boots up, this flowchart applies to their respective controller.

11.6.1 PCMM2G Flow

See "Booting the PCMM2G Controller" (→ p. 593) for more information.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

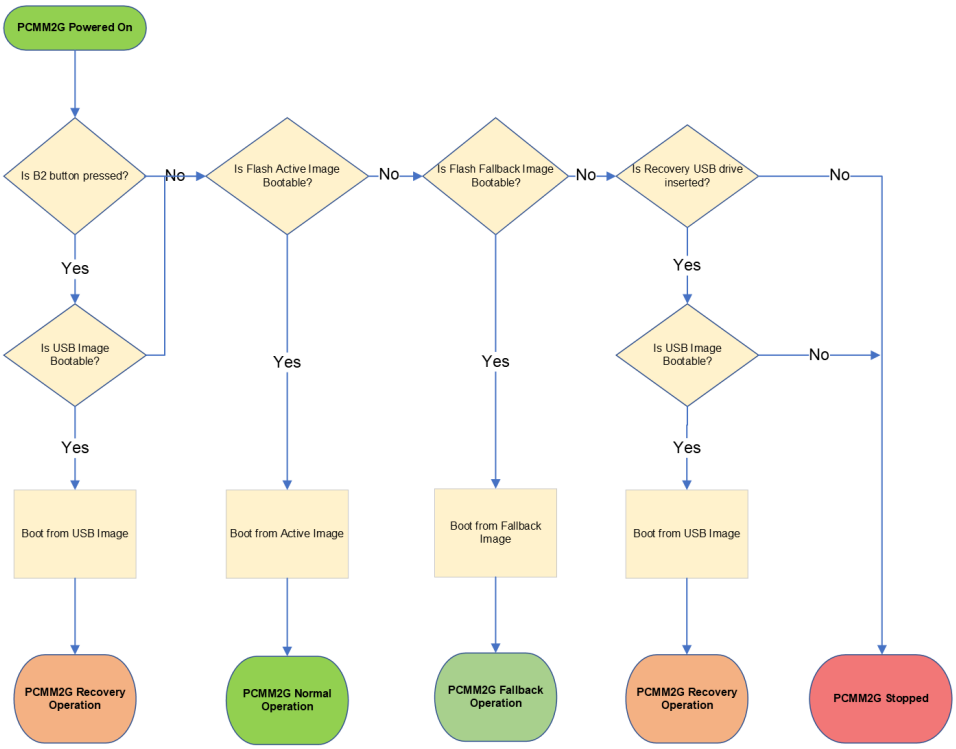


Figure 14-2: PCMM2G Flow

11.7 Use the Controller Buttons

In some cases, using the buttons on the controller is preferable to using the Web server.



On the front of the controller there are two buttons: B2 and B3.

- These buttons are used to:
  - Enter Recovery Mode.
    - See "Boot from Recovery Image (B2 Button)" (→ p. 602).
  - "View the Controller IP Address" (→ p. 602).
  - Stop and start the application.
  - Reset the controller to factory settings.
    - See "Reset the Controller to Factory Settings" (→ p. 603).
  - PCMM2G only: "Boot from Inactive Partition Image - PCMM2G" (→ p. 595).

11.7.1 Button Functionality

11.7.1.1 Power-up

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Press and Hold	Result
B2	"Boot from Recovery Image (B2 Button)" (→ p. 602)
B3	PCMM2G only: "Fallback Mode" (→ p. 595)

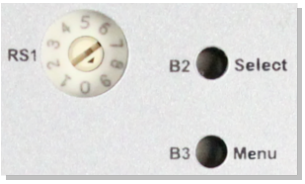
NOTE

- Press the B2 or B3 button before the Hardware Power ON stage ( ) begins.
- Hold at least until the controller reaches the Stage 2 during the "Boot Sequence" (→ p. 596).

11.7.1.2 Running

Press	Result
B2	Menu option selector
B3	"B3 Menu Button" (→ p. 601)

11.7.2 B3 Menu Button



The B3 menu push-button cycles through a list of menu items displayed on the 7-segment display.

11.7.2.0.1 Access Functions Menu

1. Press and hold the B3 button during the boot sequence.
   - Each B3 press advances to the next menu item.
   - The menu item is displayed for 10 seconds.
     If no button is pressed within the 10 seconds, the 7-segment display returns to Normal operation.
2. Press the B2 button to select the current menu item.

11.7.2.1 Application is Running

Functionality	Display	Notes
Display the IP		See "View the Controller IP Address" (→ p. 602) for more information.
Stop the application		This stops the KAS Runtime.

11.7.2.2 Application is Not Running

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Functionality	Display	Notes
Display the IP		See "View the Controller IP Address" (→ p. 602) for more information.
Start the application		This starts the KAS Runtime.
Factory **Reset**		See "Reset the Controller to Factory Settings" (→ p. 603).
Backup firmware to SD card		See "B3 Button Backup" (→ p. 605). Does not apply to PCMM2G.
Restore firmware from SD card		See "B3 Button Restore" (→ p. 606). Does not apply to PCMM2G.

NOTE

- When selected, the **Start**, **Stop**, **Backup**, **Restore**, and **Reset** functions do not initiate immediately; they require confirmation.
- The 7-segment display flashes a y, prompting for confirmation.
- Press B2 to confirm the function and begin the process.
- If the function is not confirmed within 10 seconds, the action is canceled.



11.7.3 View the Controller IP Address

The IP address assigned to the controller is shown in the 7-segment display.

- The IP address is shown at boot and can be accessed from the "B3 Menu Button" (→ p. 601).
- The default IP address depends on the Rotary Switch RS1 position.

Example



Figure 14-3: IP sequence of the 7-segment display

NOTE

- On the PCMM2G, if a network cable is not attached or no IP address is assigned, the controller shows an IP address of 0.0.0.0.

11.7.4 Boot from Recovery Image (B2 Button)

NOTE

- For PCMM2G:
  - Insert the Recovery USB flash drive into X37 USB Port of the PCMM2G before holding the B2 button during boot sequence.
  - See "PCMM2G Web server Recovery Mode" (→ p. 672).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

IMPORTANT

- The controller displays a backward C on the 7-segment display if the wrong firmware is installed.
- You must enter recovery mode to load the correct firmware file.

See Also

- "Booting the PCMM2G Controller" (→ p. 593)

11.7.5 Reset the Controller to Factory Settings

WARNING

Performing a factory reset of the controller should be considered very carefully.
The reset erases everything: User data files, the application, the controller IP address, any retain variables, passwords, shared directory settings, and all log files.

The reset is performed using these methods:

- From the "File System tab" (→ p. 666) of the Web server while the drive is running.
  - This method is recommended due to its ease of use.
- Selecting the function from the "B3 Menu Button" (→ p. 601).
  - This can be done during the boot sequence or while the drive is running.

After two seconds (or longer if pressed during power-up), the 7-segment display on the controller is changed to an animation pattern indicating the factory reset has started.



After the factory reset is complete, the controller is powered down and restarted automatically.

11.7.5.1 Factory Reset Changes

The changes which occur during a reset are:

Erased Items

- User data files.
- Any application previously downloaded.
- Retained variables.
- Shared directory settings.
- Log files.

Defaults Restored

- Password for both the web page and SSH login.
  - See "SSH Login to a Controller" (→ p. 617) and "User Authentication" (→ p. 623).
- Controller network configuration.
  - Examples: IP address, Subnet, and Gateway settings.
- Auto-Start option.

Security

- The SSH keys are regenerated.
  - SSH clients logged on during the factory reset are forced to log in again.
  - All Date and Time settings are reset to default (NTP enabled).

Notes

- The factory reset cannot be performed while an application is running.
- Do not turn off the controller during the reset procedure.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Estimated time for factory reset:
  - PCMM2G: About a minute.
  - The 7-segment display on the controller is animated during this process.
- After the factory reset is complete, the controller is powered down and restarted automatically.
- After the controller is restarted, the controller IP address may change based on the controls rotary switch.
  - If the rotary switch is at position 0 (zero), the same IP address as before should be assigned to the controller.
  - If the rotary switch is set to 1-9, a pre-configured IP address is defined.
    - This IP address must be taken into account when trying to reconnect to the webpage using a web browser.

11.7.5.2 Reset Controller While the Drive is Running

A reset may be done any time after the controller is powered on and an application is not running.

The reset is ignored if an application is running on the controller .

	Normal Operation	Press the B2 button to access the menu. Use the B3 button scroll to the Reset option in the menu. Press B2 to confirm the reset to factory defaults.
	Program Running	Reset to factory defaults is not permitted. The Reset menu item is not available.

11.7.6 Controller Backup and Restore

The Backup and Restore functions are used to replicate a Kollmorgen controller.

The controller data is replicated on the PCMM2G USB flash drive.

11.7.6.1 Backup

Backup is performed with one of these methods:

- Use the web server.
  - This is recommended because it has functionality the button method does not offer (e.g., back up the drive data).
    - See "Backup and Restore from the Web server" (➔ p. 682).
- Use the "B3 Menu Button" (➔ p. 601) on the front of the controller.
  - See "B3 Button Backup" (➔ p. 605).
  - PCMM2G: Currently does not support controller backup using the B3 menu button.

TIP

For a complete backup of both the controller and drive, use the Web Server backup.

11.7.6.2 Restore

Restoring information to the controller is performed with one of these methods.

- Use the web server.
  - See "Backup and Restore from the Web server" (➔ p. 682) for more information.

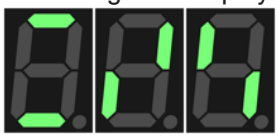
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Use the B3 button menu on the front of the controller.
  - See "B3 Button Restore" (→ p. 606) for more information.
  - PCMM2G: Currently does not support controller restore using the B3 menu button.

Backup & Restore What?	PCMM2G To/From USB flash drive via B2/B3 buttons	PCMM2G To/From USB flash drive via Web server
Controller: Firmware	✗	✓
Controller: Network Configuration	✗	✓
Controller: PLC Application	✗	✓
Controller: Retain Variables	✗	✓
EtherCAT Devices: Network Topology	✗	✓
EtherCAT Devices: AKD / AKD2G Drive Firmware	✗	✓
EtherCAT Devices: AKD / AKD2G Drive Parameters	✗	✓

‡ Restore only

11.7.6.3 Data Transfer

- The 7-segment display shows chasing light animation while the backup or restore is occurring.
   (repeats)
- The Backup and Restore functions have an all-or-nothing behavior.
  - PCMM2G: If there is no USB flash drive inserted, not enough space on the USB, or if files are missing, nothing is copied and the 7-segment display shows an error code.
- If files already exist on the USB flash drive (in the backup directory), they are deleted and replaced with the new backup configuration files.
  - The files on the controller are replaced with the USB flash drive files.

⚠WARNING

Do not modify the files on the USB flash drive.
This could result in the restore function failing.

TIP

If there are multiple backup configurations, you must use one USB flash drive per backup configuration.

11.7.6.4 B3 Button Backup

NOTE

PCMM2G: Currently does not support controller backup using the B3 menu button.

One of the functions accessible from the B3 button is backing up the controller.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The Backup function stores a copy of the data on the PCMM2G's USB flash drive.
- This function is shown in the 7-segment display as **bACUP**.



- Pressing B2 selects the function.
  - This function does not initiate automatically.
  - B2 must be pressed again to confirm the process.

11.7.6.5 B3 Button Restore

NOTE

PCMM2G: Currently does not support controller restore using the B3 menu button.

One of the functions accessible from the B3 button is restoring the controller and AKD PDMM's drive.

- The Restore function restores and load files onto the controller from an SD card.
- This function is shown in the 7-segment display as **rEStorE**.



- Pressing B2 selects the function.
  - This function does not initiate automatically.
  - B2 must be pressed again to confirm the process.
- The 7-segment display flashes a **y**, prompting for confirmation.
  - Press B2 again confirms the function.
    The data transfer begins.
  - If the function is not confirmed within 10 seconds the action is canceled.

The data restored from the SD card includes:

- AKD PDMM or PCMM configurations (auto-start and IP address).
- Network configuration.
- PLC Application (including ECAT XML configuration, cam tables, etc.)
- "Retain Variables" (➔ p. 292).
- Shared Directory settings.
- "User Data tab" (➔ p. 631).
- Web server and SSH user accounts configuration.

11.7.7 Backup and Restore tab

The controller can backup and restore EtherCAT devices on an EtherCAT network.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

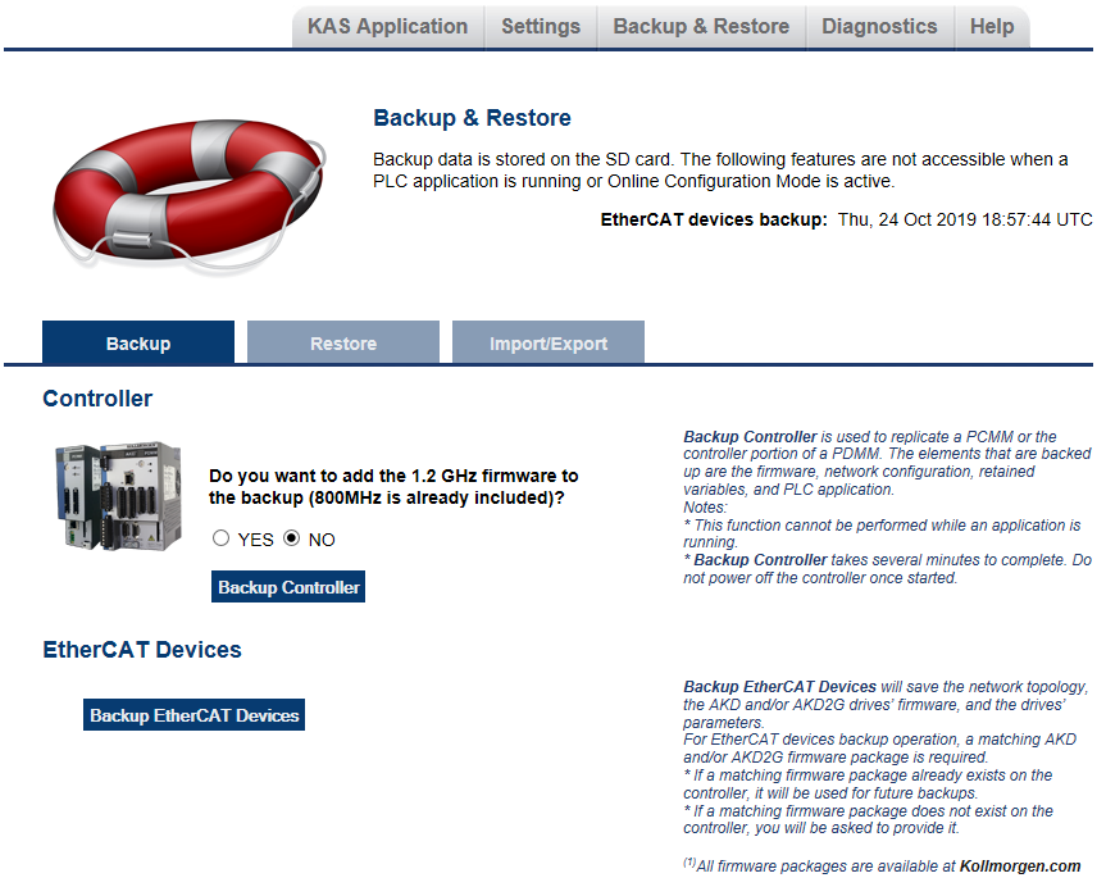


Figure 14-4: Backup and Restore tab

This feature:

- Is useful as a maintenance operation to replace any controller drives in an operational machine.
- Reduces the manual steps for saving/loading each drive's firmware and parameters into a few automated steps.

The Backup/Restore functionality is located in the controller's Web server and is accessible from a web browser.

See Also

- "Using the KAS Web server" (→ p. 619)
- "Backup tab" (→ p. 683)
- "Restore tab" (→ p. 686)

See the "Controller Backup and Restore" (→ p. 604) in the KAS-IDE help.

11.7.7.1 EtherCAT Devices Backup

The Backup operation discovers all devices on the EtherCAT network and stores the topology information, drive firmware files, and parameters.

Controller	Local Storage
PCMM2G	Controller flash memory.

11.7.7.1.1 Required Items

These items are required before starting a backup of the EtherCAT devices:

- The Kollmorgen AKD Series Servo Drives firmware files package (you are prompted with the expected file).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The package is:
  - Included in the KAS software installation directory
    - Example: `\Program Files (x86)\Kollmorgen\Kollmorgen Automation Suite {{kas.version}}\Astrolabe\DrivesFW`.
    - Example: `C:\Users\{user.name}\Local\Kollmorgen\Kollmorgen Automation Suite {{kas.version}}\Astrolabe\DrivesFw`.
  - Available for download from Kollmorgen's AKD Series Servo Drives website.
- PCMM2G requires a USB flash drive connected to the USB port.

11.7.7.1.2 Device Connection and Configuration

All EtherCAT network devices must be:

- Connected to the controller.
- Configured as necessary for machine operation.
- A PLC application downloaded to the controller, containing the EtherCAT device map.

TIP

Verify the devices are in the order as expected.
The topology information is stored in the backup and is used to check for identical topology during a restore function.
The device order is not important.**Example**: An AKD can be anywhere in the EtherCAT topology but, for restore to occur, the devices must be in the same order as when backed up.

NOTE

The PLC application cannot be running and the IDE must not be in "Online Configuration Mode" (→ p. 471). Stop the PLC application or disable Online Configuration Mode before a Backup or Restore.

11.7.7.1.3 Modular EtherCAT

This procedure should be followed to backup EtherCAT devices when the modular EtherCAT feature is used to remove devices to create a machine variant.

See "Modular EtherCAT Concept" (→ p. 1000).

1. Run the KAS application which sets up the correct EtherCAT topology for the machine variant.
2. Stop the KAS application.

IMPORTANT

Do not power cycle or reboot the controller between Steps 2 and 3.

3. Backup the EtherCAT devices.
   See "Backup Procedure" (→ p. 608).

11.7.7.1.4 Backup Procedure

1. On the Web server home page, click the **Backup & Restore** tab.
2. On the **Backup** tab, click the Backup EtherCAT Devices button.

   The web server will ask you to choose one or more firmware package files and suggest the filename that matches the firmware version on your drives.

   - Selecting the firmware package files is a one-time event.
   - The controller remembers the selection for future backups. (Figure 14-5)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



Figure 14-5: Web server Backup

NOTE

The AKD series drives have a firmware package file specific to their firmware version.
AKD2G has a different firmware package file specific to its firmware version.

3. Click the Continue Backup EtherCAT Devices button. (Figure 14-6)
   It will take a couple of minutes or longer, depending on the number of Kollmorgen AKD Series Servo Drives present in the system.

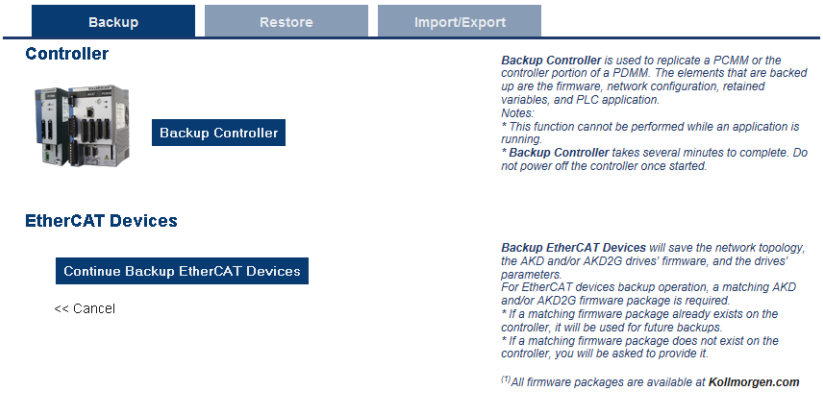


Figure 14-6: Web server Backup Continue

When the backup is complete, the Web server indicates whether the backup was successful.

TIP

After the backup is complete, it is a good idea to export the backup to an off-site location for safe keeping.
See "Controller Import / Export Backup from Web server" (➔ p. 612).

11.7.7.2 EtherCAT Devices Restore

The Restore operation discovers the devices on the EtherCAT network and compares the physical topology information to the topology information stored in the backup.

- A Restore detects the replaced EtherCAT devices and restores them.
- Advanced users can manually select the specific EtherCAT devices and restore them.
- The backup files containing the topology information, drive firmware file, and drive parameters are retrieved from the controller's local storage.

Controller	Local Storage
PCMM2G	USB flash drive

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

IMPORTANT

The application starts immediately after the Restore operation is complete if the Autostart option is enabled on the controller.
Be sure to Restore all of the replaced devices.
Verify the EtherCAT device order is correct.
Disable the Autostart option before commanding the Restore operation to check the devices before starting the application.

NOTE

The EtherCAT Devices Backup and Restore feature may be used in EtherCAT networks with third-party devices but only the Kollmorgen controller can be backed up and restored.
Specific configurations applied to third-party devices with non-Kollmorgen tools must be reapplied when the third-party device is replaced.

Procedure

1. On the Web server home page, click the **Backup & Restore** tab.
2. Click the **Restore** tab.
3. Choose to either:
   - Restore the replaced EtherCAT controller devices.
     - Click the **Restore EtherCAT Devices** button.
   - Manually select the devices with the Advanced view.
     a. Click the **Advanced** link.
     b. Select the devices to restore.
     The controller identifies the replaced drives and pre-selects them.
     c. Click **Restore selected devices**. (Figure 14-7)

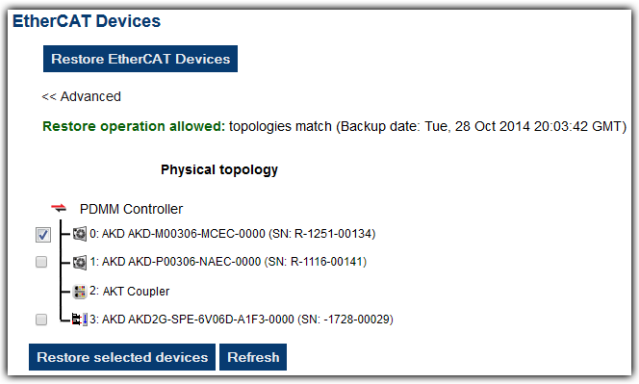


Figure 14-7: Restore selected devices

When the restore is complete, the Web server indicates whether it was successful.

11.7.7.3 AKD / AKD2G / Kollmorgen Essentials Drive Backup and Restore Compatibility

The replacement AKD / AKD2G / Kollmorgen Essentials Drive must have the same model number as the device that was originally stored in the backup.

- All AKD drives **must have** the same firmware version for the Backup and Restore to work.
- The Restore operation compares the model numbers between the backup and the replacement AKD.
  - Restore is not allowed if they are not compatible.

TIP

The **AKD-Pxxxxx-NB** and **AKD-Pxxxxx-NC** models are functionally compatible with, and may be replaced with, **AKD-Pxxxxx-ND** devices provided the backup contains AKD firmware version v01-22-00-000 and later.

Restore

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

		AKD-NB	AKD-NC	AKD-ND	AKD2G	Kollmorgen Essentials Drive
Backup	AKD-NB	✓	✓	✓	✗	✗
	AKD-NC	✓	✓	✓	✗	✗
	AKD-ND	✗	✗	✓	✗	✗
	AKD2G	✗	✗	✗	✓	✗
	Kollmorgen Essentials Drive	✗	✗	✗	✗	✓

Model Number Verification

See the sticker on the side of the drive to check the model number on your AKD / AKD2G / Kollmorgen Essentials Drive.

Compatibility

To check compatibility between your replacement drive and the backup:

1. Replace the AKD / AKD2G / Kollmorgen Essentials Drive on the EtherCAT network.
2. Press the Scan network button.
    - The Web server displays the physical topology.
      Use this to restore the selected drives if the backup and replacement drives are compatible. (Figure 14-8)

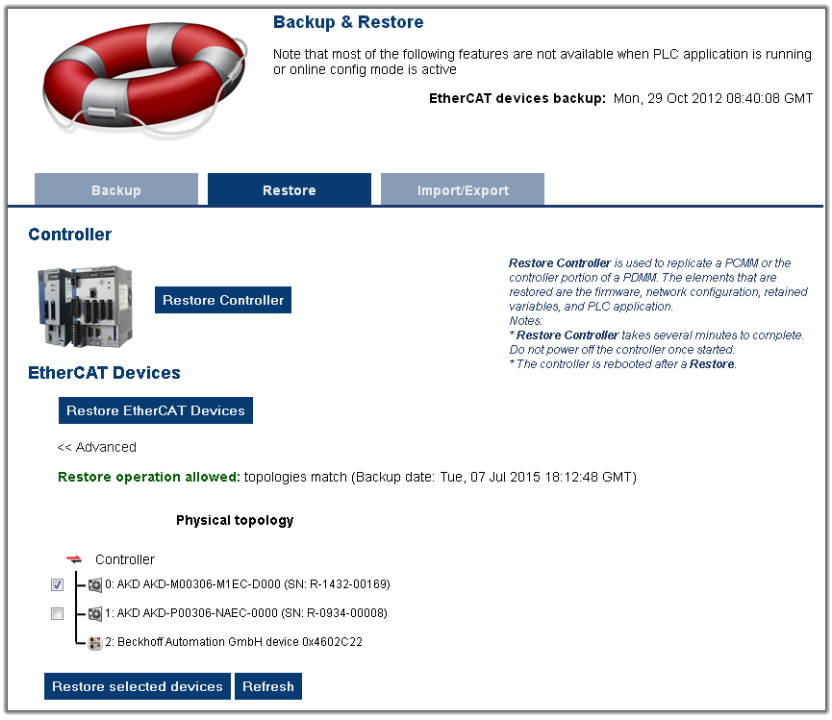


Figure 14-8: Web server Topology - Compatible

- The Web server displays the backup and physical topology and indicates the non-compatible drives if the backup and replacement drive are not compatible. (Figure 14-9)

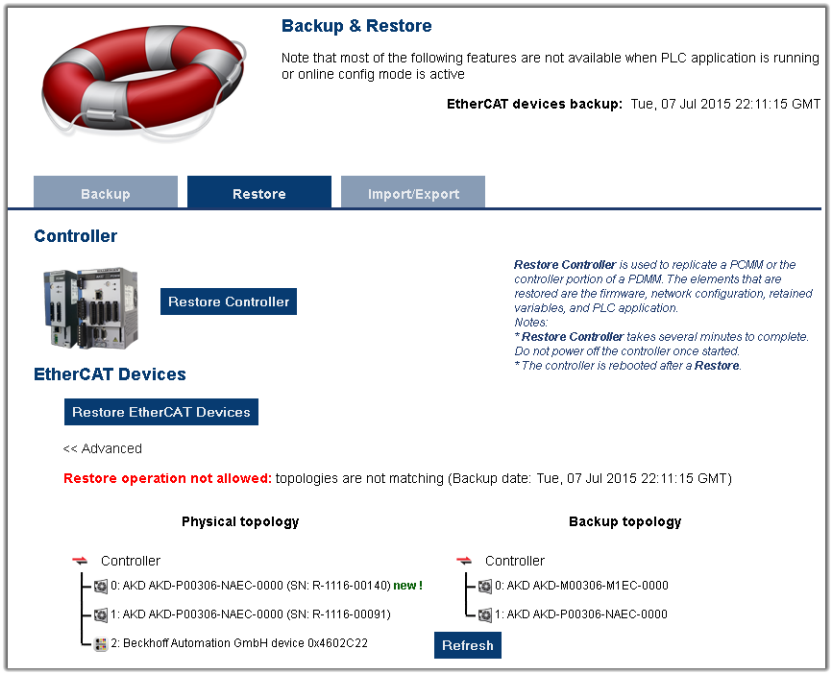


Figure 14-9: Web server Topology - Not Compatible

TIP

The serial number (SN) is shown in the Physical topology Web server view and on the sticker affixed to the AKD / AKD2G / Kollmorgen Essentials Drive.
Use the serial number to match the actual hardware with its representation on the Web server.

11.7.7.4 Controller Import / Export Backup from Web server

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

11.7.7.4.1 Controller Backup Import / Export

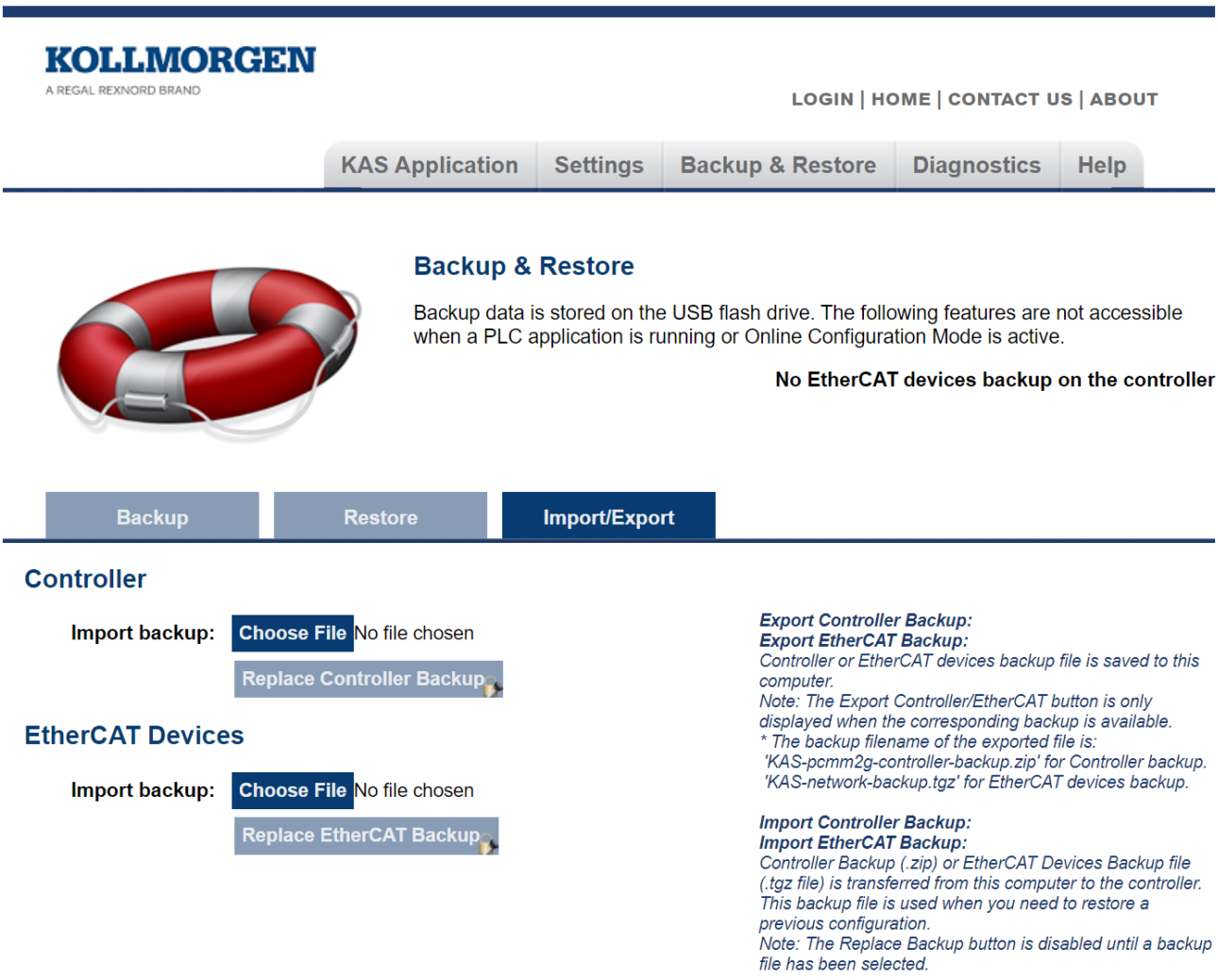
Controller backup data can be saved to or written from a removable storage devices USB flash drive).

- This functionality is available when a controller backup file exists on the device's USB flash drive.
  - See the "Backup tab" (→ p. 683) for more information.
- During the import process, this function replaces an existing backup file without further prompting.
  - If the process is interrupted or fails, it does **not** overwrite the backup file.

11.7.7.4.2 EtherCAT Devices Backup Import / Export

A network backup may be exported and imported.

- The export procedure saves a backup file to the computer running the web browser.
- The import procedure transfers a backup file onto the controller to be used later for restoring a previous configuration.



11.7.7.4.3 Export Procedure

NOTE

The Export button is only visible if a controller backup is available.

1. Click the **Export EtherCAT Backup** button.
   The browser starts transferring a backup file.
   Depending upon the browser being used, this may involve a prompt confirming receiving the file.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

2. You may move the file to a different directory once the file is saved.
   The file may be renamed to help identify the backup file with the machine.

11.7.7.4.4 Import Procedure

1. Click the **Choose File** button and select the backup file to import.
   This backup file is used to replace the current backup on the controller.
2. Click the **Replace EtherCAT Backup** button.
   This creates a backup on the controller with the data stored in the specified backup file.
   Any previously existing backup is replaced.
   If the import fails, the previous backup is not replaced.

The Replace Backup button is disabled until a backup file has been selected.

On some browsers, the **Browse** button may be labeled **Choose File**.

11.7.7.5 EtherCAT Devices Backup / Restore Limitations

- Only Kollmorgen AKD Series Servo Drives are supported for backup/restore.
  - Kollmorgen S300 and S700 drives are not supported by backup/restore.
- Only a single backup is supported in the controller at one time.

NOTE

To keep multiple backup configurations, perform the backup and then export it to a local PC.
When ready to restore, import the specific backup file to the controller and then perform the restore.

IMPORTANT

An import overwrites any backup existing in the controller's removable storage device.

11.7.7.5.1 Backup Conditions

These conditions must be met before starting an EtherCAT device backup:

1. The KAS application should be run and then stopped.
2. Do not power cycle or reboot the controller after stopping the KAS application.

The "EtherCAT Devices Backup" (→ p. 607) and "EtherCAT Devices Restore" (→ p. 609) functions are not permitted:

- While a PLC application is running.
- When the KAS-IDE is in "Online Configuration Mode" (→ p. 471).
  - Stop the PLC application or disable Online Configuration Mode before a Backup or Restore.

All drives on the EtherCAT network must have the same firmware version.

- The AKD firmware version must be 01-06-00-003 or higher.
- Kollmorgen AKD Series Servo Drives firmware packages are available for all production releases.

11.7.7.6 EtherCAT Devices Backup / Restore Troubleshooting

The web server displays an error message if there is a failure of an EtherCAT device backup or restore.

- The message describes the cause of the failure and a possible remedy.
- Note any error messages; they are helpful with remedying the problem.
- These are some common error messages and remedies.
- The message box format may appear differently depending on the web browser, but the message content is the same.

During backup, the EtherCAT fails to reach pre-op mode:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

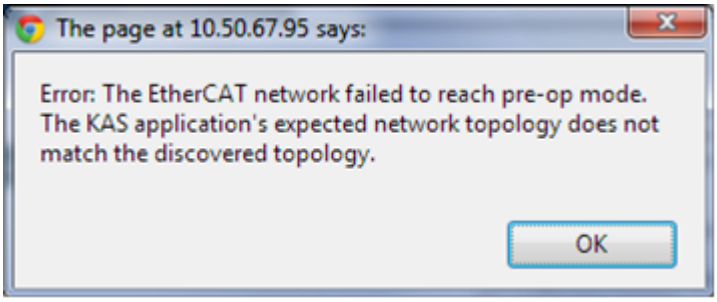


Figure 14-10: Error message: AKD Backup/Restore Fail

1. Using the KAS-IDE, open the EtherCAT view and scan the network.
2. Compare the nodes, their order, and types to the topology in your application.
3. After you identify the differences, either:
   - Modify the application's devices to match the physical network.
   - Correct the physical network by adding/moving/removing nodes.

During backup, at least one drive has an unsupported firmware error for backup:

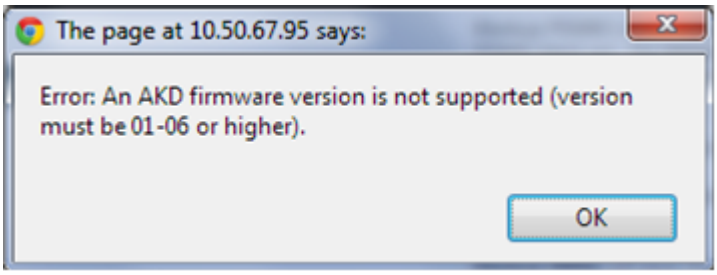


Figure 14-11: Error message: AKD Firmware not Supported

1. Using the KAS-IDE, open the EtherCAT view.
2. Upgrade all of the AKDs drive firmware.
   See AKD Firmware downloads.

During restore, the backup file is corrupt:

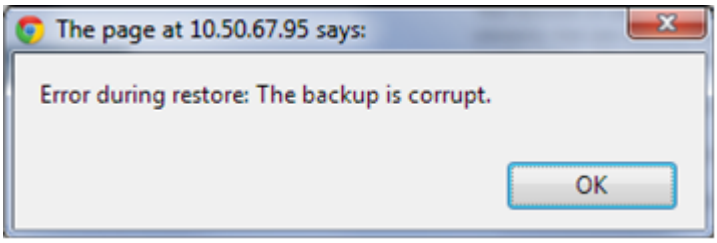


Figure 14-12: Error message: Backup is corrupt

Before restoring the AKD / AKD2G / Kollmorgen Essentials Drive, the backup file must be valid.

- To correct a bad backup file on a controller, import a known good backup file from another source (e.g., local hard drive, network drive, USB flash drive, etc.)
- If a valid backup file is not available, you must manually configure the replacement drives by downloading firmware and changing the drive's parameters using the drive views in the KAS-IDE.

Not able to restore Kollmorgen AKD Series Servo Drives successfully.

The error message describes at which step the restore failed (e.g., failing to restore parameters).

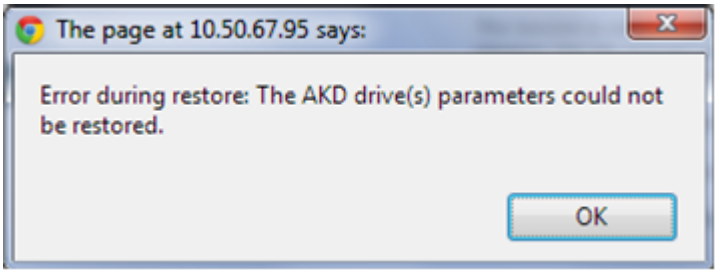


Figure 14-13: Error message: Restore failure

1. If an AKD / AKD2G / Kollmorgen Essentials Drive restore fails and you have already verified the controller has a valid backup and the network topology is correct, then retry the Restore.
2. If you still cannot restore successfully after two or three attempts, check the network cables and try a different replacement drive(s).
   This test isolates the problem to the specific drives or the controller problem.
3. If you still cannot restore a replacement AKD / AKD2G / Kollmorgen Essentials Drive, then you must manually configure the replacement AKD / AKD2G / Kollmorgen Essentials Drive by downloading firmware and modifying the drive's parameters using the drive views in the KAS-IDE.

11.8 Controller Onboard I/O

The procedure to define the local I/Os of the controller is similar to the one for I/O slices.

- The Onboard digital I/O update rate for:
  - PCMM2G matches the EtherCAT update rate.

EtherCAT Update Rate	Onboard Digital I/O Update Rate PCMM2G
500Hz	500Hz
1kHz	1kHz
2kHz	2kHz
4kHz	4kHz

See "Map Input and Output to Variables" (→ p. 526).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

11.9 Using SSH

The Kollmorgen controllers can be accessed using SSH.

- SSH is used to perform maintenance operations that are not available through the web interface.
- Kollmorgen Application Engineers may need to use SSH to access the controller to help resolve customer issues.

11.9.1 Login

- PCMM2G only allows access to the **admin** user.

Because the admin user is in the sudo group, the user can use root privileges to perform maintenance using the sudo command.

- Internally, the controller uses a **kas** user account.
  - It is recommended to not use this account because files in the **kas** user directory are critical to the proper functioning of the controller.

IMPORTANT

To ensure security, it is critical that the login password be changed from the default value.
See "Change Controller's Password using SSH" (→ p. 618) for more information.

See Also

- "SSH Login to a Controller" (→ p. 617)
- "Restore Controller Default Password" (→ p. 618)

11.9.2 SSH Login to a Controller

An SSH client is required to log in to a controller using SSH.

NOTE

The first time you log into the controller, you may be prompted (depending on the client) whether you trust the device (controller).
The admin/root password and the webserver password are separate. One does not affect the other.

11.9.2.1 Login to PCMM2G

1. Configure the SSH client using these settings:

Setting	Value
Host Name or IP Address	IP Address of the controller.
Connection Type	SSH
Port	22

2. At the login prompt, enter the user name and password.
   The admin password is unit-specific.
   It is assigned based on the serial number of the unit.
   - Default User Name: `admin`.
   - Default Password: pcmm2g + the last three digits of the controller serial number.
     - Example: If the controller serial number is KL2023180027, default password is pcmm2g027.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

The controller serial number can be found from the "Web server Home page" (➔ p. 621) or by checking the label printed on the unit.

11.9.2.1.1 List of Common SSH Clients

Kollmorgen does not recommend any particular program.

These are some common SSH clients:

- PuTTy
- TeraTerm

IMPORTANT

To ensure security, it is critical that the login password be changed from the default value.
See "Change Controller's Password using SSH" (➔ p. 618) for more information.

11.9.3 Change Controller's Password using SSH

This method is used to change the controller's root password using SSH.

1. Log on through SSH.
2. At the prompt, type **passwd** and press <Enter>.
3. If prompted, enter the current password.
4. In the next two prompts, enter the new password.

NOTE

- PCMM2G uses the admin account.

11.9.3.1 PCMM2G: Change Password

```
kas-pcmm2g:~$ passwd
Changing password for admin.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

11.9.4 Restore Controller Default Password

This procedure is used to restore the controller default password.

11.9.4.1 Restore to Factory Settings

WARNING

Performing a factory reset of the controller should be considered very carefully.
The reset erases everything: User data files, the application, the controller IP address, any retain variables, passwords, shared directory settings, and all log files.

This restore is applicable to all controllers.

See "Reset the Controller to Factory Settings" (➔ p. 603) for more information.

11.9.4.2 Restore PCMM2G

The PCMM2G password is reset to the factory default using the "Reset the Controller to Factory Settings" (➔ p. 603) procedure.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

IMPORTANT

To ensure security, it is critical that the login password be changed from the default value.
See "Change Controller's Password using SSH" (→ p. 618) for more information.

11.10 Using the KAS Web server

Kollmorgen Automation Suite comes with a Web server used to perform these operations:

- Read information about the controller (i.e., model, runtime or firmware version, KAS version).
- Interact with your application (Start and Stop your KAS application).
- View real and simulated axes.
- Connect to a shared directory.
- See all the log messages.
- Upgrade the controller firmware.
- Change the IP address.
- View system diagnostics including storage space, memory and CPU temperature.
- Reset the controller to factory settings.
- Backup/Restore the controller.
- Backup/Restore EtherCAT devices.

11.10.1 Access the Web server

The Web server is accessed either with:

- "Web Browser Access" (→ p. 619)
- "KAS-IDE Access" (→ p. 619)

11.10.1.1 Web Browser Access

1. Open a web browser.
2. Enter the controller's IP address for the URL.
   See "Find the Controller IP Address" (→ p. 620).

11.10.2 KAS-IDE Access

1. Open KAS-IDE.
2. Open an existing project or create a new project.
3. In the **Project Explorer**, right-click the controller to open the menu and click **Properties**.
   The **Controller Properties** dialog box opens. (Figure 15-1)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

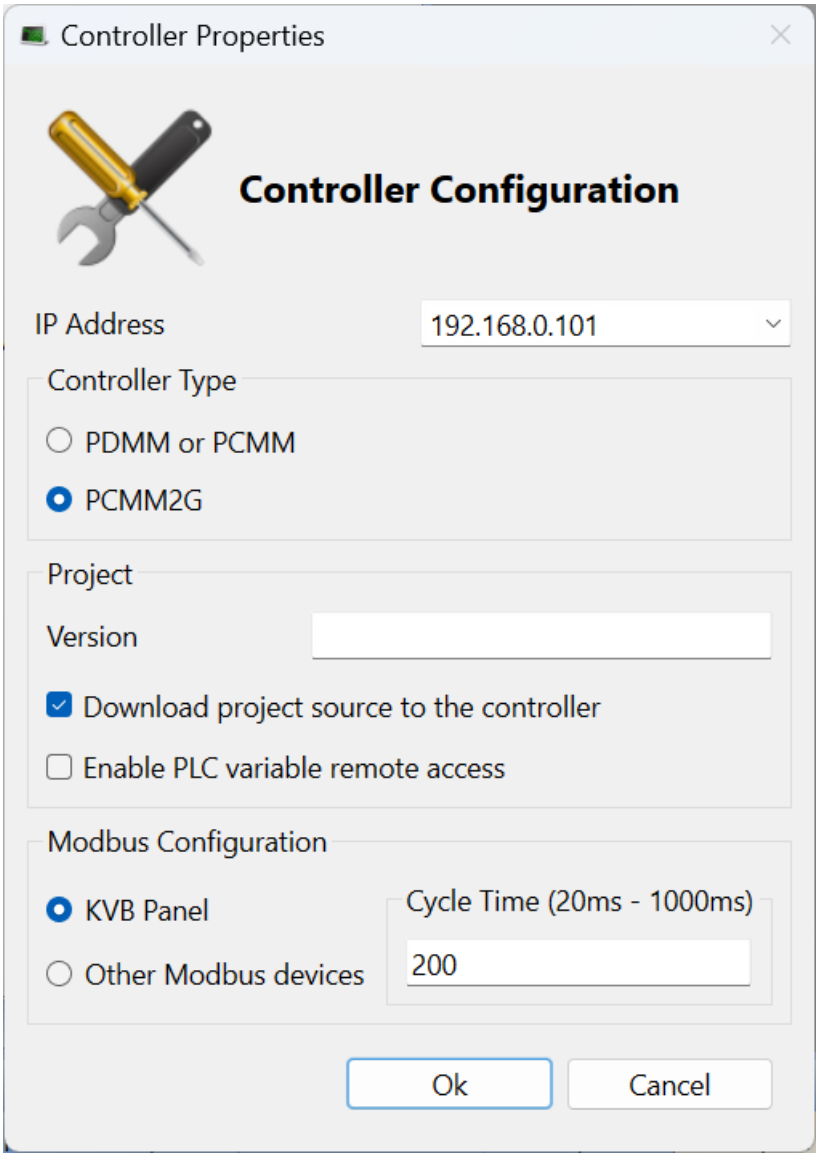


Figure 15-1: Controller Properties dialog box

4. In the IP Address text box, enter the controller's address or
   Click the **IP address** drop-down arrow and select the applicable IP address.
   See "Find the Controller IP Address" (→ p. 620).
5. Click **OK** to save the changes or selections and close the dialog box.
6. Double-click the controller node or right-click the controller node and click Access Webserver menu item.
   The Web server opens.

NOTE

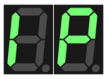
PCMM2G employs the HTTPS protocol to ensure a secure connection to its web server.
The secure link ensures that all sensitive transmitted data or entered information (e.g., their KAS web server and Shared Directory login passwords) is encrypted.

11.10.3 Find the Controller IP Address

Use this procedure if you do not know the IP address assigned to the controller.

1. Press B3 once.
   The 7-segment display flashes the letters I and P.
   Example:

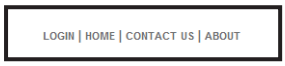
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



2. Press B2 to select the IP option.
   The 7-segment display shows the IP address.
   Example: 

11.10.3.1 Links

The controller Web server home page shows these links:



- Login
  - Opens the Kollmorgen controller login dialog.
- "Web server Home page" (→ p. 621)
- Contact Us
  - Shows Kollmorgen Support contact information.
- About
  - Shows the Kollmorgen Automation Suite copyright information.

11.10.3.2 Tabs

The controller Web server home page shows these tabs:



- "KAS Application tab" (→ p. 625)
- "Settings tab" (→ p. 640)
- "Backup and Restore tab" (→ p. 606)
- "Diagnostics tab" (→ p. 688)
- Help tab - This tab opens the WebHelp for the controller.

TIP

Browser Requirements: We recommend using Microsoft Edge, Mozilla Firefox, or Google Chrome to access the Web server.

11.10.4 Web server Home page

To access the KAS Web server home page, enter the controller's IP address.



Kollmorgen Automation Suite ™

Manufacturer	Kollmorgen
CPU Speed	*1.5 GHz*
CPU Cores	*4*
Model Name	*PCMM2G Essentials*
Model Number	*PCMM2G-C8-16-000-00-0000*
Serial Number	*KL2023180027*
CPU Serial Number	*77a0804e*
Hardware Revision	*1B*
TCP/IP Mac Address	*00:23:1b:01:10:e7*
EtherCAT Mac Address	*e4:5f:01:6e:da:93*
Axis Count	*8*

This page provides an overview of the device including:

- Manufacturer
- CPU Speed
- CPU Cores
- Model Name and Number
  - See "Controllers" (➔ p. 767) for the model name and number.
- Serial Number
- CPU Serial Number
- Hardware Revision
- TCP/IP MAC Address: A unique value associated with the TCP/IP network adapter that uniquely identifies the adapter on a LAN.
- EtherCAT MAC Address: A unique value associated with the EtherCAT network adapter that uniquely identifies the adapter on an EtherCAT network.
- Axis Count
  - See "Controllers" (➔ p. 767) for the motion axes count supported by each controller model.

11.10.4.1 Security

Some parts of the web server are locked to protect critical operations from unauthorized users.

- Log into the web server to enable access to the locked functions.
- See "User Authentication" (➔ p. 623).

NOTE

Functions are not locked if you access the web server through the KAS-IDE.
This automatically grants administrator access.

These functions are locked:

- "Backup and Restore" (➔ p. 623)
- "Diagnostics tab" (➔ p. 623)
- "KAS Application tab" (➔ p. 623)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Settings tab" (→ p. 623)

11.10.4.1.1 Backup and Restore

- Backup
- Restore
- Export Backup
- Choose File and Replace Backup

11.10.4.1.2 Diagnostics tab

- Reboot the controller.
- Clear errors and alarms.
- Clear a crash dump.

11.10.4.1.3 KAS Application tab

- Start, stop, or cold-start an application.
- Clear user data.
- Clear all errors.
- Configure auto-start.
- Apply or clear shared directory settings.

11.10.4.1.4 Settings tab

- Firmware upgrade - choose File and Upgrade.
- Reboot the controller or device.
- Reset to factory settings.
- SD card format.
- Change password.
- Change the network settings (IP address).

11.10.4.1.5.1 PCMM2G Only

- Configure the NTP.
- Update the time zone.
- Format a USB flash drive.

11.10.4.1.6 Timeout after Inactivity

To prevent misuse, if the webserver has been idle (no keyboard activity or mouse clicks) for 20 minutes, the user account is automatically logged out.

A dialog box opens to alert you that the session has timed out.

NOTE

The idle logout only occurs if you've logged into the webserver with a web browser.
Logging in through the KAS-IDE will never time out.

11.10.4.2 User Authentication

11.10.4.2.1 Login In

Login to the Web server is required to prevent unauthorized access or changes.

Click the **LOGIN** link at the top of all Web server pages. (Figure 15-2)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

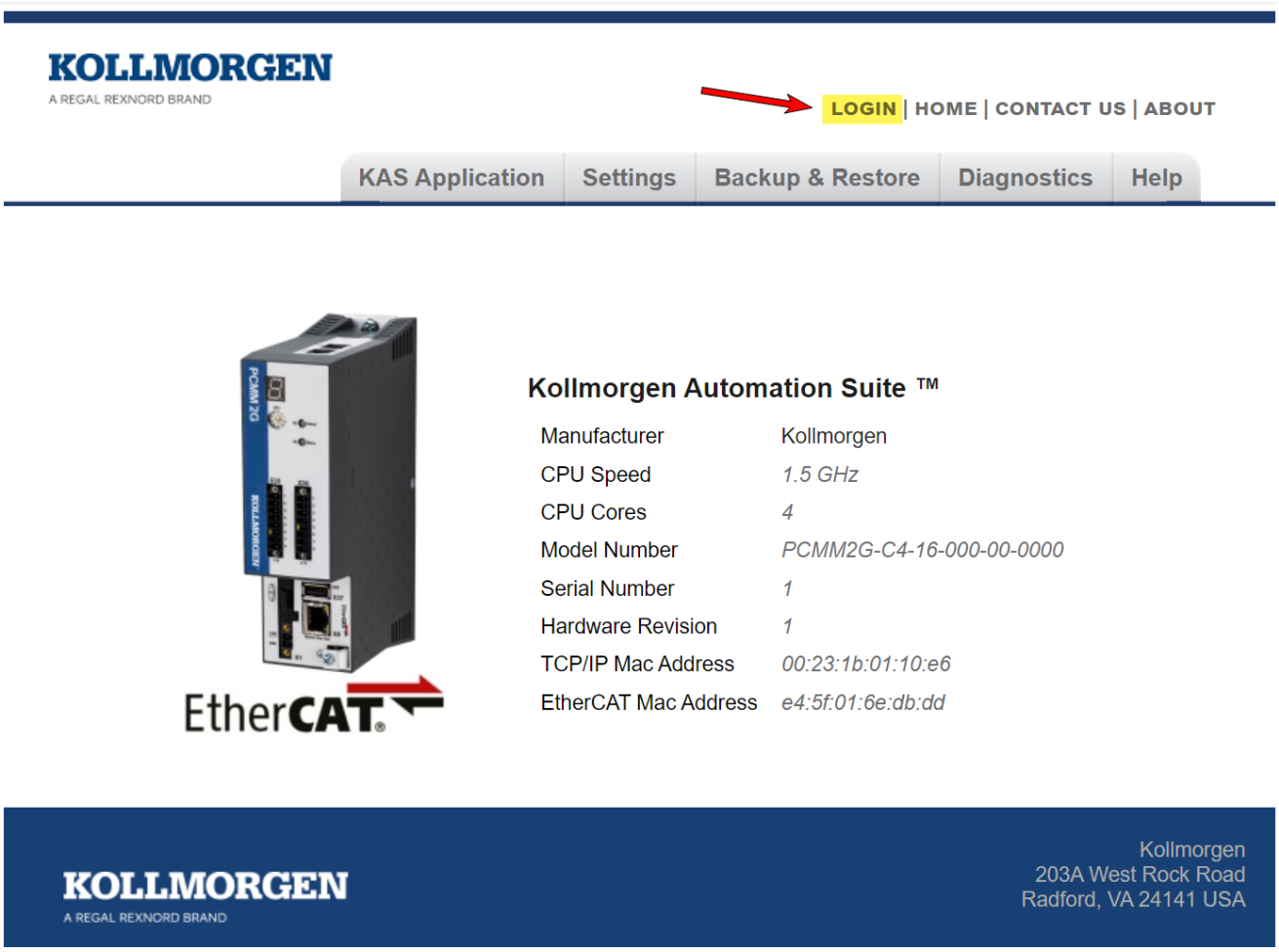


Figure 15-2: LOGIN link

The **Login** dialog opens. (Figure 15-3)

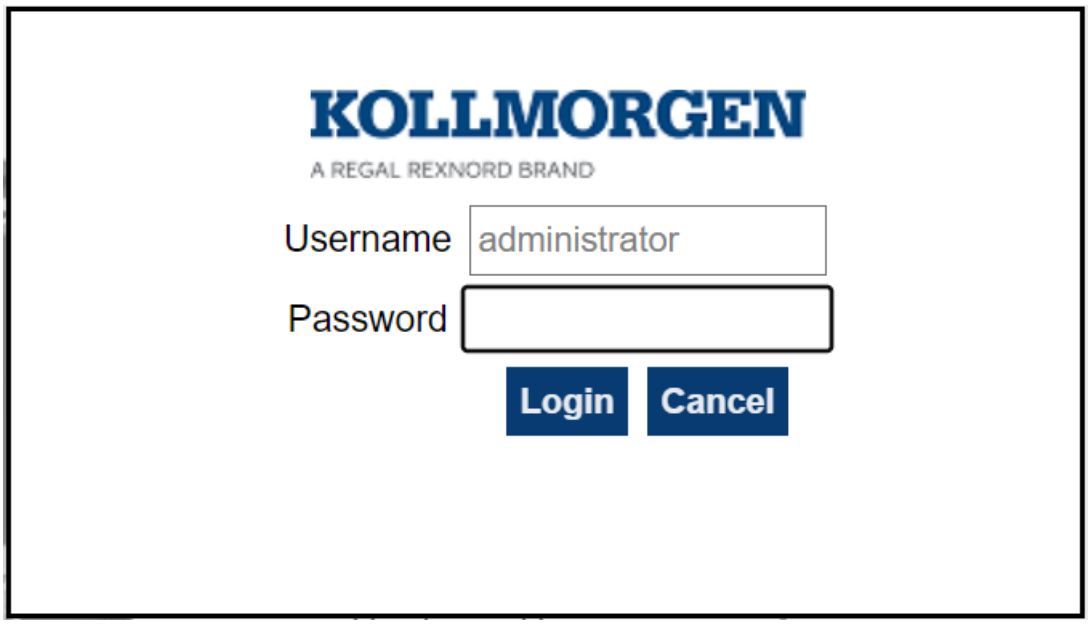


Figure 15-3: Login dialog

Enter the password to log in.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The factory default password is `administrator`.
This can be changed after logging in.

NOTE

There is only one user Administrator; **administrator** is the default user name.
- Re-enter the password each time you log in.
- The entered password is **not** stored in the **Login** dialog.
- Accessing the Web server from the IDE automatically logs you in as Administrator.

When successfully logged in, the user name appears in the top-right corner of all Web server screens. (Figure 15-4)

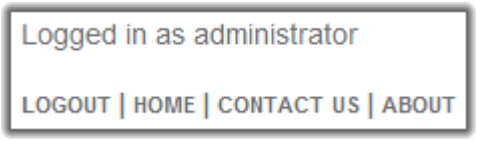


Figure 15-4: Logged in user name

11.10.4.2.2 Log Out

After successfully log in, the menu in the top right corner of the web server contains a link to **LOGOUT**.

1. Click this link to immediately log out of the Web server.
   A confirmation message appears. (Figure 15-5)

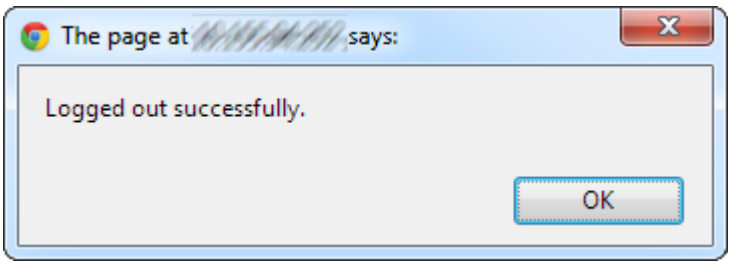


Figure 15-5: Logout confirmation message

2. Click **OK** to close the message.

11.10.4.2.3 Changing the Password

The user password is managed from the **User Account** tab of the **Settings** tab.

See "Change the User's Web server Account Password" (→ p. 679).

11.10.5 KAS Application tab

Usage

This tab is used to:

- Display general information about the currently loaded project on the controller.
- Start and stop the motion.
- Display the Axes run by the controller from the "Axis sub-tab" (→ p. 626).
- Manage log messages from the "Log Configuration tab" (→ p. 627) and "Log Data tab" (→ p. 628).
- Display User Data present on the controller from the "User Data tab" (→ p. 631).
- Connect to a remote computer from the "Shared Directory tab" (→ p. 634).
- Open the web-based Control Panels using the "Diagnostic Panels sub-tab" (→ p. 637).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Item	Description
Version of KAS App	This label provides information about the name and version number of the application that is in the controller. • The format is `<project_name>:<version>`. • The application's source code may be downloaded to the local computer if it is present on the controller. • Click the download button (). • This icon is found by the name and version information and is only present when source code is available.
Status of KAS App	The state of the application, either: **Started** or **Stopped**.
Start	• Default mode (warm start) where the Retain Variables are loaded at the application startup. • They are **Not** re-initialized; whereas other variables are started with their initial values. • See "Retain Variables" (→ p. 292).
Cold Start	• Use retain variables with their default values. • Such starts occurs from time to time but are few.
Stop	Stop the application.
Auto-start	Select this option to automatically start the KAS application when the controller is powered up. • The application starts using retained variables (a "warm start") after the controller has booted up. • To change this setting, click the **Auto-start** checkbox to either activate or deactivate this option and click the **Apply** button. • The control uses the new setting at the next power-up. • This option is recommended when the system is in production.
Clear all errors	Click this button to clear the error log for all axes.

11.10.5.1 Axis sub-tab

Usage

This sub-tab is used to:

- View a visual representation of the motors.
- The axis wheels are visible after the application is started.

These items can be monitored from the display:

- Real and Simulated axes.
- Actual position with solid line and actual position value.
- Command position with the dotted line and (command position value) in parentheses.
- Axis State: Powered-off, Powered-On, Error, and Simulated Powered Off and ON.
- Identify the axes from the label, as defined by the axis name in your application.
- Axis status or positions snapshot.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



Figure 15-6: Axis examples

NOTE

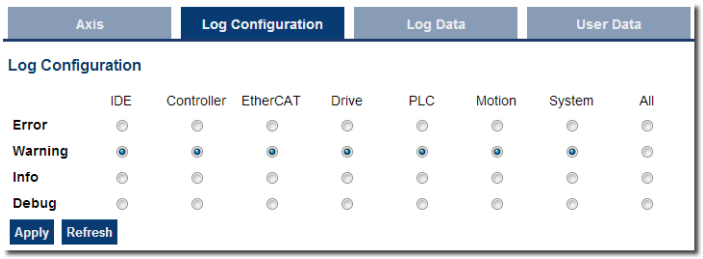
- Click the axis wheel to view the information in this table.
- If an axis is in error, click the text below the axis title to clear the error.

Title	Image (PN axis)	Image (PLCopen axis)	Image (digitizing axis)
AXIS1 **Axis status snapshot** Initialised true Power ON false Enable true Found true Configured true Running false Error false Simulated false Connected false Warning false Stopping false Stopped false	**AXIS1** **Axis positions snapshot** Actual 174.220505 Current 0.000000 Feedback 174.220505 Generator 0.000000 Pipe offset 0.000000 Pipe 0.000000 Power ON Delta Offset 0.000000 Reference 0.000000 Zero offset 0.000000 See "Axis Pipe Block" (→ p. 332).	**PLCOpenAxis2** **Axis positions snapshot** Actual 3141.856728 Command 3141.856728 Normal 3141.856728 Phase 0.000000 Super imposed 0.000000 See "Axis Parameters" (→ p. 347).	**Axis3** **Axis positions snapshot** Actual 0.000000

11.10.5.2 Log Configuration tab

You can configure the log to filter the messages that are displayed.

Each "Source" (→ p. 628) can be set with its own "Level" (→ p. 628).



Each message has one of these levels, with importance in descending order:
Error > **Warning** > **Info** > **Debug**

11.10.5.2.1 Choose the Appropriate Level

- When a level is set for a source, only messages with the same or higher importance are recorded.
  - Example: If a source is set to WARNING, then all messages with levels WARNING, ERROR and CRITICAL are recorded (DEBUG and INFO messages are discarded).
- DEBUG is the most verbose and ERROR is the least verbose level.
  - Filtering is quicker with less verbose levels because of the number of messages.

NOTE

Critical messages are always recorded.
The Critical level is not visible.

11.10.5.2.2 Source

Source	Apply to...
Controller	For the KAS Runtime items: Drivers, IOEngine, SinopEngine, etc.
Drive	Messages from Kollmorgen drives.
EtherCAT	For all kinds of EtherCAT items: Motion bus, I/Os
IDE	Win32 applications: • KAS-IDE • KAS Runtime Server (i.e., the KAS Runtime Front-end)
Motion	Messages coming from the Motion engines: • Pipe Network • PLCopen • VM
PLC	For application engineers to create a custom log within the PLC programs. See "Printf Function" (→ p. 272).
System	For common API and libraries. Includes messages issued from the operating system.

11.10.5.2.3 Level

Level	Icon	Description
CRITICAL		• Application crashes or becomes unstable. • Data is corrupted. • The application behavior can be unpredictable.
DEBUG		• Any information logged for development purpose. • This log can be safely ignored.
ERROR		The application does not behave as expected but the processes remain stable.
INFO		• Information status of the current process. • This log can be safely ignored.
WARNING		• System is stable but the KAS-IDE warns that an unexpected event can occur. • This log can be safely ignored.

11.10.5.3 Log Data tab

KAS log files may be viewed from the Log Data tab.

These messages can help describe the current state of the system and identify any operation errors.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

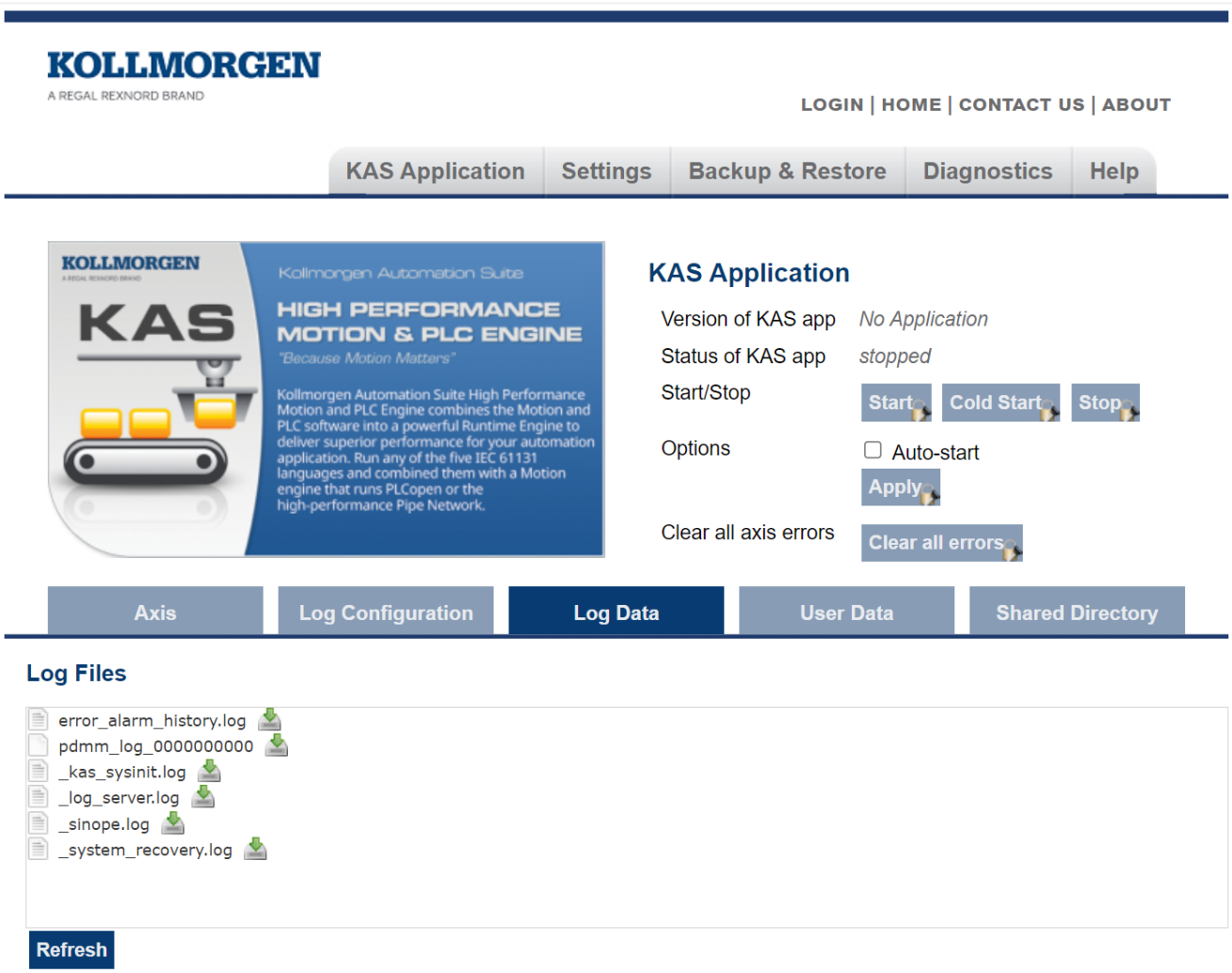


Figure 15-7: Example of Log Files from a Kollmorgen Controller's Web server

Click a listed log file to open it in a web browser.

- Click the green download icon next to the log entry to download the log file.
- The default name is the same as the file's name.
- If a file no longer exists, the message **/logfiles/<selected file name> not found.** appears.
  - Refresh the browser window and try again.

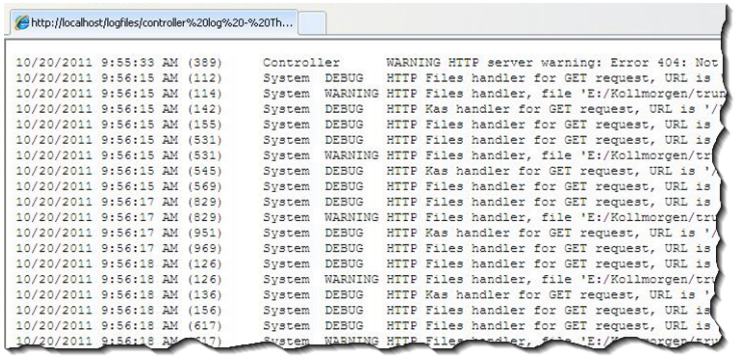


Figure 15-8: Example of a log file's content in a browser.

NOTE

- Controller log data is collected and updated every 15 seconds.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- A new log file is created when the current file is full.
- You may need to wait for up to 15 seconds for a log to show up in the list.

11.10.5.3.1.1 PCMM2G Only

- The file `error_alarm_history.log` contains a history of errors and alarms.
- Files whose names begin with an underscore are internal.
  - They may be requested by support engineers during troubleshooting.

11.10.5.3.2 Log Message Content

Log files contain one message per line, formatted as a CSV file.

Every log message contains this information:

Field	Description
Time	Time when the log was recorded with the format: DD-MMMM-YY hh:mm:ss (millisecond)
Source	Identifies a software or hardware component issuing the messages. Each source is configured with a specific Level. See "Log Messages Settings" (➔ p. 124).
Level	Each message has one of these levels with importance in ascending order: DEBUG > INFO > WARNING > ERROR > CRITICAL. See "Log Messages Settings" (➔ p. 124).
Message	Text of the message issued from the source.
Origin	The source code line where the message originated.

TIP

Log messages are an important source of information when you are troubleshooting your project.
When reporting an issue to Kollmorgen Support, copy/paste the logs in the report.

11.10.5.3.3 Controller Log Files

- PCMM2G generated logs are stored in flash memory at `/home/kas/kas/log`.

Files are stored in a rotating pool consisting of a maximum of 10 files.

- The files have a maximum size of 200kb each.
- The most amount of space the log files will consume is 2MB.
- When a files is created after the tenth file, the earliest file is deleted to make room for the new file.

The controller-generated log levels can be controlled from the KAS-IDE and Web server.

KAS-IDE-generated log levels can be filtered in the configuration window in the **Logs and Information** tab.

11.10.5.3.4.1 Log File Naming Convention

Logs have this naming convention: `pdmm_log_n`.

Where:

- `n` is a value ranging from 0000000000 to 4294967295.
- This is the maximum value a 32-bit location can store.

Example

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- When the files are first created they are named:
  - `pdmm_log_0000000000`
  - `pdmm_log_0000000001`
  - `pdmm_log_0000000002`, etc.
- The file that is created after `pdmm_log_4294967295` is `pdmm_log_0000000000`.
  - The naming is reset and continues.

11.10.5.3.5.2 Error and Alarm History File (PCMM2G only)

The PCMM2G stores a file named `error_alarm_history.log` at `/home/kas/kas/log`.

This file stores a history of "Controller Error and Alarm Descriptions" (➔ p. 577) that have occurred on the controller.

11.10.5.4 User Data tab

The User Data tab provides access to the User Data Storage area on the Controller's internal flash memory. (Figure 15-9)

- This area is intended for user-managed file storage to support PLC application needs.
- The system does not enforce structure or usage – it is entirely up to the user to organize and maintain the content.

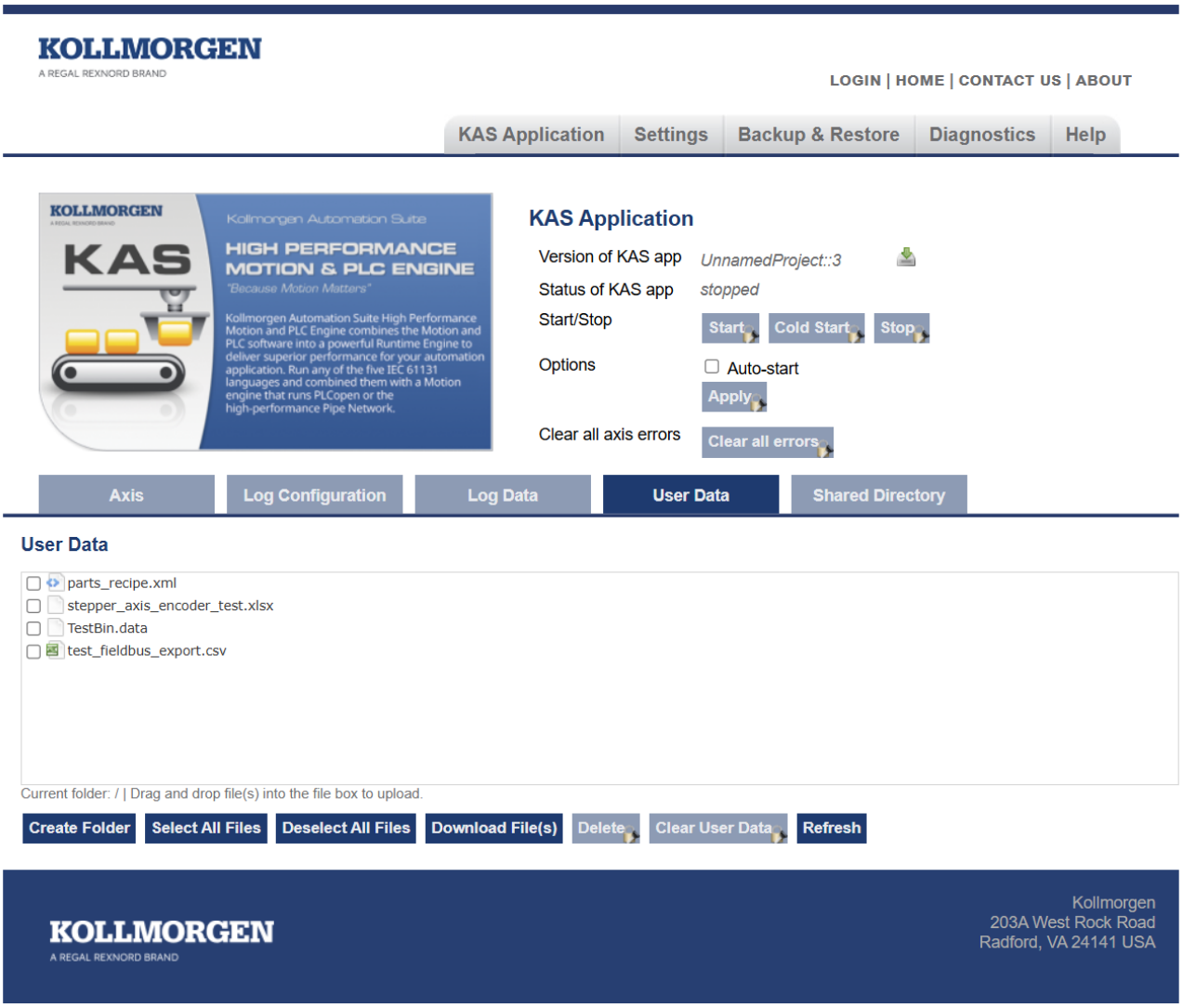


Figure 15-9: User Data tab

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The User Data area is flexible, user-controlled storage space where you can maintain folders and files required for application management.

Typical use cases include:

- Storing and updating recipe files.
- Maintaining lookup tables or datasets.
- Exchanging data between external systems and the controller.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

User Data File Explorer

- Display all folders and files currently stored in the User Data area on the controller flash.
- Each folder and file has a check box for selecting an action.

IMPORTANT

- Folders can only be deleted.

11.10.5.4.1 View File Contents from the User Data File Explorer

1. Double-click any file in the list.
   The file contents open in a separate tab. (Figure 15-10)

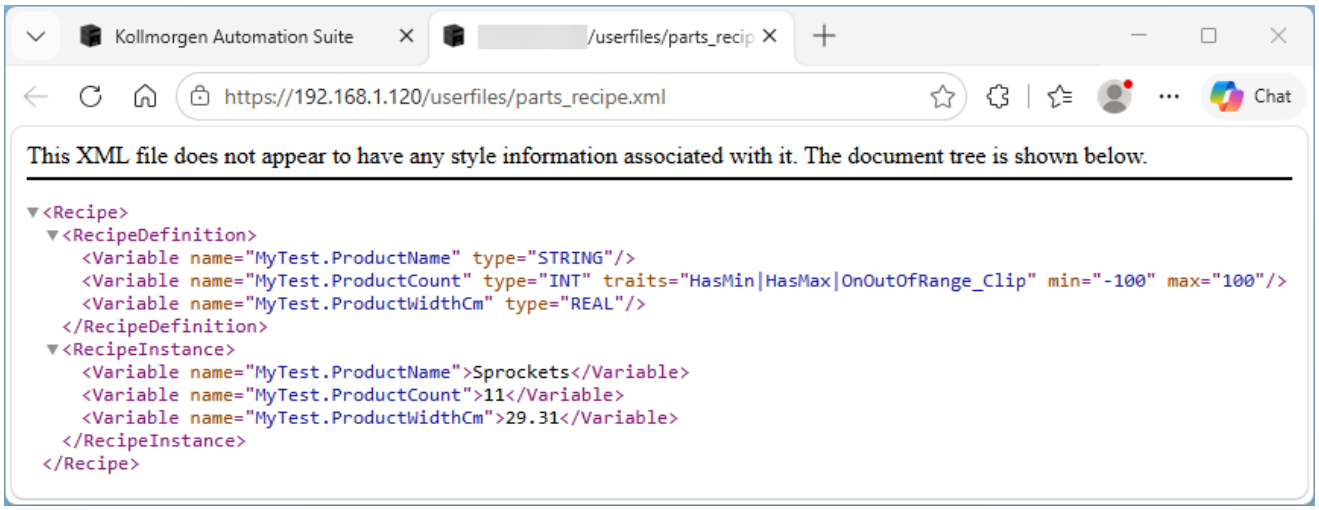


Figure 15-10: User Data tab - View File Contents

This new tab:

- Provides quick access for inspecting file data without downloading.
- Is useful for verifying configuration or recipe files.

11.10.5.4.2 Upload Folders and Files to the Controller

Folders and files can be uploaded to the PCMM2G controller flash using drag-and-drop.

1. On your computer, open Windows Explorer.
2. Locate and select the applicable folders and/or files.
3. Drag and drop the selected items into the User Data area.

   - Folders and files are stored immediately in the controller's User Data area.
   - Multiple folders and files can be uploaded at once.
   - There is no upload button – drag-and-drop is the only available method.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

11.10.5.4.3 File Management Actions

All actions apply directly to the folders and files stored on the controller flash:

11.10.5.4.4.1 Create Folder

- Create a new folder in the current directory.

11.10.5.4.5.2 Select All Files

- Selects all files in the list.
- Primarily used to support bulk operations such as:
  - Downloading multiple files.
  - Deleting multiple files.

11.10.5.4.6.3 Deselect All Files

- Deselects all selected files.

11.10.5.4.7.4 Download Files

- Downloads selected files from the controller to your local computer.

IMPORTANT

- Downloading folders is not supported.

11.10.5.4.8.5 Delete

- Deletes the selected folders and files from the controller.

WARNING

- Before deleting folders and files, confirm they are no longer required by the application and create a backup if needed.
- Deleting folders and files cannot be undone.

11.10.5.4.9.6 Clear User Data

- Deletes all folders and files in the User Data area.

WARNING

- Use with caution - this removes all user-managed content.
- This action cannot be undone.

11.10.5.4.10.7 Refresh

- Updates the file explorer to reflect current contents on the controller.

11.10.5.5 Shared Directory tab

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

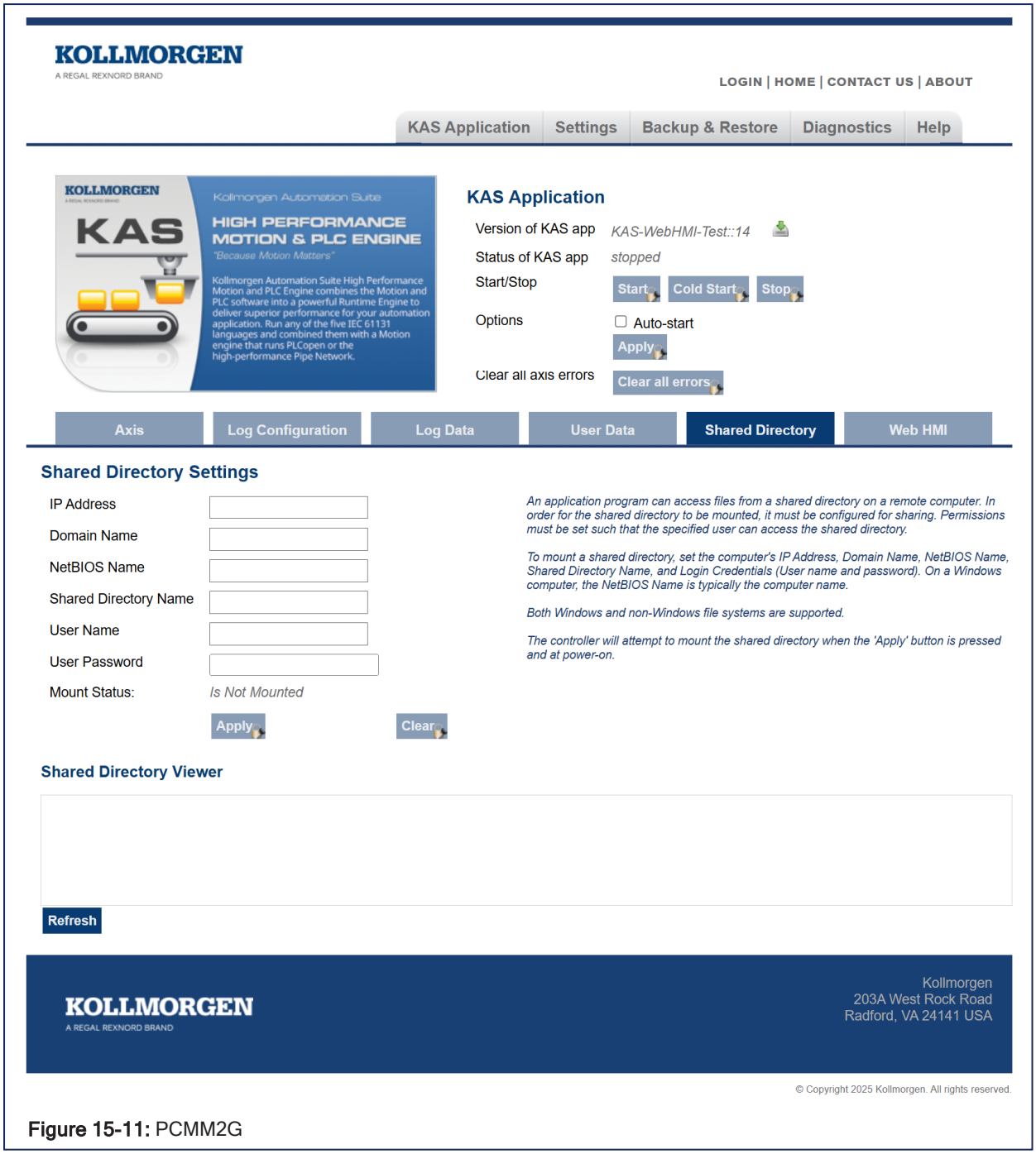


Figure 15-11: PCMM2G

Usage

This tab is used to connect to and mount a shared directory on a remote computer (Window 7 or later and non-Windows).

This allows a program to read and parse instructions for motion.

Field Descriptions

These fields must be set to connect to the share.

NOTE

See this Microsoft article File Sharing Over a Network in Windows to share a directory.

Field	Description
IP Address	Enter the IP Address of the computer that has a shared directory.
Domain Name	• The network domain of the computer that is sharing the directory. • This field may or may not be necessary.
NetBIOS Name	• The unique name that identifies the computer on the network. • For Windows computers, this is typically the computer name.
Shared Directory Name	• Enter the name of the shared directory as defined on the remote computer. • The contents of this directory is listed in the **Shared Directory Viewer**.
User Name	• The name of a user allowed access to the shared directory. • If the user name is a domain user (instead of a local user), the computer that hosts the shared folder must be connected to the domain to validate the name.
User Password	The password of a user allowed access to the shared directory.
Windows File System	• Indicate whether the remote computer is Windows or a different operating system. • This is used to automatically set file permissions. • This checkbox is not visible for PCMM2G. • File permissions set on the remote computer are automatically managed on PCMM2G.
Mount Status	This field indicates whether the directory **Is Mounted** or **Is Not Mounted**.
Apply	Click this button to: • Mount the share. • Save the configuration to flash memory so it can be automatically mounted at power-on. • This button is disabled when the application is running and when the user is not logged in.
Clear	Click this button to: • Clear all the fields. • Unmount the share. • Clear the configuration from flash memory. • This button is disabled when the application is running and when the user is not logged in.
Shared Directory Viewer area	This area: • Lists the contents of a mounted shared directory. • Is read-only. Once the shared directory is mounted, its contents can be navigated in the **Shared Directory Viewer** section of this tab.
Refresh	Click this button to refresh the contents of the fields and the **Shared Directory Viewer** area.

TIP

We strongly recommend consulting with your IT department for setting up and connecting to a shared directory. With differences between operating systems and the variability of security settings, this can become challenging.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

SMB1 must be enabled to use the Shared Directory feature with a Windows file system.

See Shared Directory Path Conventions.

11.10.5.6 Diagnostic Panels sub-tab

Usage

This sub-tab is used to:

- View Control Panels designed in the KAS-IDE in a web browser.
  - This feature is only available when the **Generate Web-based Control Panels** check box is selected in the "Project Settings dialog box" (➔ p. 461).

Access

1. Open a web browser.
2. Enter the controller's IP address for the URL.
   The Web server Home page opens.

   NOTE

   See Find the Controller IP Address.
3. Click the **KAS Application** tab.

   The **Diagnostic Panels** sub-tab is on the right of the window. (Figure 15-12)

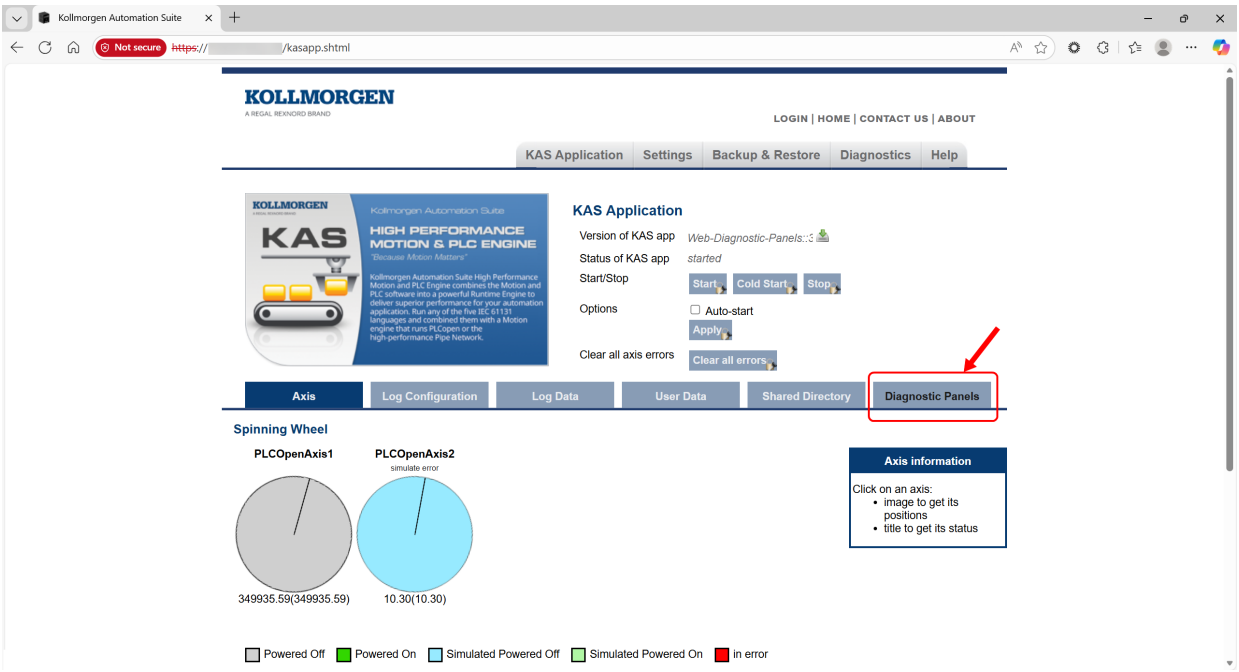


Figure 15-12: Diagnostic Panels sub-tab

Tab Description

Click this tab to open the "Diagnostic Panels - Home page" (➔ p. 637).

IMPORTANT

This tab is only available when the **Generate Web-based Control Panels** check box is selected in the Project Settings dialog box.

11.10.5.6.1 Diagnostic Panels - Home page

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Usage

This window is used to:

- View Control Panels designed in the KAS-IDE in a web browser.
  - This tab is only available when the **Generate Web-based Control Panels** check box is selected in the Project Settings dialog box.
- See Web Diagnostic Panels to enable this feature.

Access

1. Open a web browser.
2. Enter the controller's IP address for the URL.
   The Web server Home page opens.

   NOTE

   See Find the Controller IP Address.
3. Click the "Diagnostic Panels sub-tab" (➔ p. 637) on the right of the window. (Figure 15-13)

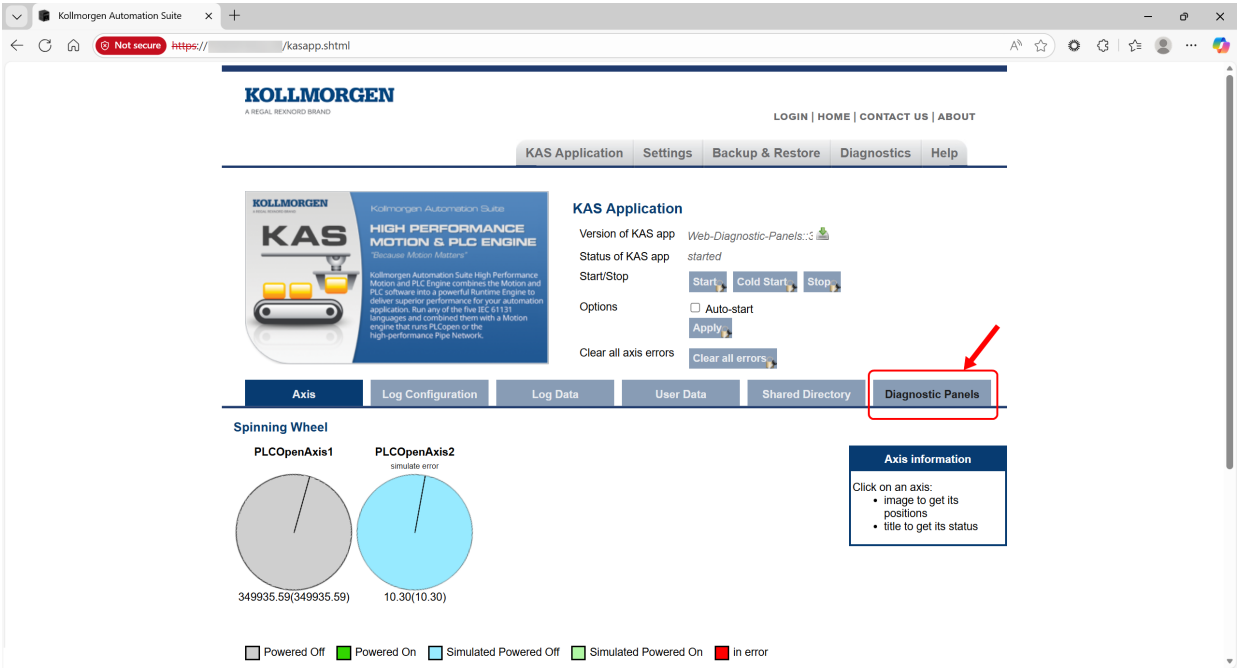


Figure 15-13: Diagnostic Panels sub-tab

The Diagnostic Panels - Home page opens. (Figure 15-14)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

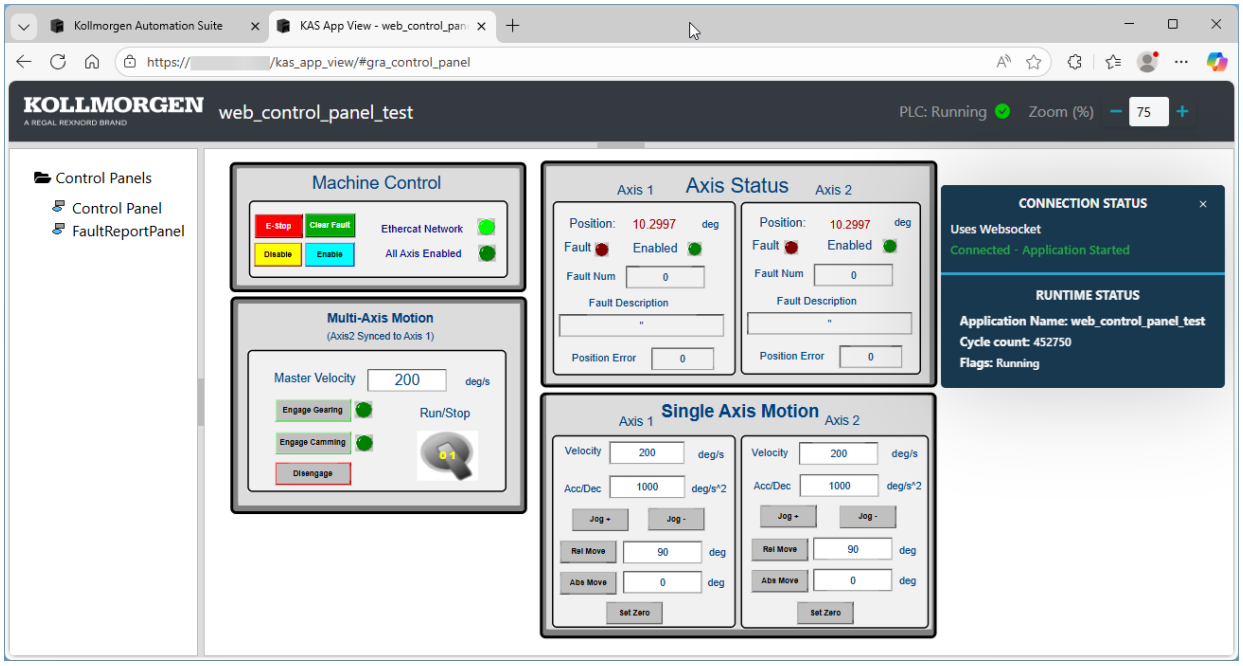


Figure 15-14: Example: Diagnostic Panels - Home page

Window Description

NOTE

The content of this window depends on the specific application project controls.

The window shows this information:

- Connection status with KAS Runtime.
- Control page zoom setting adjustments.
- A collapsible side bar with links to view all control panels present in the KAS project.
  - See "Design the HMI with the Internal Editor" (➔ p. 715).

Notes

- After a period of inactivity, the Diagnostic Panel session times out and disconnects from the KAS Web server.
  - This is indicated by the message Connection is closed displayed on the top bar. (Figure 15-13)

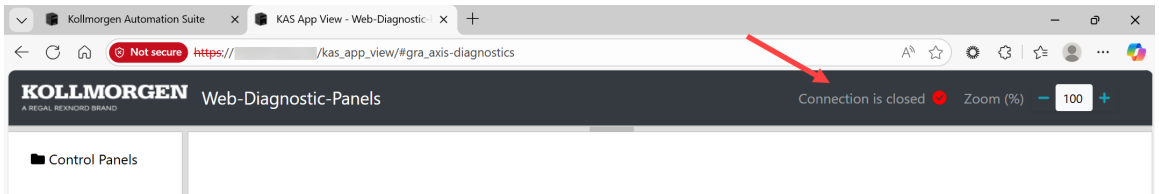


Figure 15-15: Diagnostic Panel - Connection closed

- During this time, widgets in the panel stop refreshing with new data.
  - Text widgets may display a question mark (?) or remain blank.
  - This indicates there is no active connection to the KAS Web server.
- To restore the connection and update the data, reload the page using the browser's refresh option.
- When the KAS project is not running, the top bar shows this message: **PLC: Stopped**. (Figure 15-16)



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Figure 15-16: Diagnostic Panel - PLC Stopped

- During this time, widgets in the panels are inactive.
  - Text widgets may show a question mark (?) or remain blank.
- When the project starts, the status bar and data update automatically.
  - If data does not refresh, reload the page using the browser's refresh option.

See Also

"Diagnostic Panels sub-tab" (➔ p. 637)

11.10.6 Settings tab

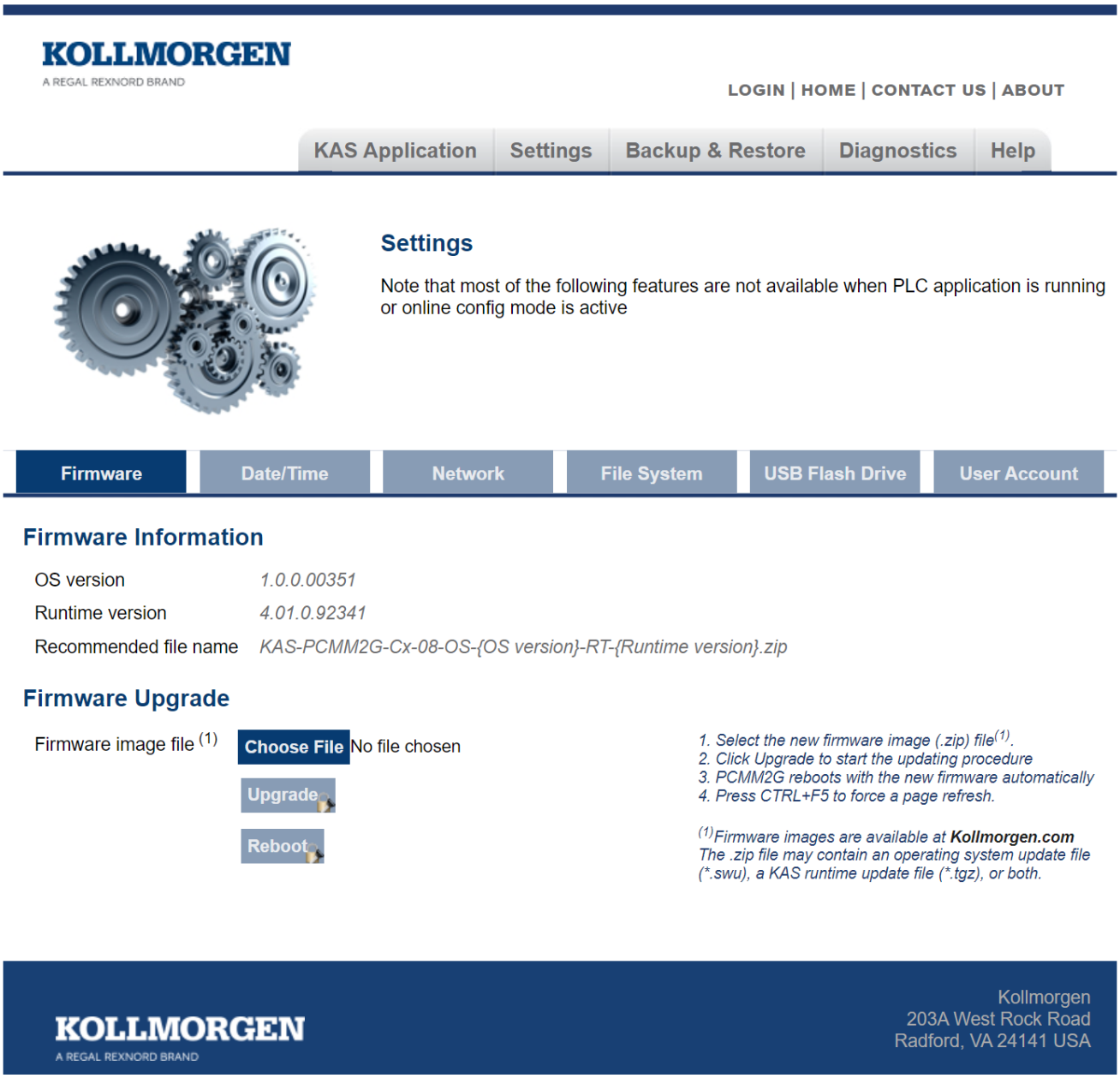


Figure 15-17: Settings tab with Active Firmware tab

The **Settings** tab is used to:

- View and update the firmware for the KAS Runtime.
  - See "Upgrade the Controller's Firmware" (➔ p. 668).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Change the date or time in the "Date/Time tab" (➔ p. 662).
- View the network settings and modify the IP address.
  - See "Network tab" (➔ p. 675).
- "Reset the Controller to Factory Settings" (➔ p. 603)
- PCMM2G:
  - Use the "USB Flash Drive tab" (➔ p. 679) for a USB flash drive.
  - Use the "Security tab" (➔ p. 676) to:
    - Change the user's Web server account password.
      See "Change the User's Web server Account Password" (➔ p. 679).
    - Manage remote access to the controller via SSH.
      See "SSH Connection" (➔ p. 677).
- Access the "Change the User's Web server Account Password" (➔ p. 679) to change the password.

11.10.6.1 Certificates tab

This tab is used to:

- View the current SSL Server Certificate's details.
  - See "SSL Server Certificate Details" (➔ p. 641).
- Create SSL Server Certificates for HTTPS connections.
  - "Create SSL Server Certificate (Self-Signed)" (➔ p. 642).
- Upload SSL Client Certificates for HTTPS connections.
  - See "Upload SSL Client Certificate" (➔ p. 642).
- Manage the SSL Client Certificates on the PCMM2G.
  - See "SSL Client Certificates" (➔ p. 642).

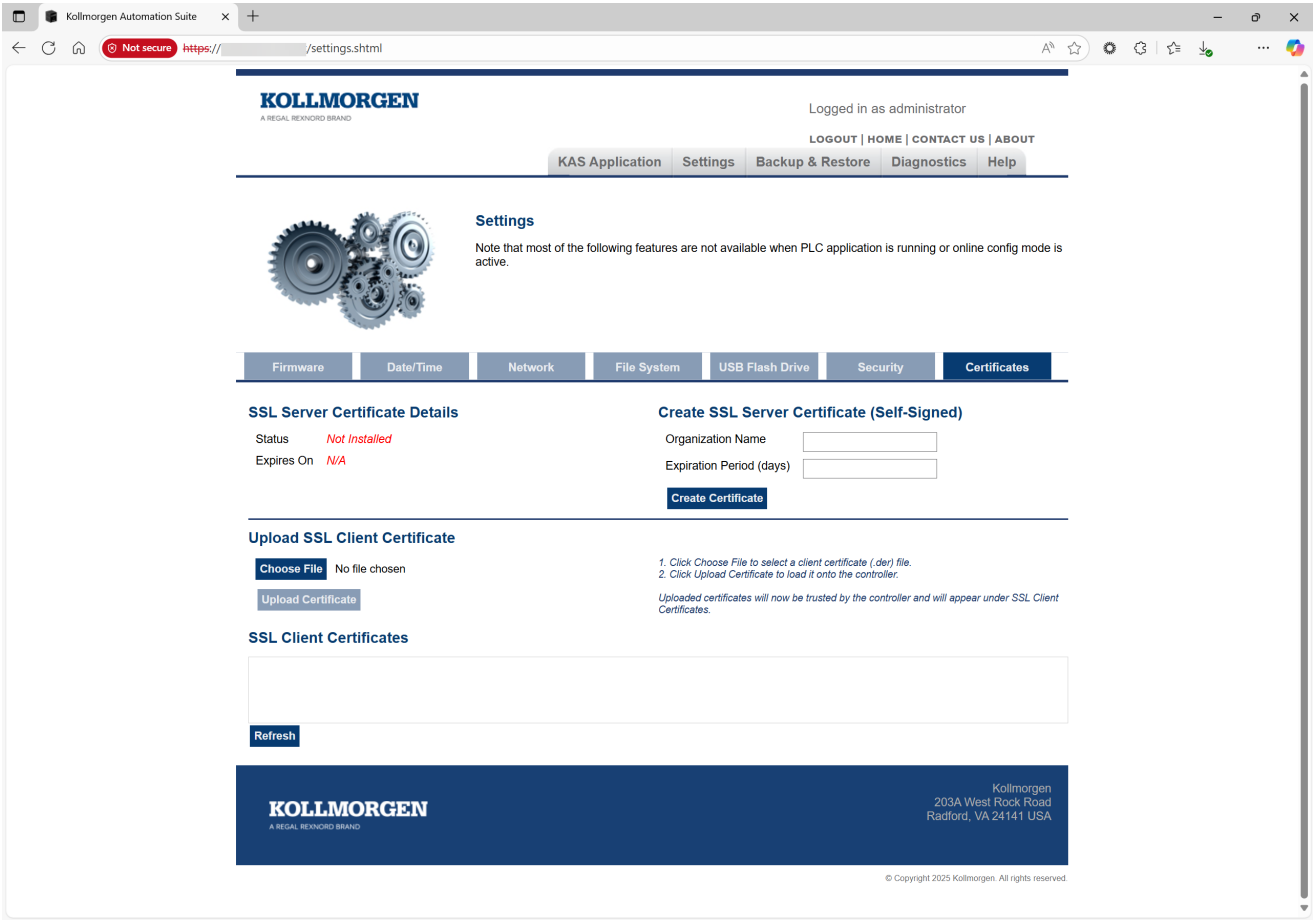


Figure 15-18: Certificates tab

11.10.6.1.1 SSL Server Certificate Details

This area is used to view the status of the certificate and its expiration date.

The certificate statuses are:

- Installed
- Not Installed
- Invalid - IP Address Mismatch
- Invalid - Expired

11.10.6.1.2 Create SSL Server Certificate (Self-Signed)

This area is used to create a Secure Sockets Layer (SSL) certificate on the PCMM2G.

- SSL certificates:
  - Are used to enable HTTPS connections on the PCMM2G.
  - Enable encrypted communication between a client (e.g., web browser, OPC UA client) and a PCMM2G controller to ensure sensitive data (e.g. passwords) cannot be intercepted or read by third parties.
- When creating an SSL certificate, the PCMM2G's current IP address is linked to the certificate with an expiration date.
  - The certificate is valid only when the IP address listed on the certificate is identical to the PCMM2G's IP address.
  - It is recommended to set a static IP address for the PCMM2G when using an SSL certificate.
- To ensure secure and trusted communications with the Web server through HTTPS, the SSL certificate must be installed on the user's machine after creating it.
  - See "Install the SSL Server Certificate (Self-Signed)" (→ p. 642).

11.10.6.1.3 Upload SSL Client Certificate

This area is used to select and upload SSL client certificates to enable secure and trusted communication between a PCMM2G and other devices.

- All valid SSL certificates uploaded through this interface are automatically added to the trusted certificate store of the PCMM2G.
- Once uploaded, these certificates are used by the OPC UA Server running on PCMM2G to authenticate incoming client connections.
- If a client attempts to connect using a certificate that is not in the trusted store, the PCMM2G OPC UA Server rejects the connection.

11.10.6.1.4.1 OPC UA Server Integration

The OPC UA Server on a PCMM2G leverages the uploaded SSL certificates using:

- **Authentication**: Verifies the identity of OPC UA clients using their X.509 certificates.
- **Encryption**: Ensures secure data exchange using the public key from the client certificate.
- **Trust Management**: Only clients with certificates in the trusted store can establish secure sessions when using **Sign** or **SignAndEncrypt** security modes.

This aligns with the OPC UA specification's security model, ensuring that communication is both authenticated and encrypted, and that only trusted clients can interact with the controller.

11.10.6.1.5 SSL Client Certificates

This area lists all SSL Client Certificates that have been uploaded to the trusted certificate repository on the PCMM2G controller.

Users can also download or remove certificates from the trusted store directly through this interface.

11.10.6.1.6 Install the SSL Server Certificate (Self-Signed)

IMPORTANT

This procedure is only for PCMM2G controllers.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The Secure Sockets Layer (SSL) certificate is a file stored on the PCMM2G that enables encrypted connections (HTTPS).

- The SSL certificate is only applicable to PCMM2G.
- By default, the PCMM2G contains a certificate generated by Kollmorgen.
- The web browser shows a warning message when the connection is not secure.
  - To prevent this warning, the default certificate should be replaced by creating a new certificate in the "Security tab" (➔ p. 676).
  - This certificate must be downloaded and installed onto the PC connected to the PCMM2G.
  - This user-created certificate is assigned to the PCMM2G's IP address.
    - The certificate becomes invalid if this IP address is modified.

11.10.6.1.7.1 Certificate Expiration Period

- The certificate expiration period is set by the user.
  - Kollmorgen recommends 365 days.
- When the expiration period is over, the certificate becomes invalid.
- The user must create and install a new certificate to prevent security warnings.

11.10.6.1.8.2 Installation Procedure

Assumptions

The PCMM2G is connected to the router or laptop.
See the PCMM2G Installation Manual.

IMPORTANT

- The images in this procedure are from the Microsoft Edge browser.
- Other browsers have different options.
- It's recommended to have only one instance of the web browser open for this procedure.

NOTE

- The self-signed SSL Certificate generated from this procedure can be used by the OPC UA Server to enable encrypted and authenticated data exchange between OPC UA clients and the PCMM2G controller.
- See "OPC UA - Create New Driver Configuration" (➔ p. 931) for setup instructions.

Procedure

1. Open a web browser.
2. Enter the controller's IP address for the URL.
   The default IP address for the PCMM2G is 192.168.0.101.
   A connection message appears. (Figure 15-19)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

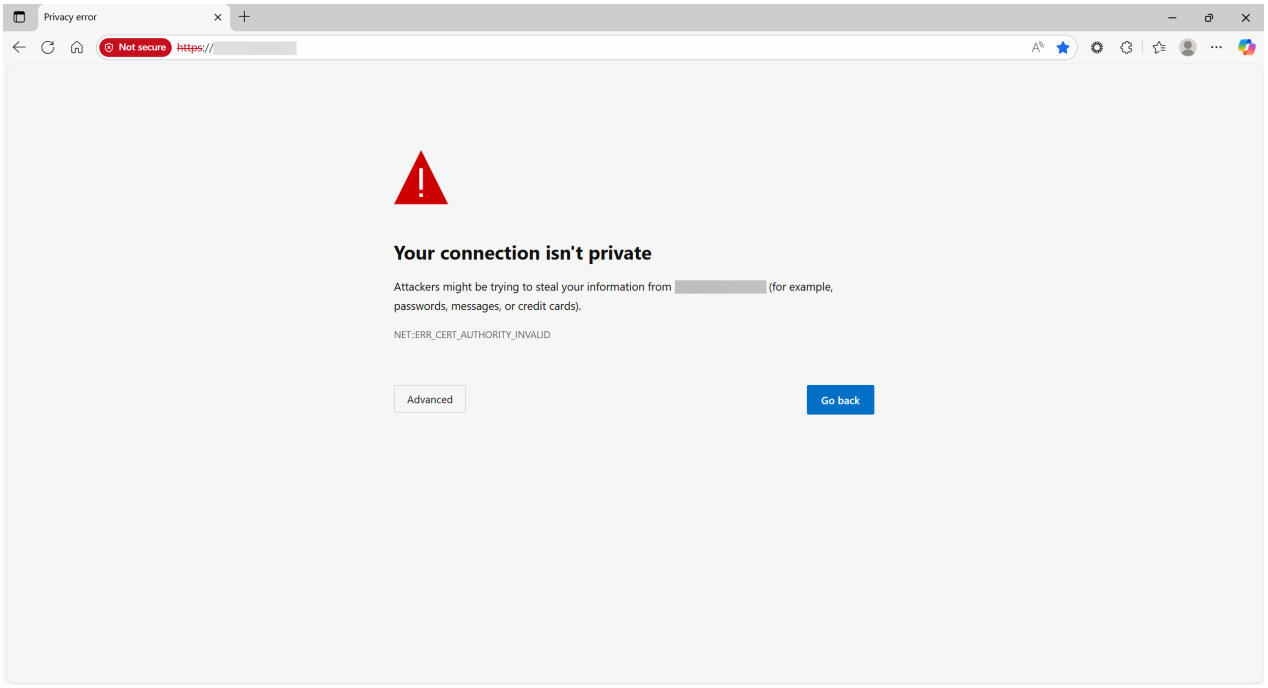


Figure 15-19: Connection message

3. Click the **Advanced** button.
   Additional information about the connection appears. (Figure 15-20)

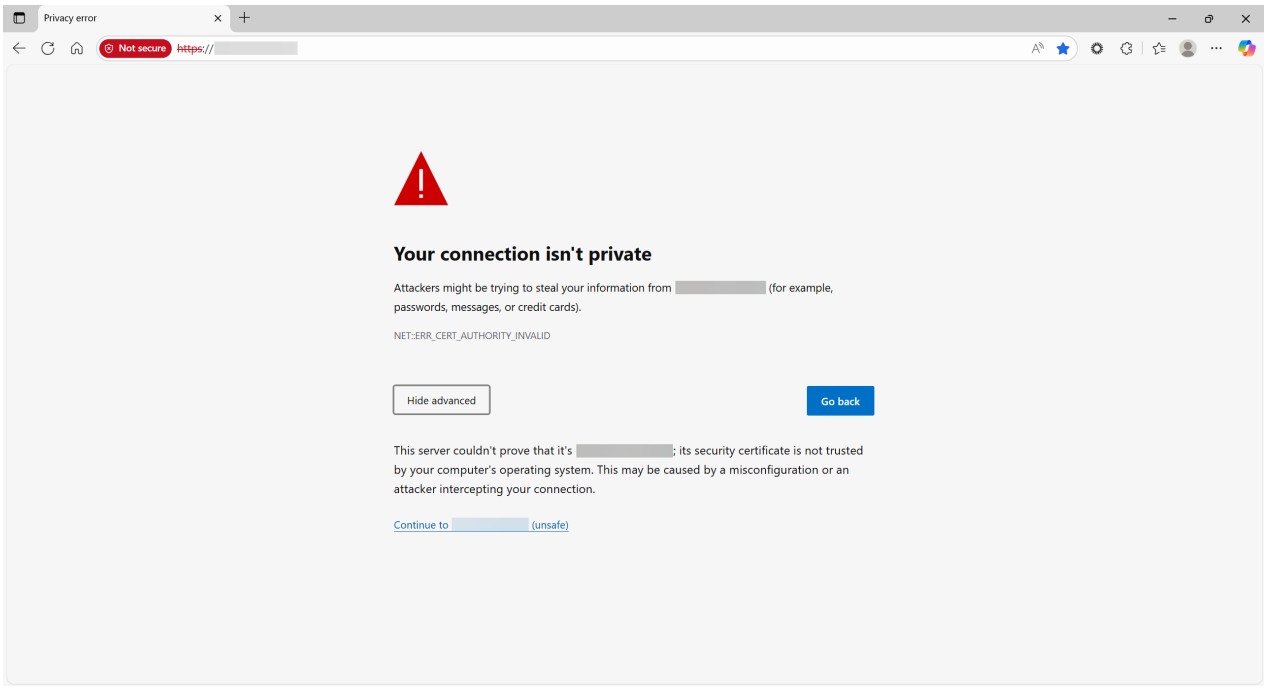


Figure 15-20: Connection message - Advanced information

4. Click the **Continue to (IP address) (unsafe)** link.
   The Web server opens for the PCMM2G. (Figure 15-21)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

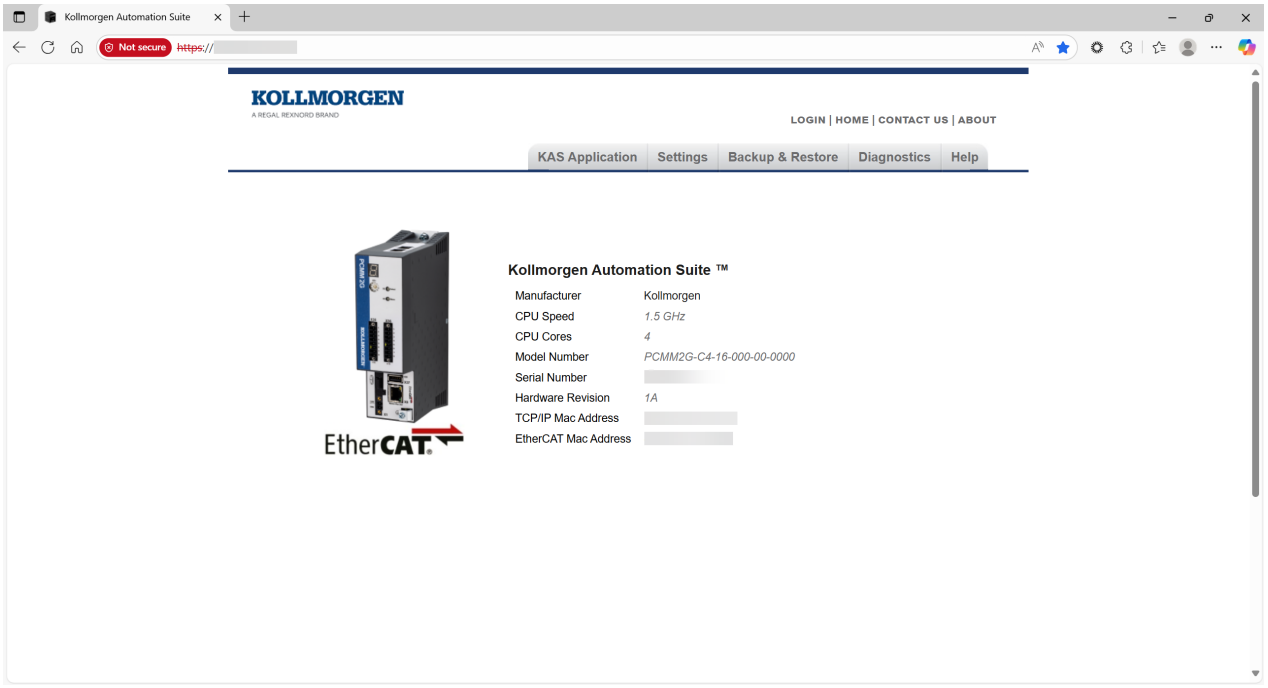


Figure 15-21: Not Secure PCMM2G

5. Login to the PCMM2G.
   See "User Authentication" (➔ p. 623).
6. Click the **Settings** tab.
7. Click the **Certificates** tab. (Figure 15-22)

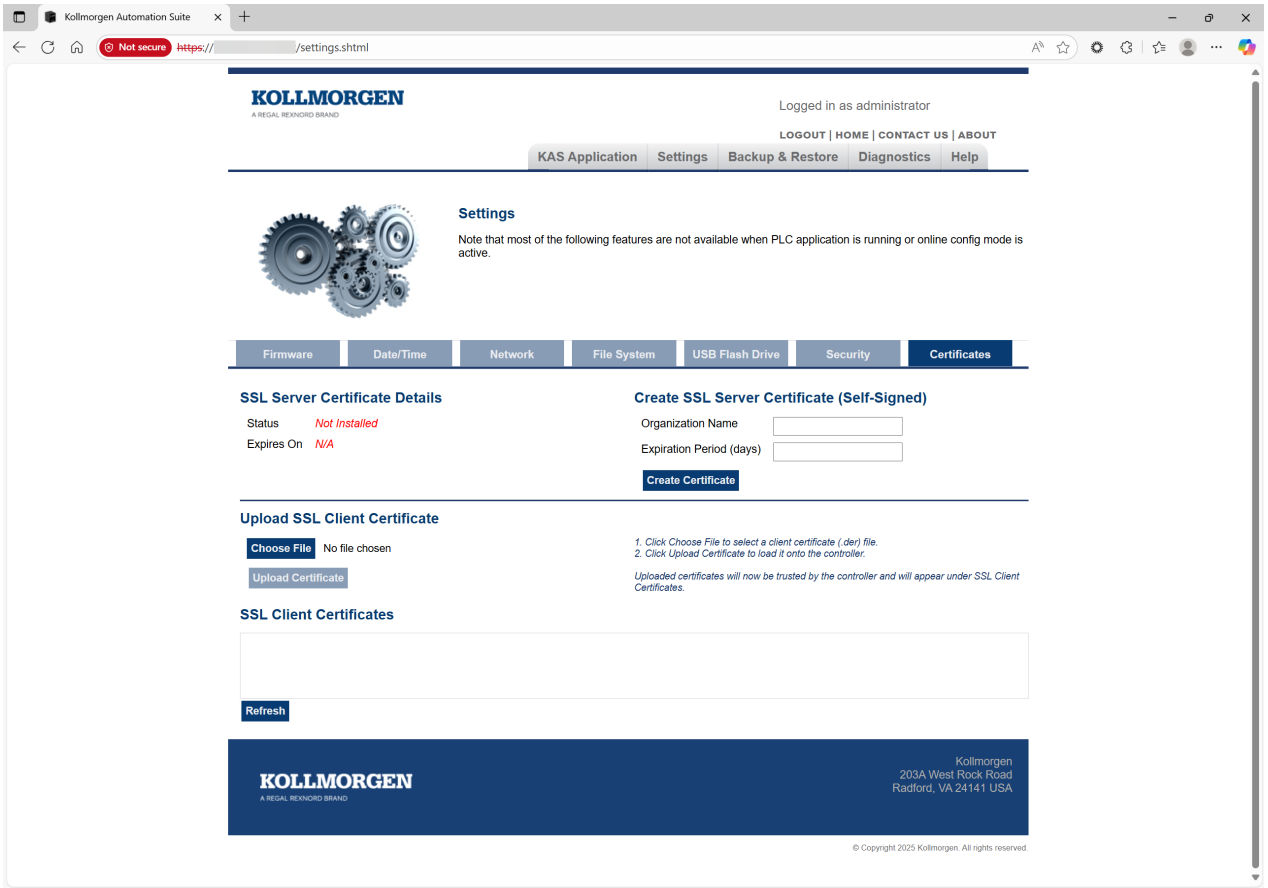


Figure 15-22: Certificates tab

8. In the **Create SSL Server Certificate (Self-Signed)** area:
   a. Enter the **Organization Name**.
   b. Enter the **Expiration Period (days)**. (Figure 15-23)

TIP

Kollmorgen recommends 365 days.

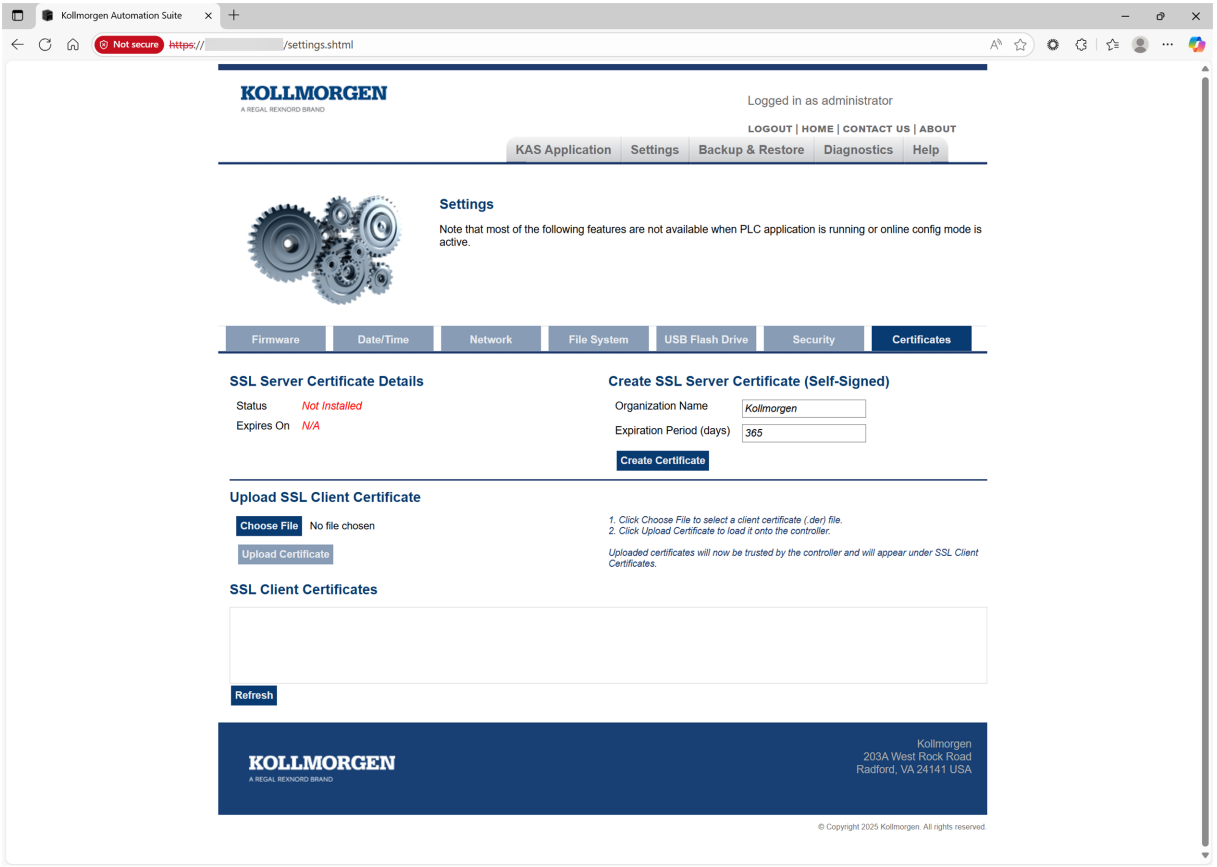


Figure 15-23: Create SSL Certificate area with content

   c. Click the **Create Certificate** button.
   A confirmation message appears. (Figure 15-24)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

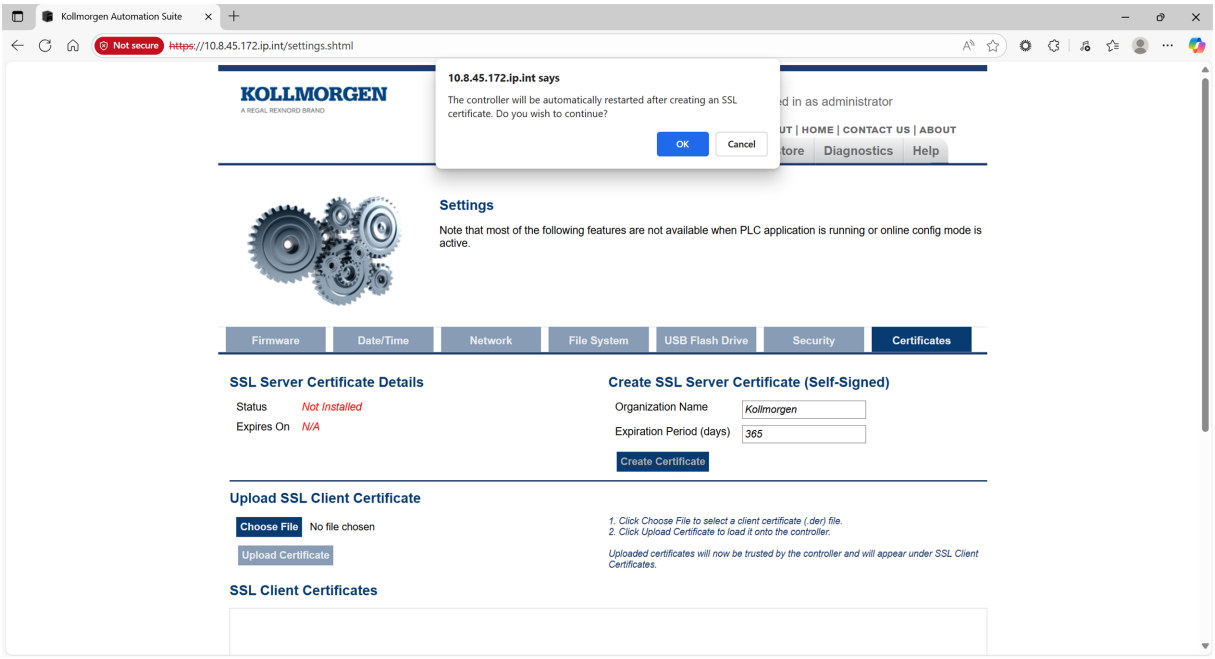


Figure 15-24: Confirmation message

9. Click **OK** to continue.
   The SSL certificate successfully created message appears. (Figure 15-25)

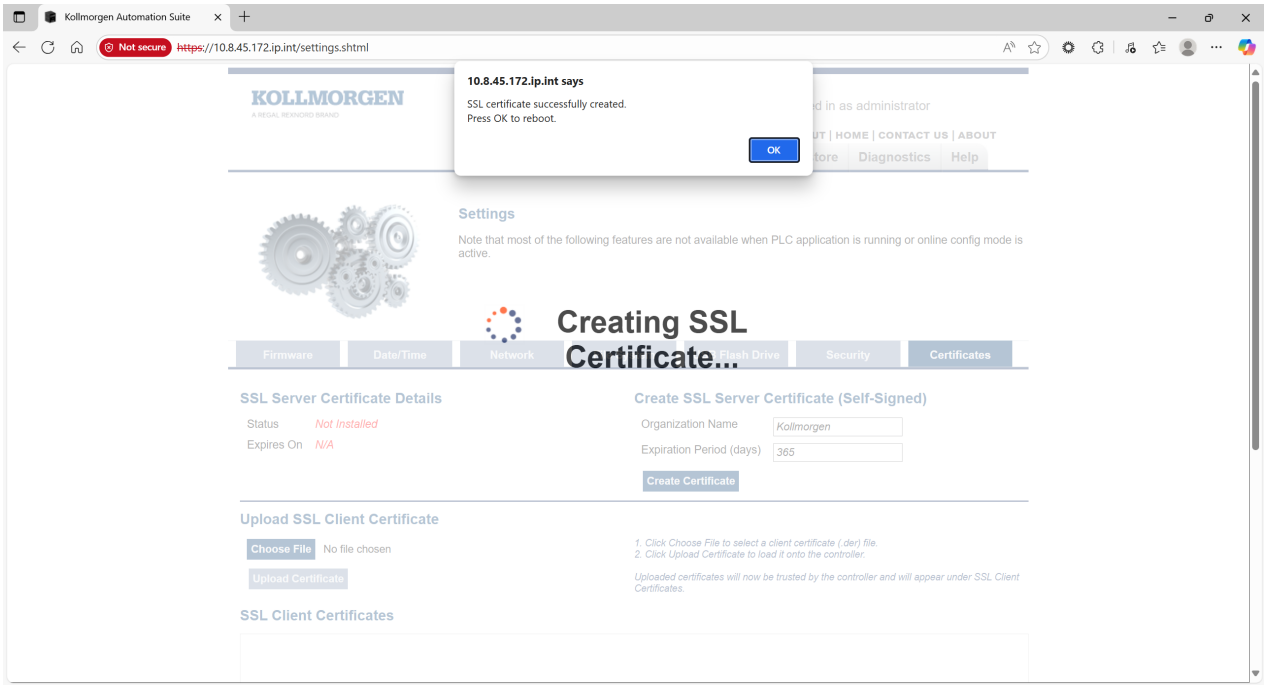


Figure 15-25: SSL certificate successfully created message

10. Click **OK** to reboot the controller.
    A rebooting message appears.
11. Wait for the **Disconnected** message to appear. (Figure 15-26)

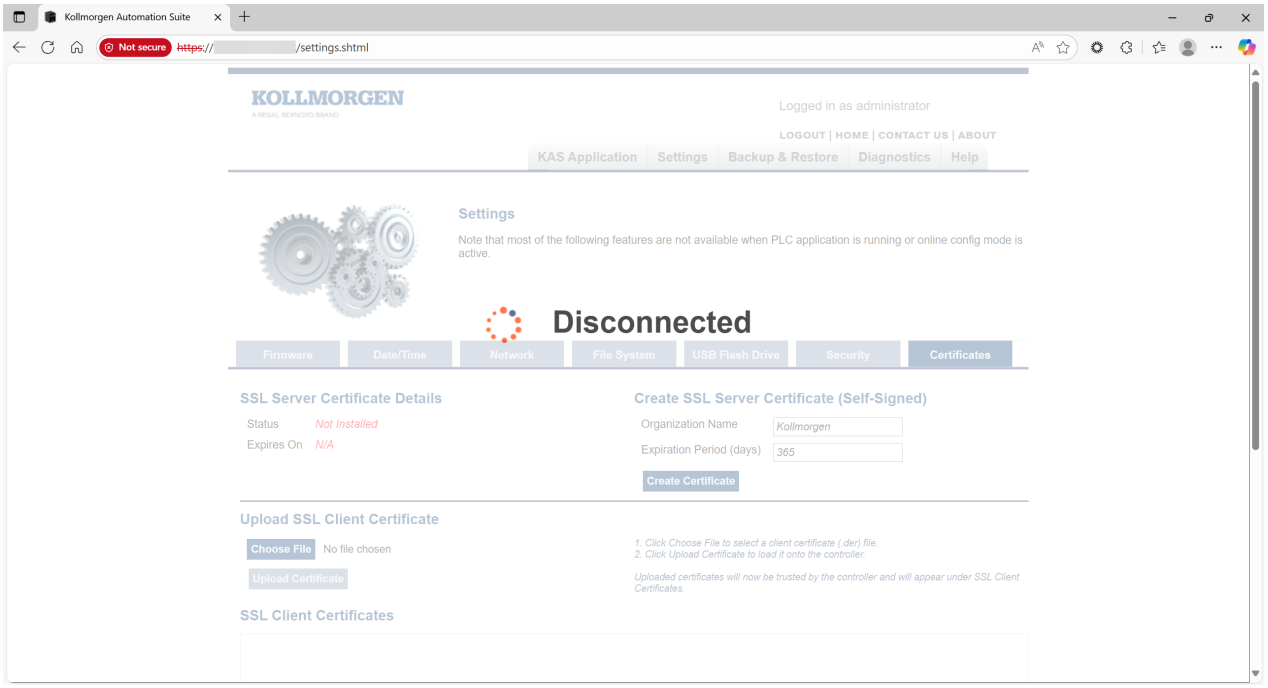


Figure 15-26: Disconnected message

12. Refresh the browser.
The connection message reappears. (Figure 15-27)

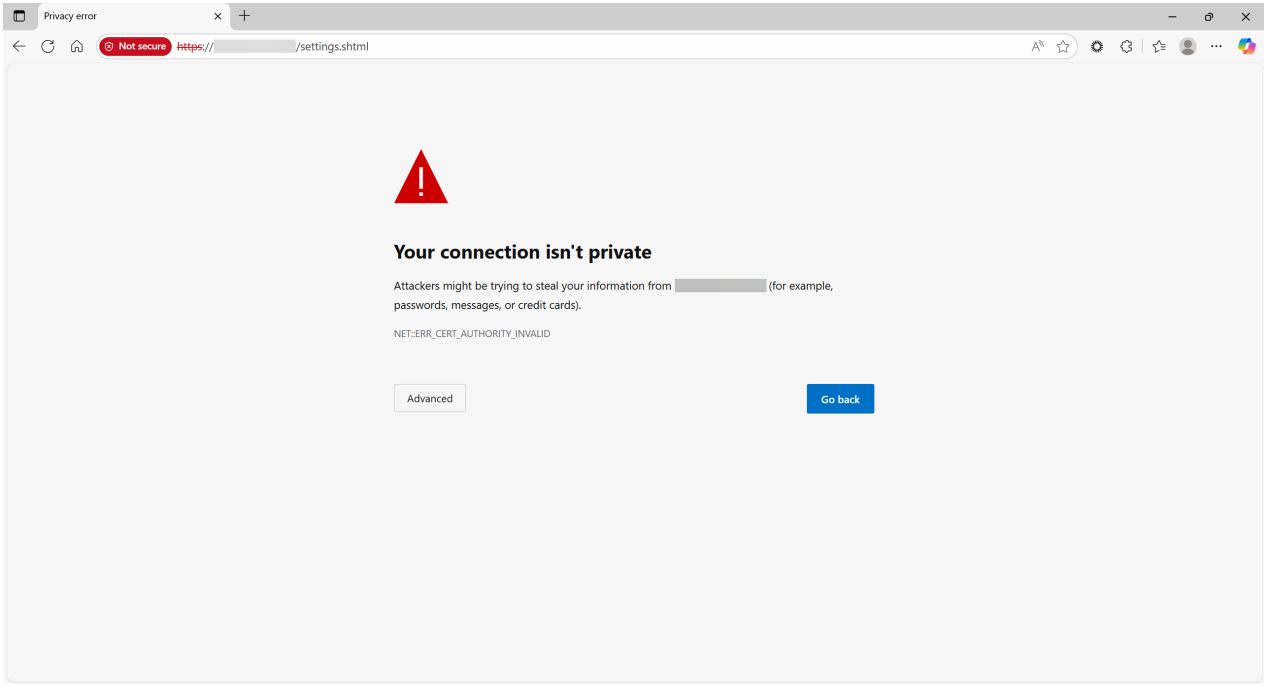


Figure 15-27: Connection message

13. Click the **Advanced** button again.
Additional information about the connection reappears. (Figure 15-28)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

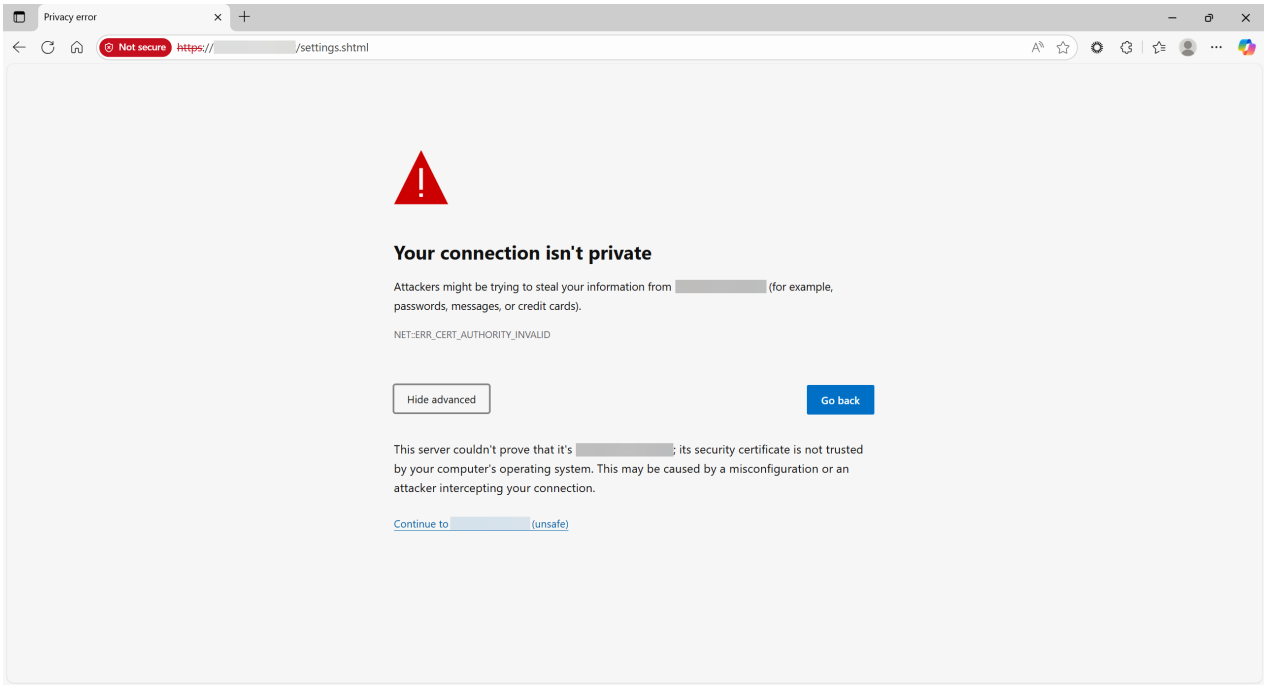


Figure 15-28: Connection message - Advanced information

14. Click the **Continue to (IP address) (unsafe)** link.
The Web server opens for the PCMM2G.
The PCMM2G shows it is **Not secure**. (Figure 15-29)

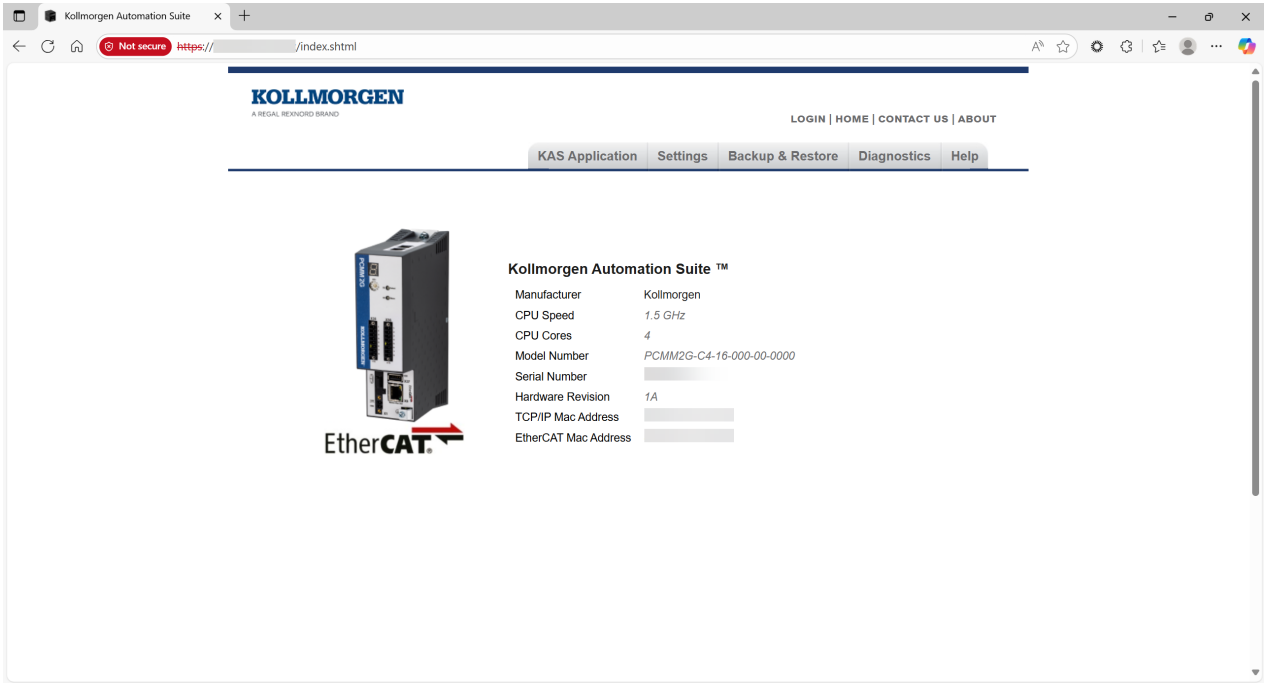


Figure 15-29: Not secured PCMM2G

15. Click the **Not secure** button and click the **Your connection to this site isn't secure** option. (Figure 15-30)

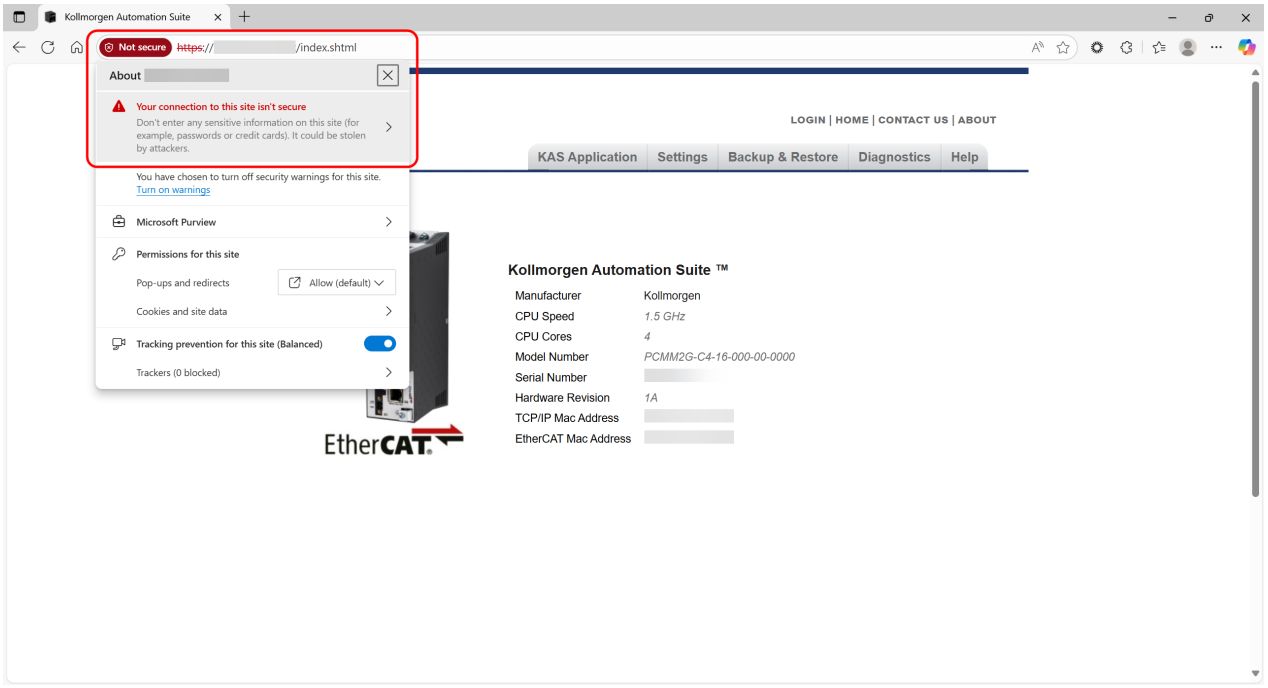


Figure 15-30: Your connection to this site isn't secure option

The certificate for this site is not valid message appears.

16. In the message header, click the **Show certificate** button. (Figure 15-31)

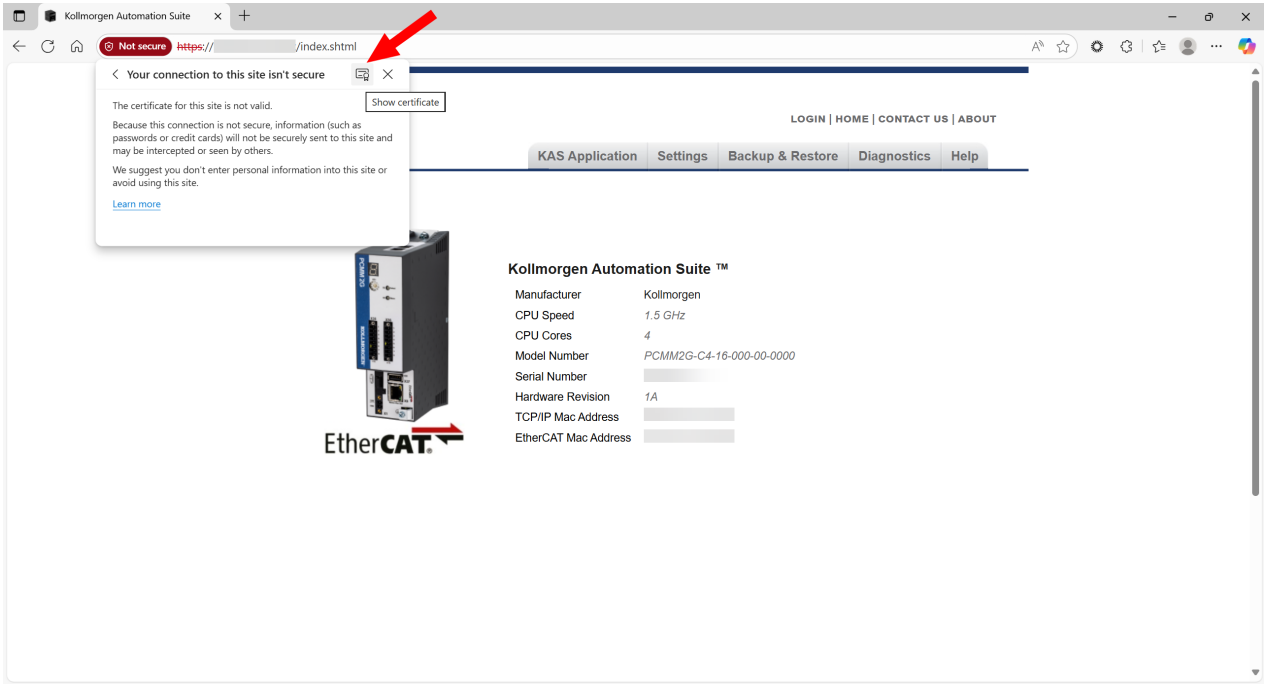


Figure 15-31: Show certificate button

The **Certificate Viewer** dialog opens.
The **General** tab is active. (Figure 15-32)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

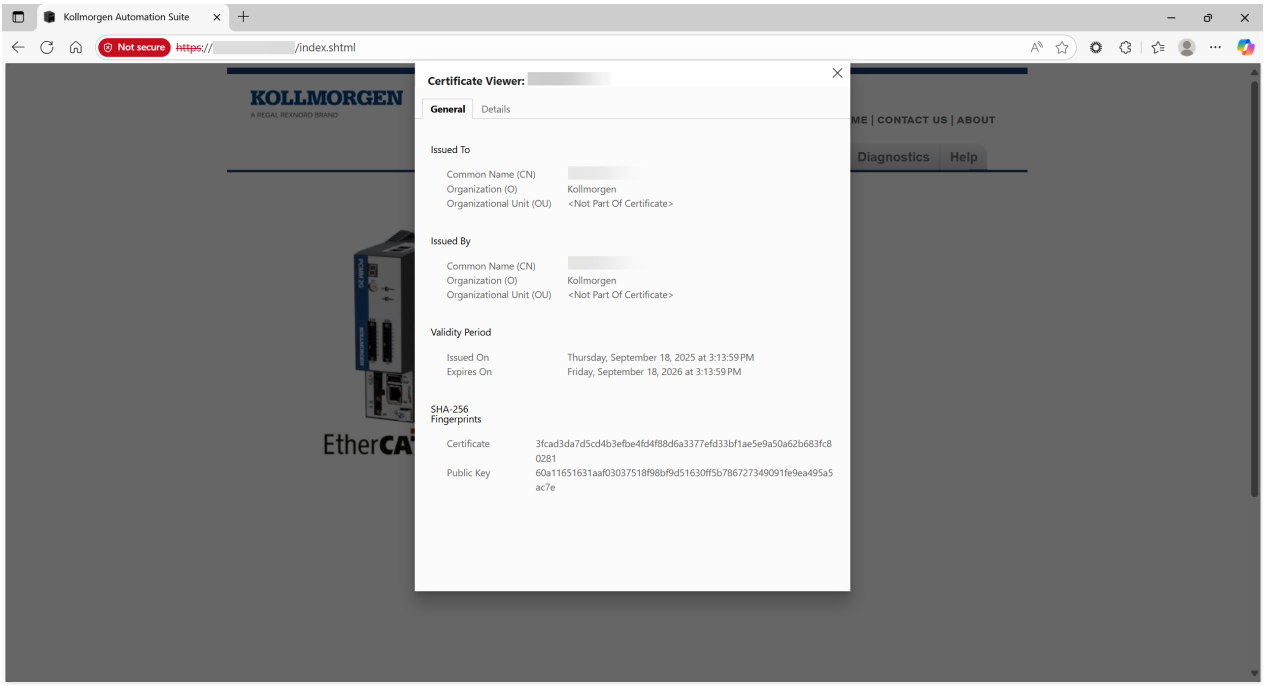


Figure 15-32: Certificate Viewer dialog - **General** tab

17. Click the **Details** tab. (Figure 15-33)

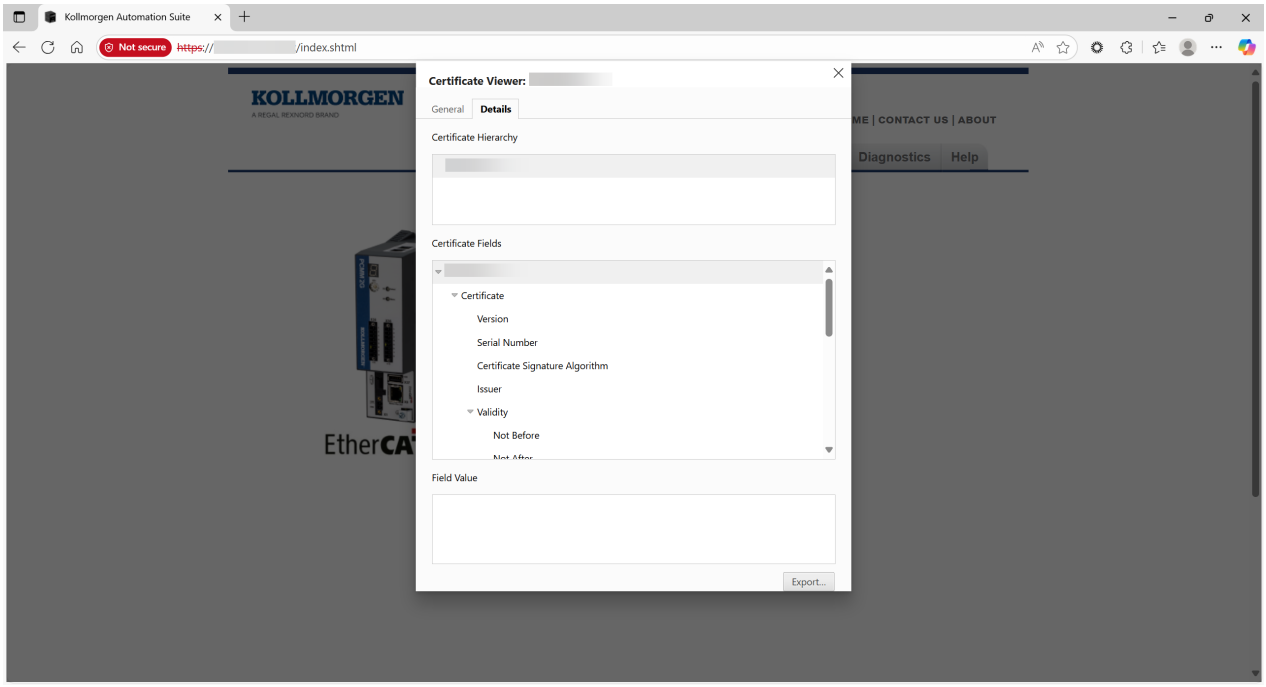


Figure 15-33: Details tab with **Export** button.

18. Click the **Export** button.
The **Save As** dialog opens. (Figure 15-34)

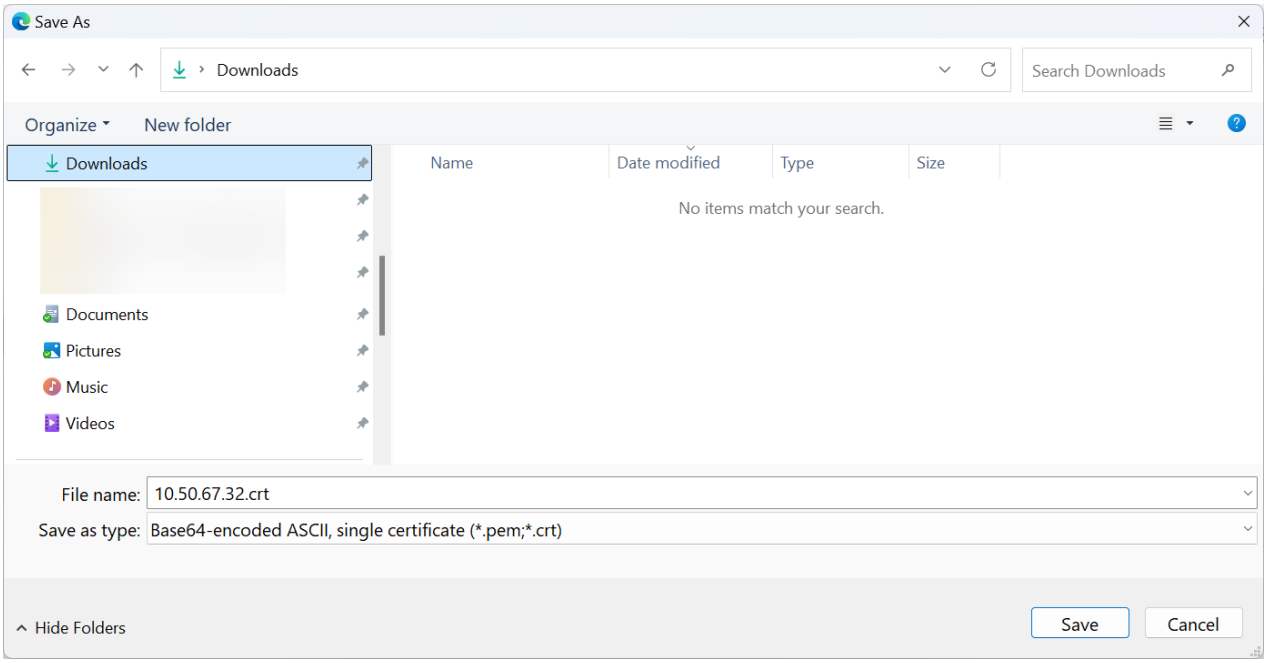


Figure 15-34: Save As dialog

19. Select a folder to save the **.crt** file.
    This example procedure uses the **Downloads** folder. (Figure 15-35)

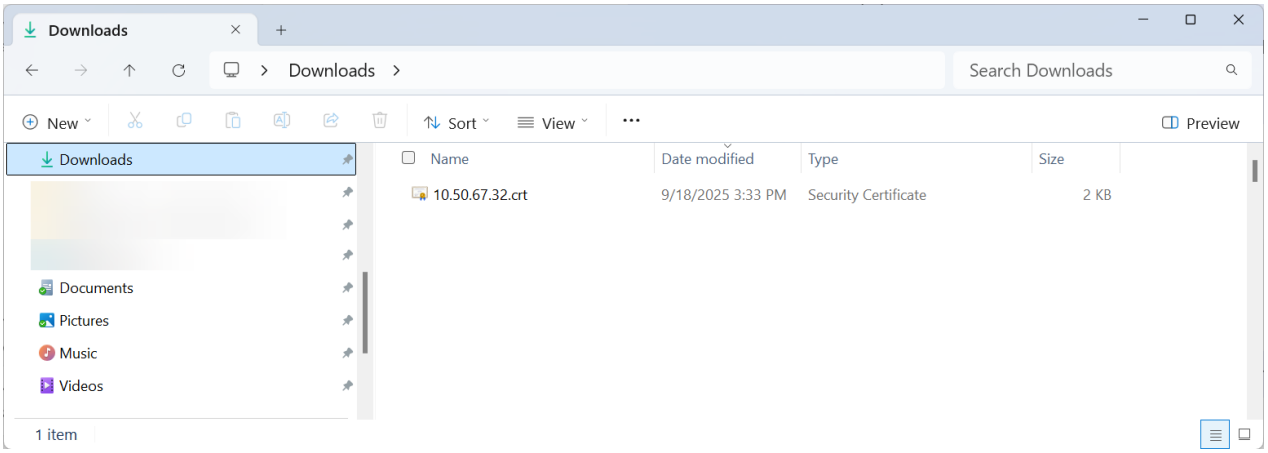


Figure 15-35: Save As dialog with saved **.crt** file

20. Click the **Save** button.
    The **Save As** dialog closes and the **Details** tab returns.
21. Use Windows Explorer to locate and select the **.crt** file.
    - The **.crt** file name is the PCMM2G's IP address.
      Example: 192.168.0.101.crt.
22. Right-click the file and click the **Install Certificate** option. (Figure 15-36)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

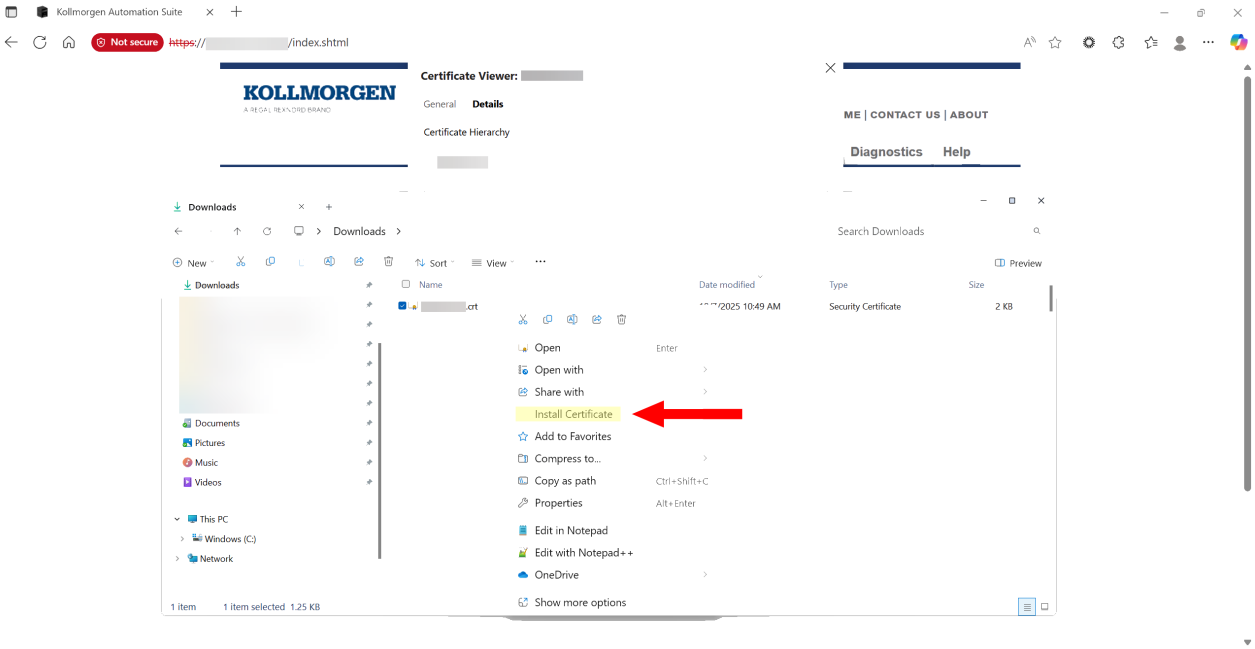


Figure 15-36: Selected **.crt** file and the **Install Certificate** option

The **Certificate Import Wizard** opens. (Figure 15-37)

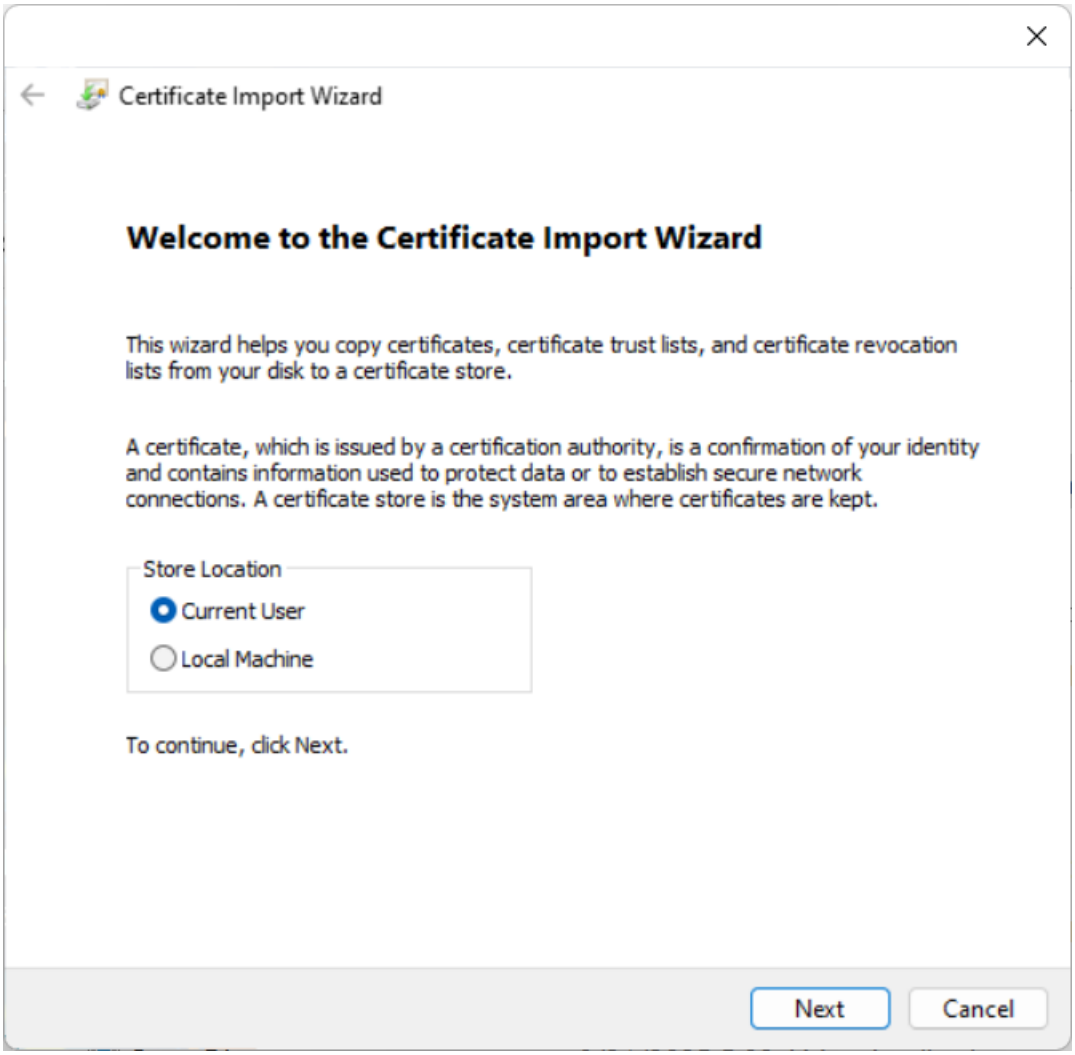


Figure 15-37: Certificate Import Wizard - Welcome page

23. Accept the default on the **Welcome** page and click **Next**.
    The **Certificate Store** page opens.
24. Select the **Place all certificates in the following store** option. (Figure 15-38)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

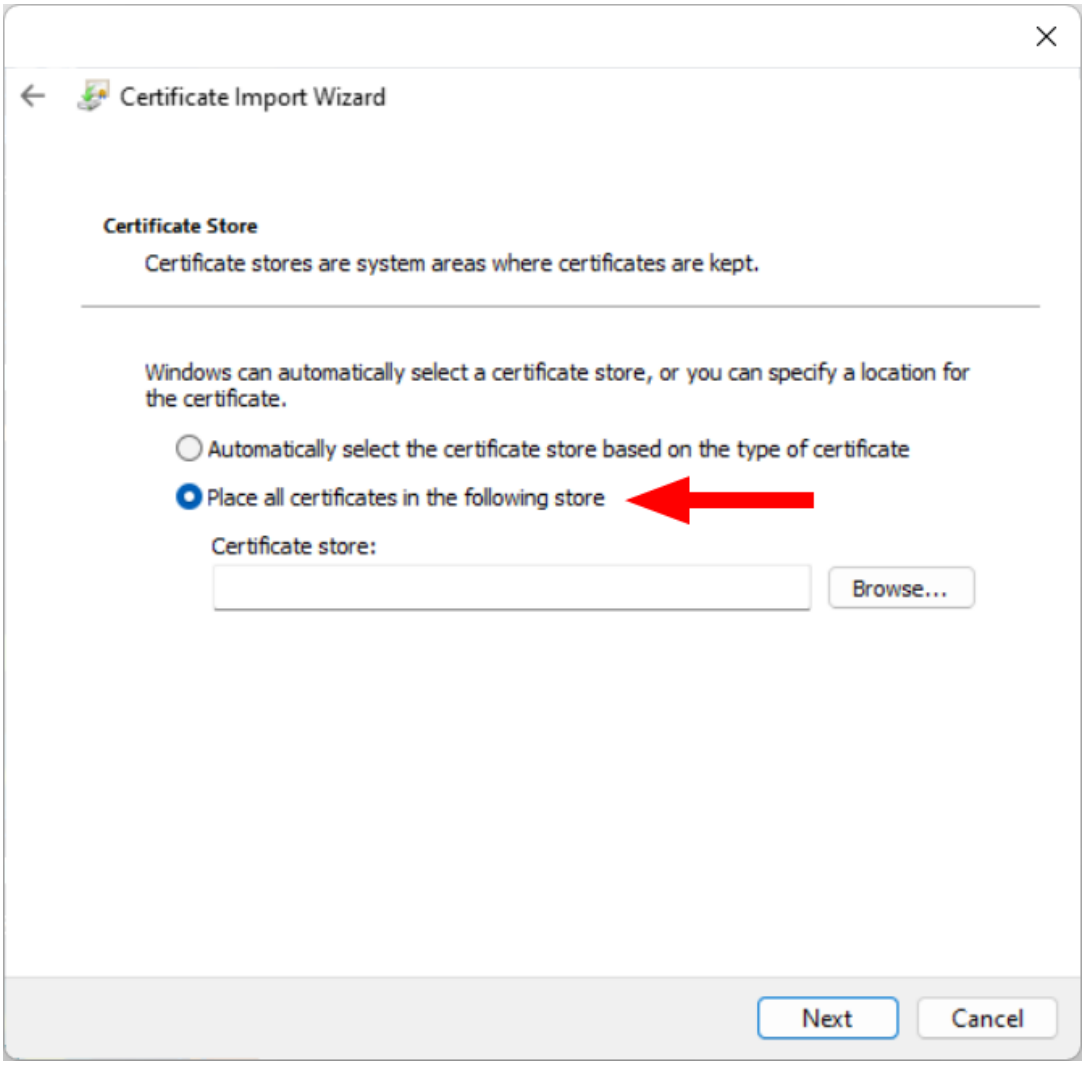


Figure 15-38: Certificate Import Wizard - Certificate Store page with selection

25. Click the **Browse...** button.
    The **Select Certificate Store** dialog opens.
26. Select the **Trusted Root Certification Authorities** certificate store. (Figure 15-39)

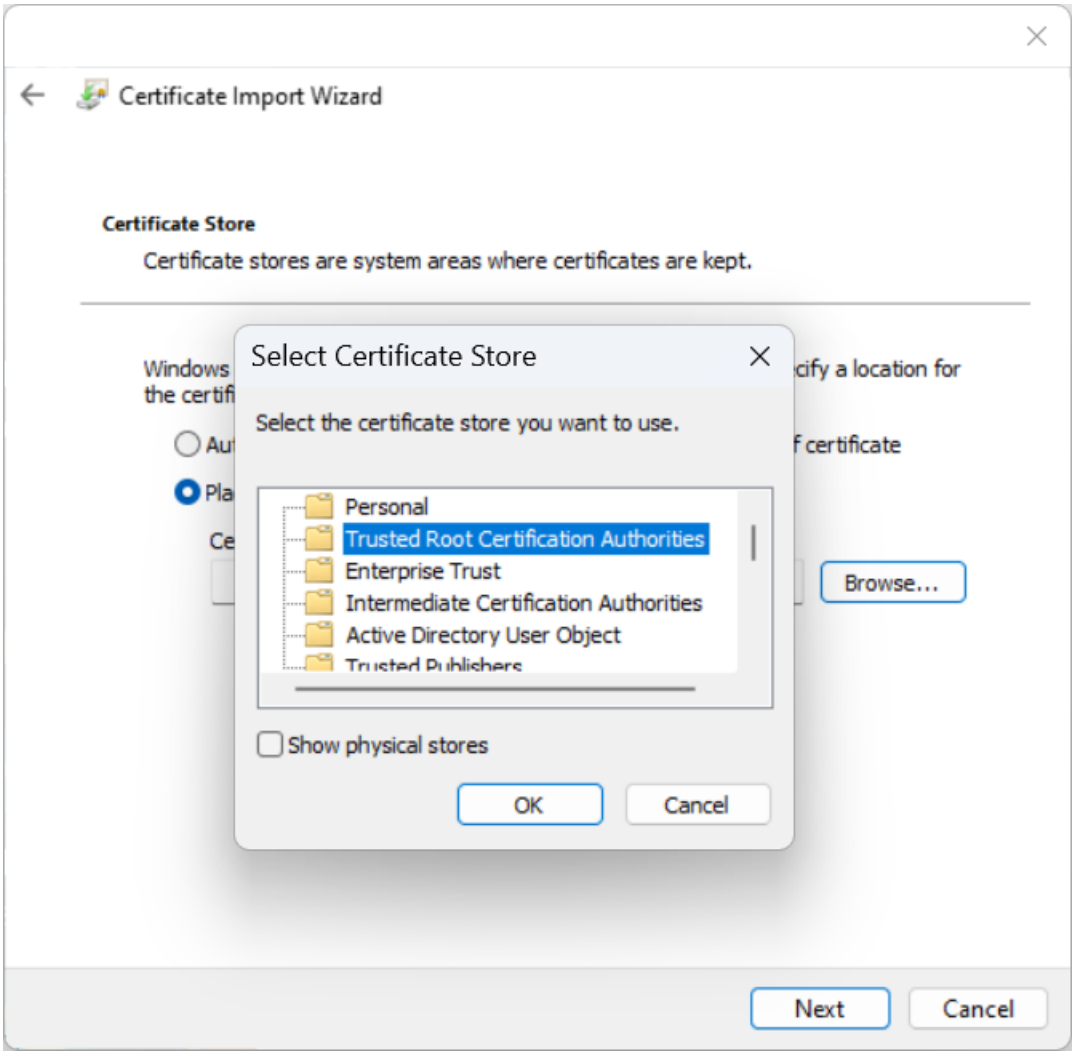


Figure 15-39: Selected Certificate Store dialog - **Trusted Root Certification Authorities**

27. Click **OK** to save the changes or selections and close the dialog box.
The **Certificate Store** page returns and shows the selected **Certificate store**. (Figure 15-40)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

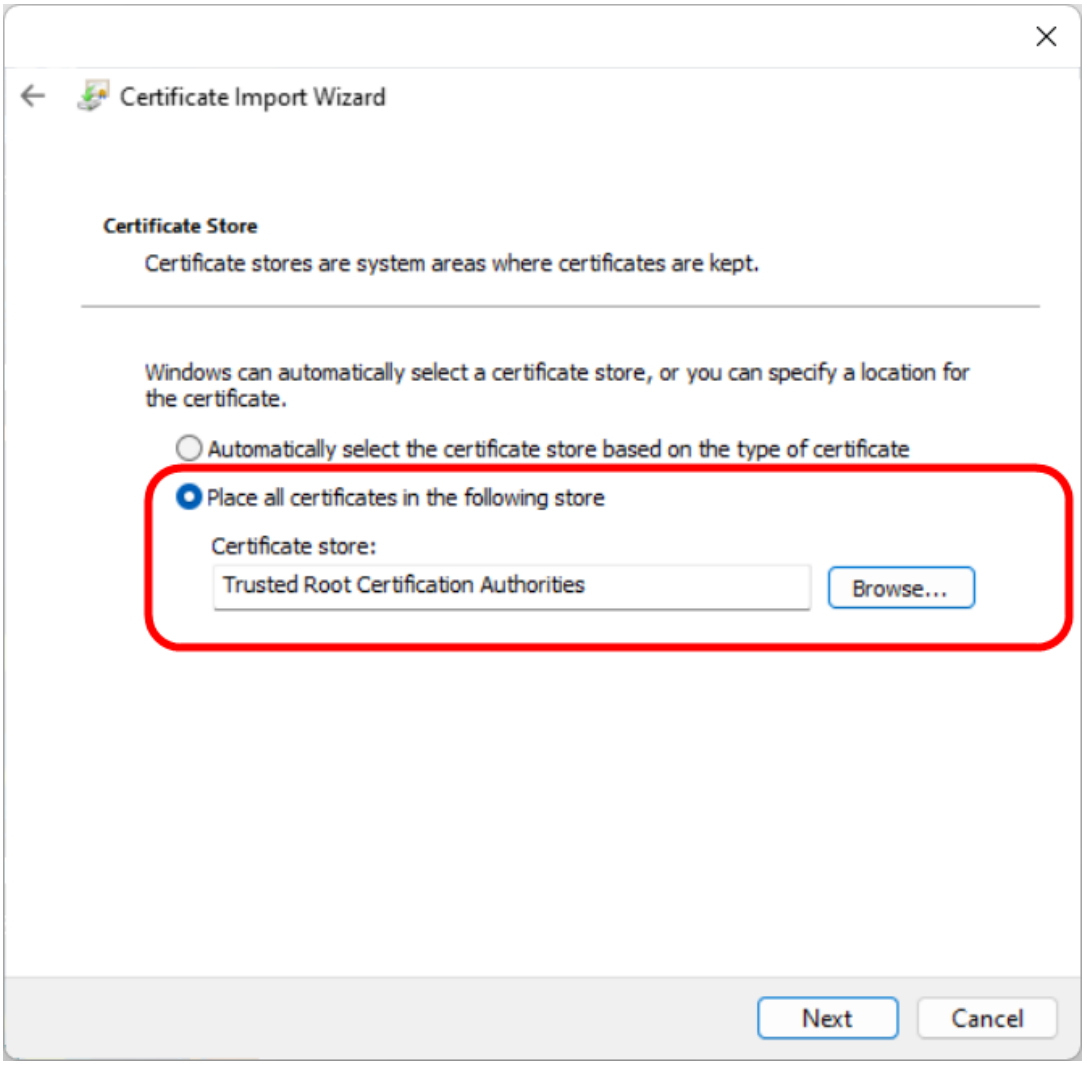


Figure 15-40: Certificate Import Wizard - Certificate Store page with selected **Certificate store**

28. Click **Next**.
The **Completing the Certificate Import Wizard** page opens. (Figure 15-41)

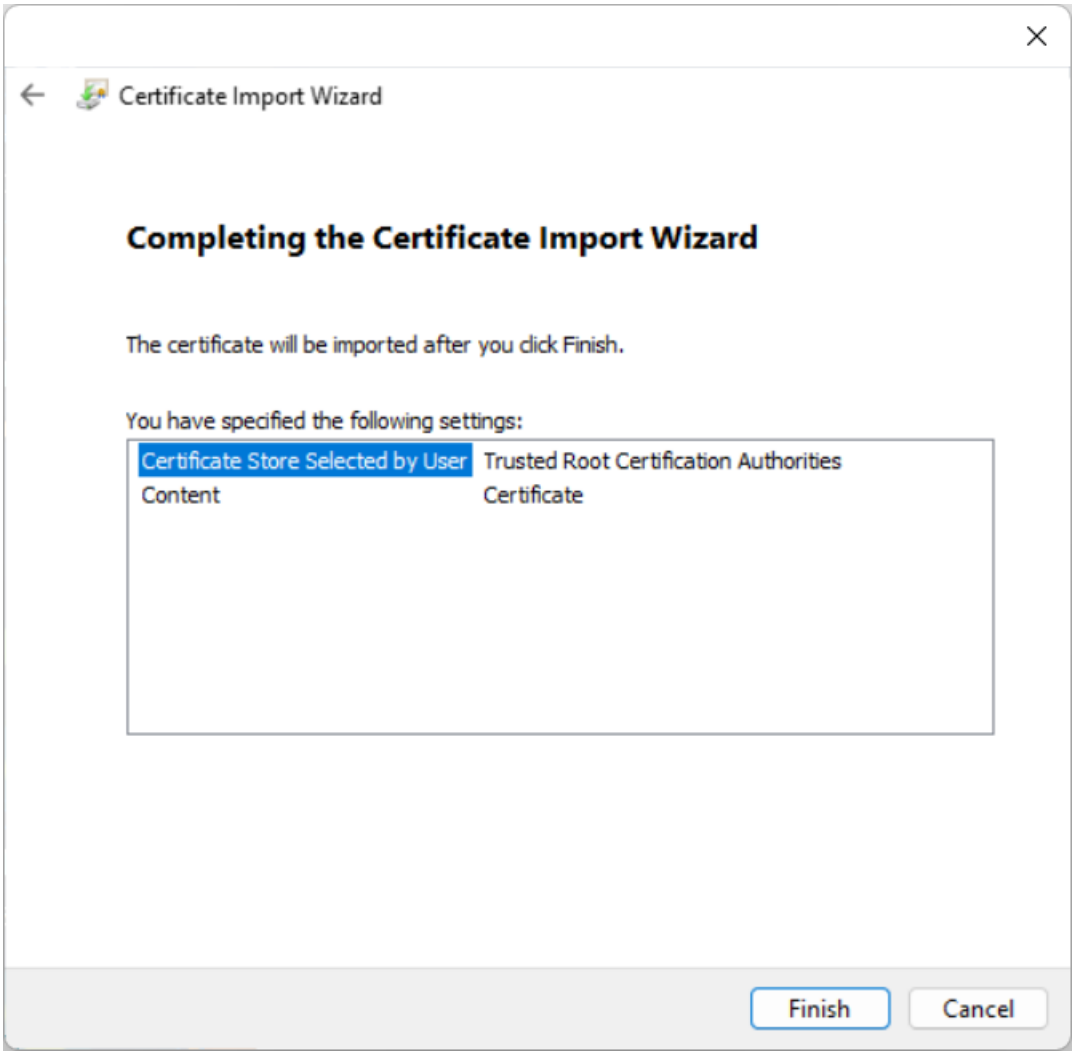


Figure 15-41: Certificate Import Wizard - Completing the Certificate Import Wizard page

29. Click **Finish**.
A **Security Warning** dialog opens. (Figure 15-42)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

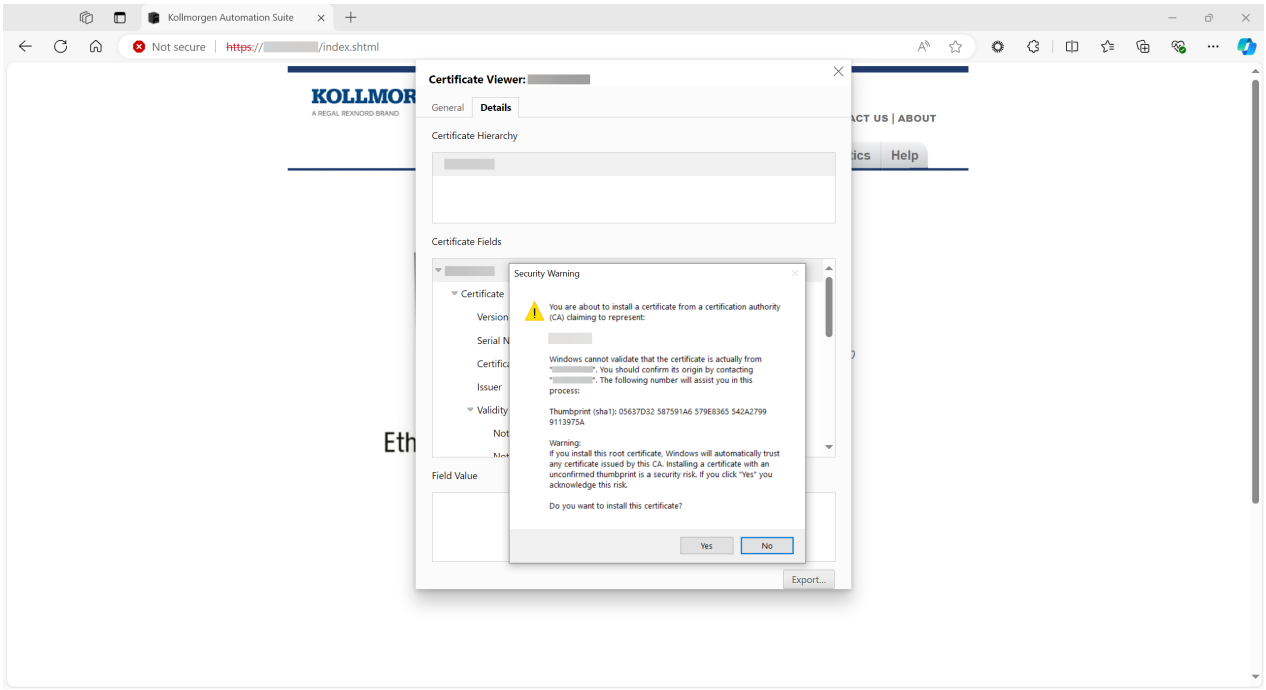


Figure 15-42: Security Warning dialog

30. Click **Yes** to install this certificate.
The **Certificate Import Wizard - The import was successful message** appears. (Figure 15-43)

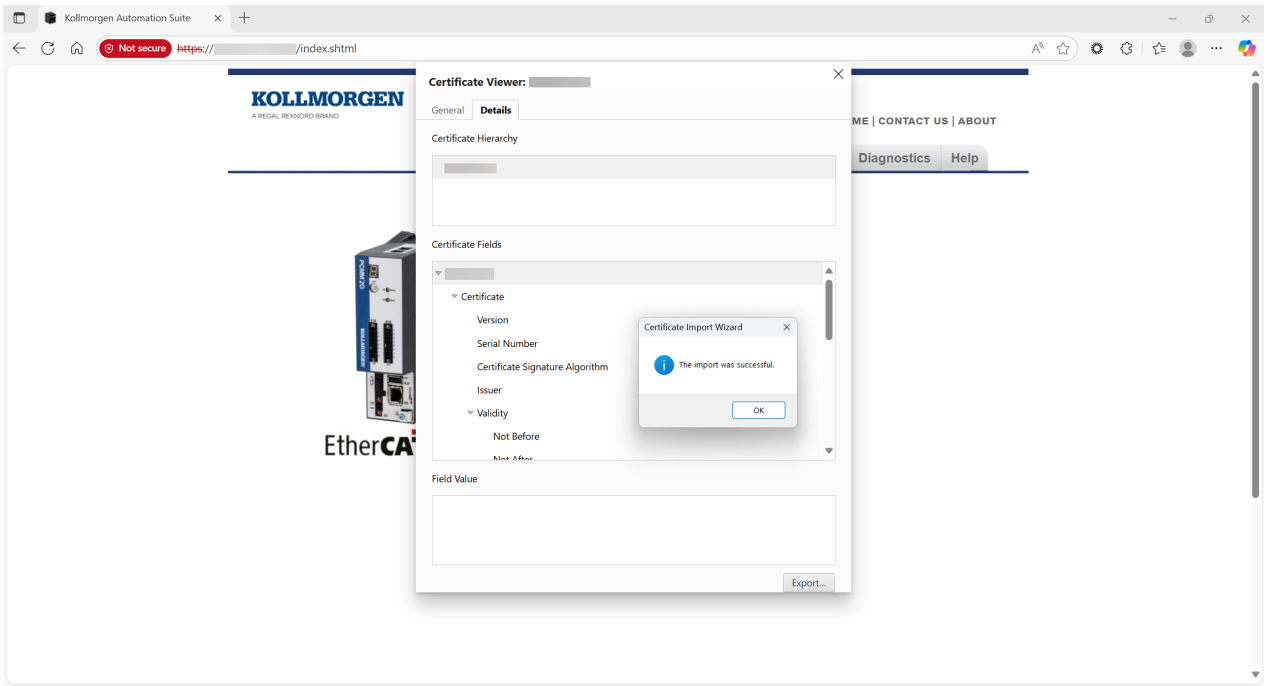


Figure 15-43: Certificate Import Wizard - The import was successful message

31. Click **OK** to continue.
The **Certificate Import Wizard - Details** tab returns.
32. Close the wizard.
The connected PCMM2G Web server page returns.
33. Close the web browser.
34. **Verify ALL web browser windows are closed.**
35. Open a new web browser instance and use the URL to connect to the Web server.
The PCMM2G shows it is secure.

36. Click the **Secure** button. (Figure 15-44)

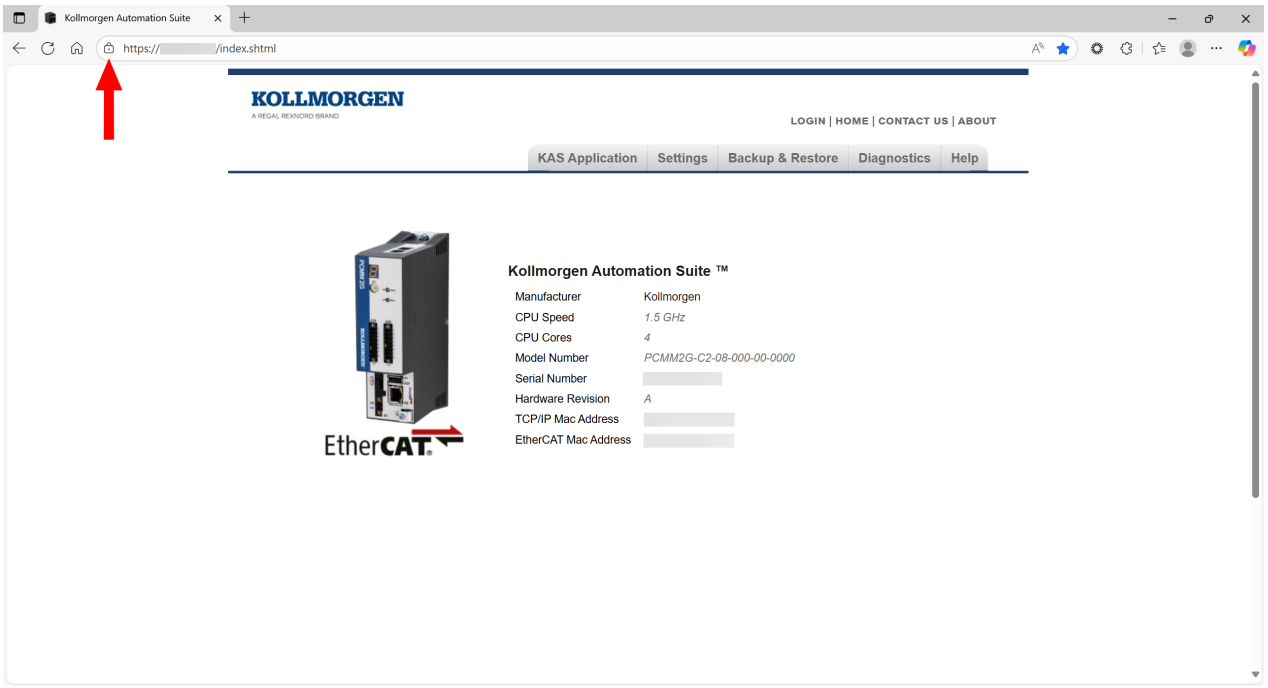


Figure 15-44: Secure PCMM2G

37. Click the **Connection is secure** option. (Figure 15-45)

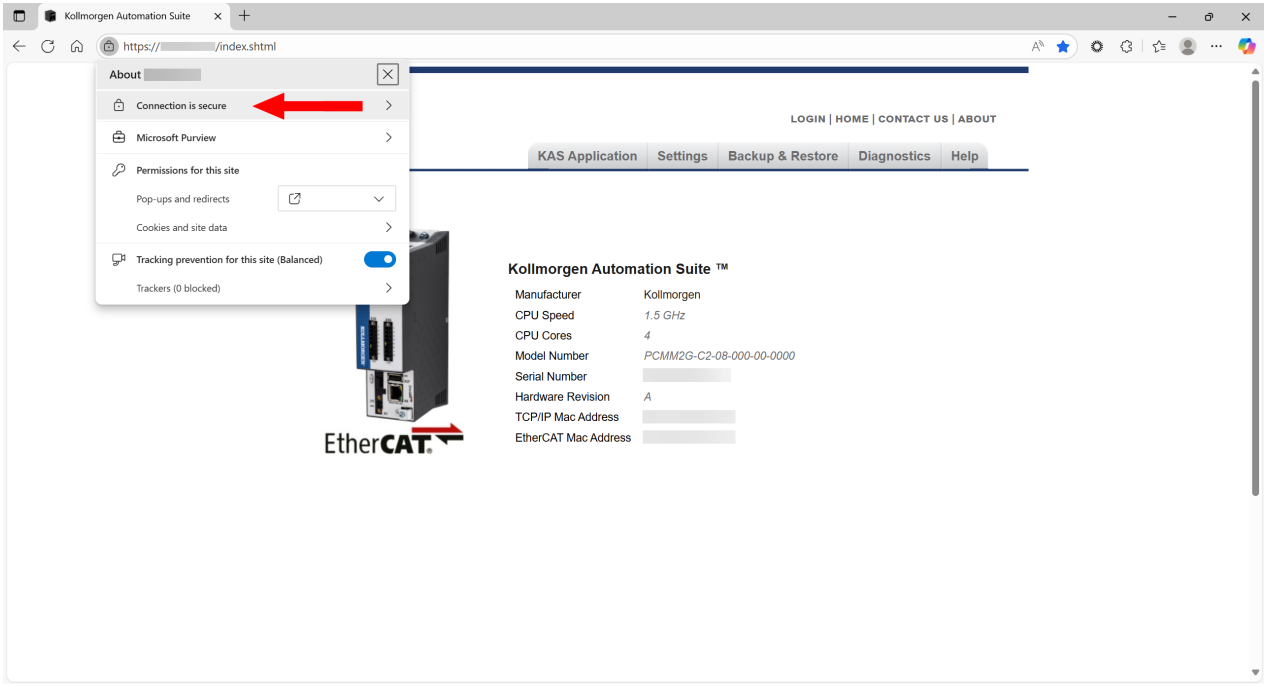


Figure 15-45: Connection is secure option

The **Certificate is secure** message appears. (Figure 15-46)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

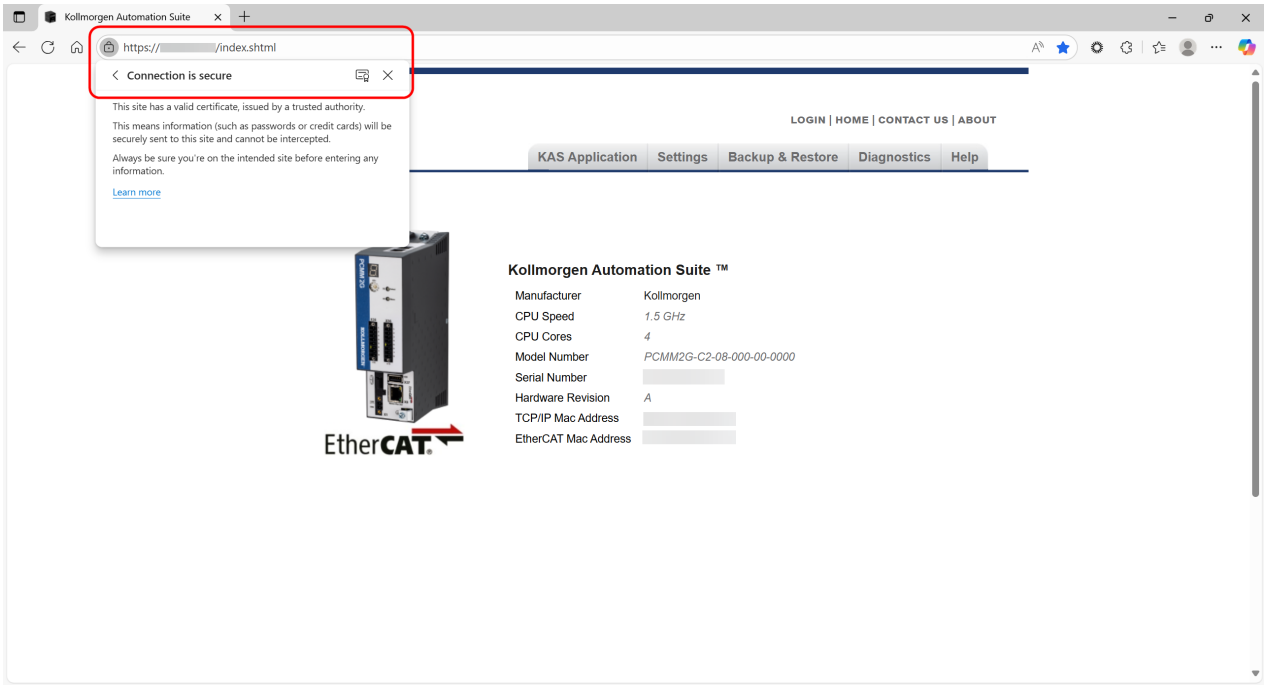


Figure 15-46: Certificate is secure message

38. Click the **Settings** tab.
39. Click the **Certificates** tab.
The **SSL Server Certificate Details** area shows the **Status** and expiration date. (Figure 15-47)

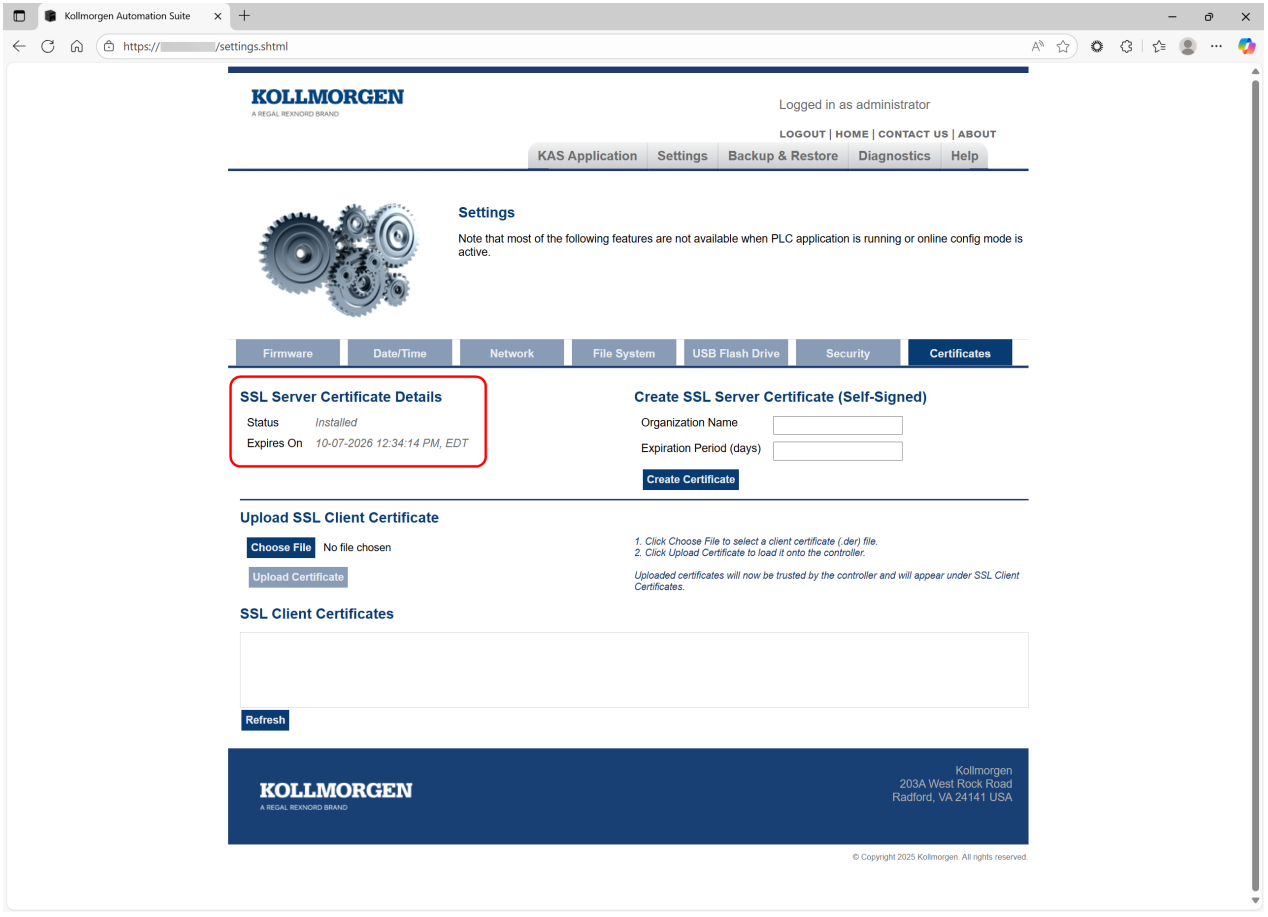


Figure 15-47: SSL Server Certificate Details area with certificate information

11.10.7 Troubleshooting

The webpage displays a warning:

- **KAS IDE** warning: The identity of this web site or the integrity of this connection cannot be verified.
- **Microsoft Edge / Google Chrome** warning: Your connection isn't private.
- **Mozilla Firefox** warning: Warning: Potential Security Risk Ahead.

Remedies

- Verify the current certificate:
  - has been downloaded and installed onto the machine.
  - matches the IP address of the controller.
  - has not expired yet

The IP address of the PCMM2G does not match the IP address listed on the SSL certificate.

Remedies

- Create, download, and install a new certificate with the new IP address.
- Configure the IP address with a static IP address matching the certificate's IP address using the rotary switch.
  - See "Network tab" (➔ p. 675).
- Reserve the certificate's IP address for the PCMM2G using a router or server.

11.10.7.1 Date/Time tab

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

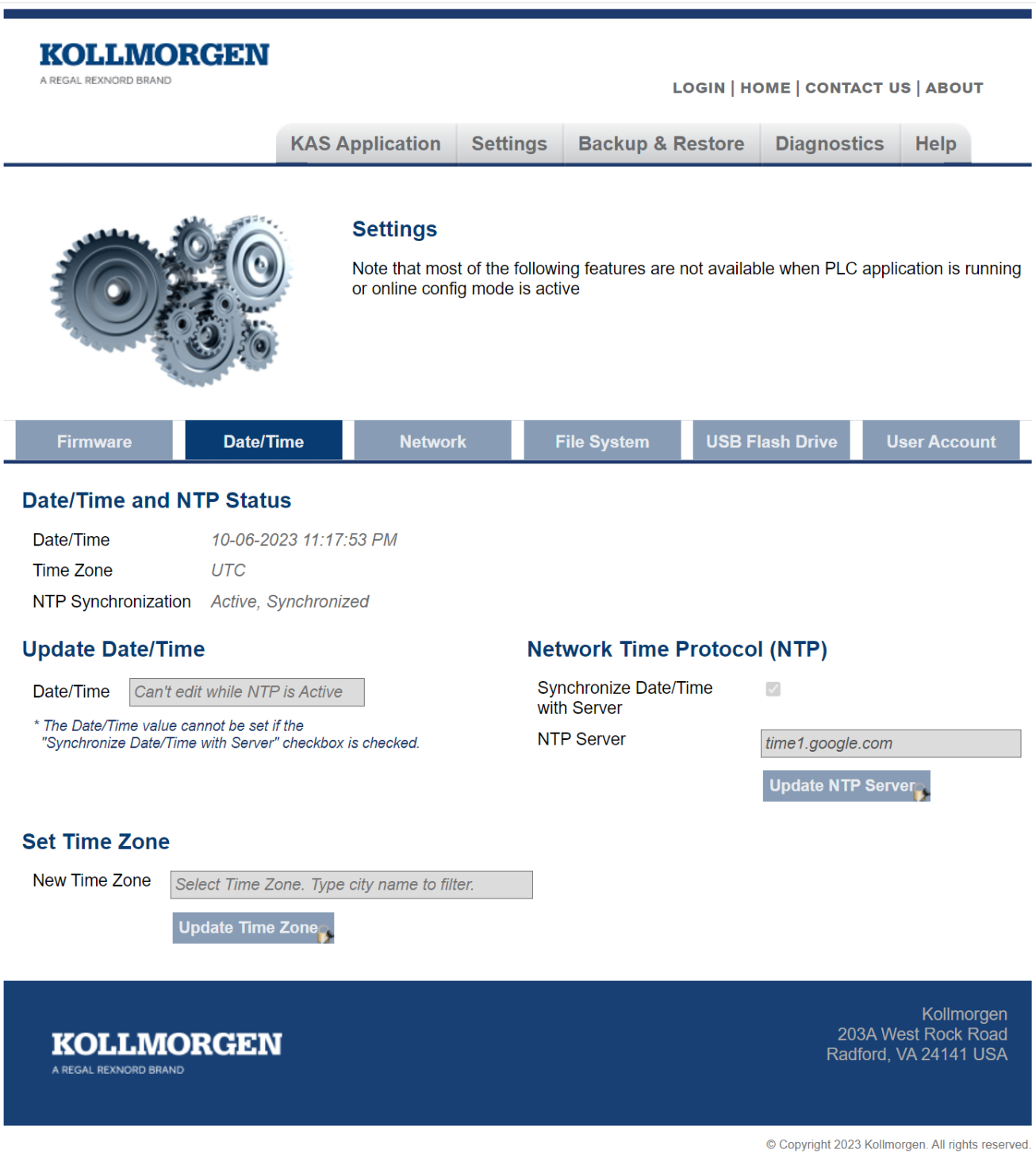


Figure 15-48: Date/Time tab with Kollmorgen Manually Defined Time Zones

This tab is used to view or change the current date, time, and Network Time Protocol (NTP) synchronization information.

- "Date/Time and NTP Status" (→ p. 664)
- "Update Date/Time" (→ p. 664)
- "Network Time Protocol (NTP)" (→ p. 665)
- "Set Time Zone" (→ p. 665)

The controller's time zone, NTP server address, NTP synchronization configuration, and the date and time can be manually changed.

11.10.7.1.1 Date/Time and NTP Status

The information in this area updates every 30 seconds or when a setting is changed using the Date/Time tab.

This area is read-only.

Information Field	Description
Date/Time	This is the local date and time as reported by the controller.
Time Zone	This is the time zone used by the controller.
NTP Synchronization	This is the synchronization status for the NTP service. The synchronization is defined in two parts. • The first part of the status reports: • **Active** if the controller is configured to synchronize its time with an NTP server. • **Disabled** if the controller is **not** configured to synchronize its time with an NTP server. • The second part of the status reports: • **Synchronized** if the controller is synchronized to an NTP server. • **Not Synchronized** if the controller is **not** synchronized to an NTP server. **NOTE** The controller stays synchronized for several minutes after disabling NTP synchronization. A **Disabled, Synchronized** status is possible.

11.10.7.1.2 Update Date/Time

Click in the **Date/Time** field to open the **Date/Time** picker.

Select a new date and time and click **OK**.

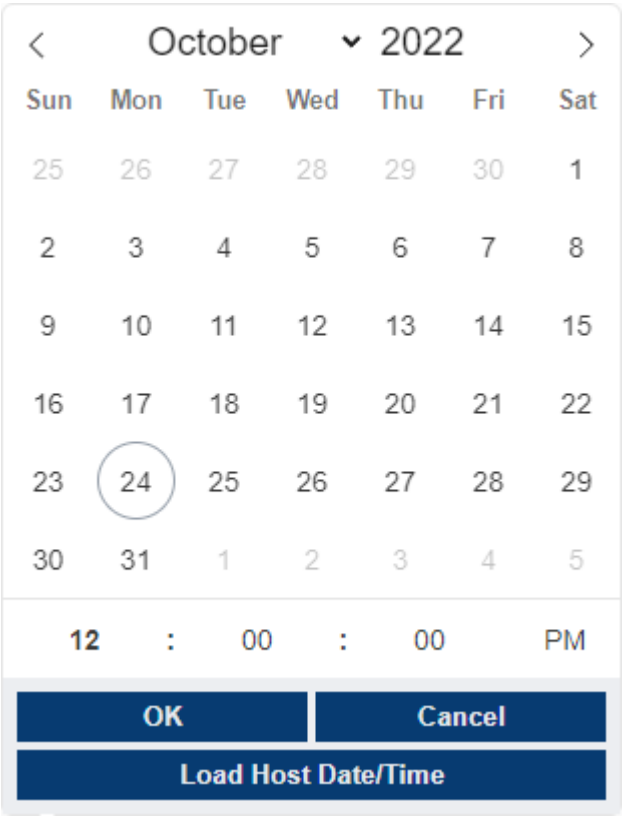


Figure 15-49: Date/Time Picker

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Button	Description
OK	Click to load the date and time selected in the **Date/Time** picker onto the controller.
Cancel	Click to close the **Date/Time** picker without changing the date/time on the controller.
Load Host Date/Time	Click to copy the date/time from the host (i.e., computer or device running the web browser) onto the controller.

NOTE

The **Date/Time** picker is disabled if the controller is configured to synchronize the date/time with an NTP server or if not logged into the controller.

11.10.7.1.3 Network Time Protocol (NTP)

This area is used to update the NTP settings.

Setting	Description
Synchronize Date/Time with Server	Changing the state of the checkbox immediately changes the configuration on the controller. • A checked box indicates that an NTP server is used to update the date and time. • An unchecked box indicates the controller does not synchronize its time or date with an external server.
NTP Server	This is the address of an NTP server to use for synchronizing the date and time. • If the controller does not have access to the internet, it is still possible to connect to a local NTP server if the local intranet has an NTP server configured. • A local NTP server can be useful for manufacturing areas that do not want to connect controllers and other equipment to the internet for security reasons but still want to keep all computer's and controller's date and time synchronized for data collection purposes.
Update NTP Server button	• Click the **Update NTP** button to apply the new NTP server address. • The **Update NTP** button is disabled if not logged into the controller.

11.10.7.1.4.1 Correct an A05 Alarm

NOTE

If NTP is disabled, and the battery is either not installed or dead on power-up, then the system time is wrong and an "A05" (→ p. 583) alarm is generated.

To prevent an A05 alarm on startup, either:

- Enable NTP (see "Synchronize Date/Time with Server" (→ p. 665)).
- Follow this procedure:

1. Remove power from the controller.
2. Insert a new battery.
3. Power on the controller.
   There will likely still be an A05 alarm.
4. Use the "Update Date/Time" (→ p. 664) procedure to set the system time.
5. Power off the controller and wait at least 10 seconds before powering it back on.
6. Verify:

   - The A05 alarm does not occur.
   - The time is correct on the web page.

11.10.7.1.5 Set Time Zone

This area is used to update the time zone used by the controller.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

To set the controller's time zone, select a new time zone from the drop-down box and click the **Update Time Zone** button.

- Time zones are often specified by the largest city in that time zone.
- **Example**: The US Eastern Time Zone is specified by **America/New_York**.
  Typing the first two or three characters of a city or region name filters the list to easily locate a specific time zone.
  For New York, typing **ne** reduces the time zone list to two entries.

NOTE

The **Update Time Zone** button is disabled if not logged into the controller using the Web server.

11.10.7.2 File System tab

This tab is used to reset the controller to the factory settings.

NOTE

The steps to reset the controller vary slightly based on the platform.



Figure 15-50: File System tab on a Kollmorgen controller's web server and when using Simulator.

Webpage Notes

- This webpage does not update during the reset procedure and can be closed.
- The **Reset to Factory Settings** button is disabled while an application is running.

11.10.7.2.1 Reset the Controller to Factory Settings

⚠WARNING

Performing a factory reset of the controller should be considered very carefully.
The reset erases everything: User data files, the application, the controller IP address, any retain variables, passwords, shared directory settings, and all log files.

When the **Reset to Factory Settings** button is clicked, the controller is reset to factory default settings.

A confirmation message opens to confirm this action before the function is performed.

11.10.7.2.1.1 Factory Reset Changes

The changes which occur during a reset are:

Erased Items

- User data files.
- Any application previously downloaded.
- Retained variables.
- Shared directory settings.
- Log files.

Defaults Restored

- Password for both the web page and SSH login.
  - See "SSH Login to a Controller" (→ p. 617) and "User Authentication" (→ p. 623).
- Controller network configuration.
  - Examples: IP address, Subnet, and Gateway settings.
- Auto-Start option.

Security

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The SSH keys are regenerated.
  - SSH clients logged on during the factory reset are forced to log in again.
  - All Date and Time settings are reset to default (NTP enabled).

Notes

- The factory reset cannot be performed while an application is running.
- Do not turn off the controller during the reset procedure.
- Estimated time for factory reset:
  - PCMM2G: About a minute.
  - The 7-segment display on the controller is animated during this process.
- After the factory reset is complete, the controller is powered down and restarted automatically.
- After the controller is restarted, the controller IP address may change based on the controls rotary switch.
  - If the rotary switch is at position 0 (zero), the same IP address as before should be assigned to the controller.
  - If the rotary switch is set to 1-9, a pre-configured IP address is defined.
    - This IP address must be taken into account when trying to reconnect to the webpage using a web browser.

11.10.7.3 Firmware tab

The **Firmware** tab is used to:

- "View Firmware Information" (→ p. 671)
- "Upgrade the Controller's Firmware" (→ p. 668)

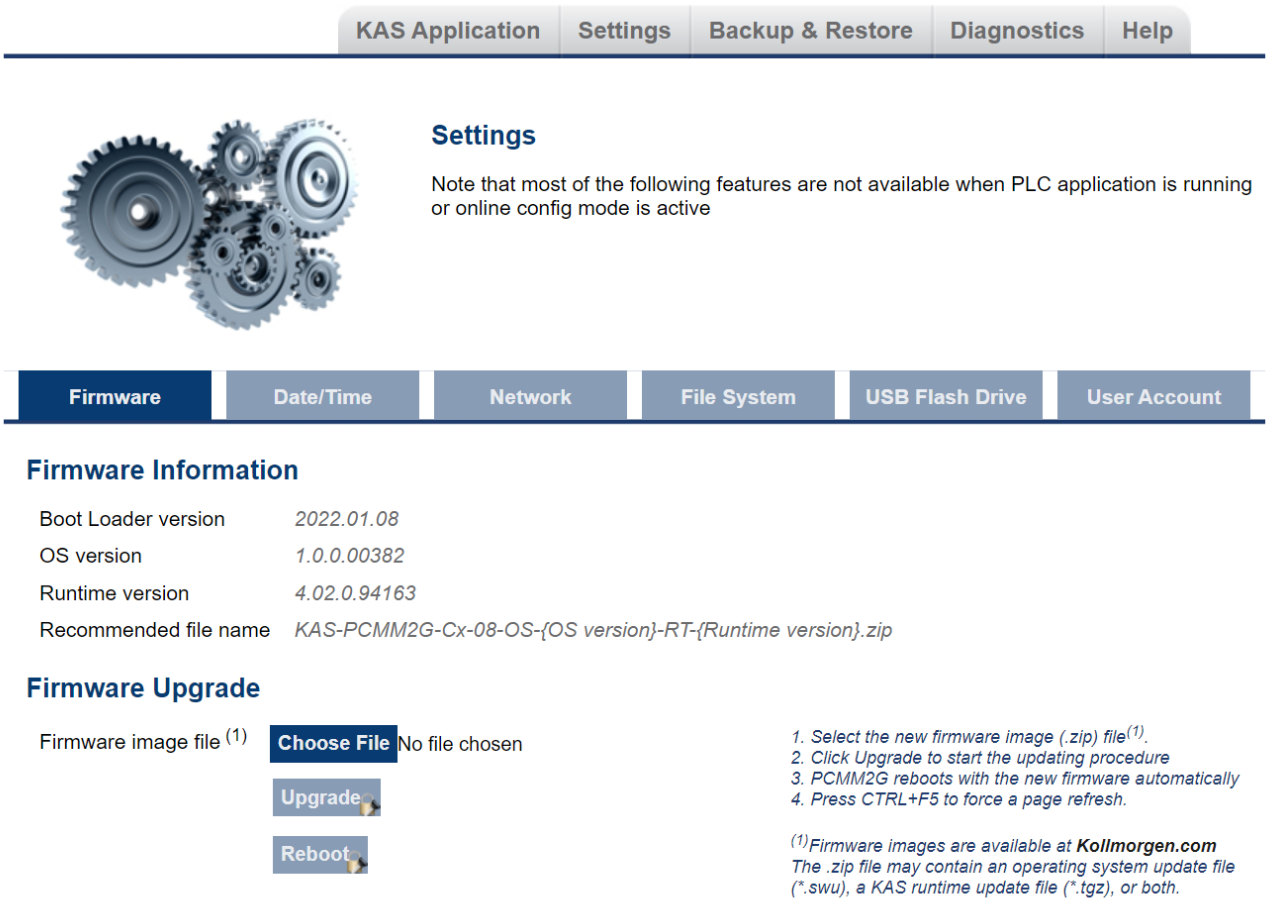


Figure 15-51: Firmware tab

See Also

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Booting the AKD PDMM or PCMM Controller" (➔ p. 593)
- "Booting the PCMM2G Controller" (➔ p. 593)
- "Recover Using the Web server" (➔ p. 672)

11.10.7.3.1 Upgrade the Controller's Firmware

This procedure is used to upgrade the controller's firmware using the web server.

1. Identify the controller's IP address.
   See "View the Controller IP Address" (➔ p. 602).
2. Open a web browser and enter the controller's IP address.
   The Web server KAS Home window opens. (Figure 15-52)

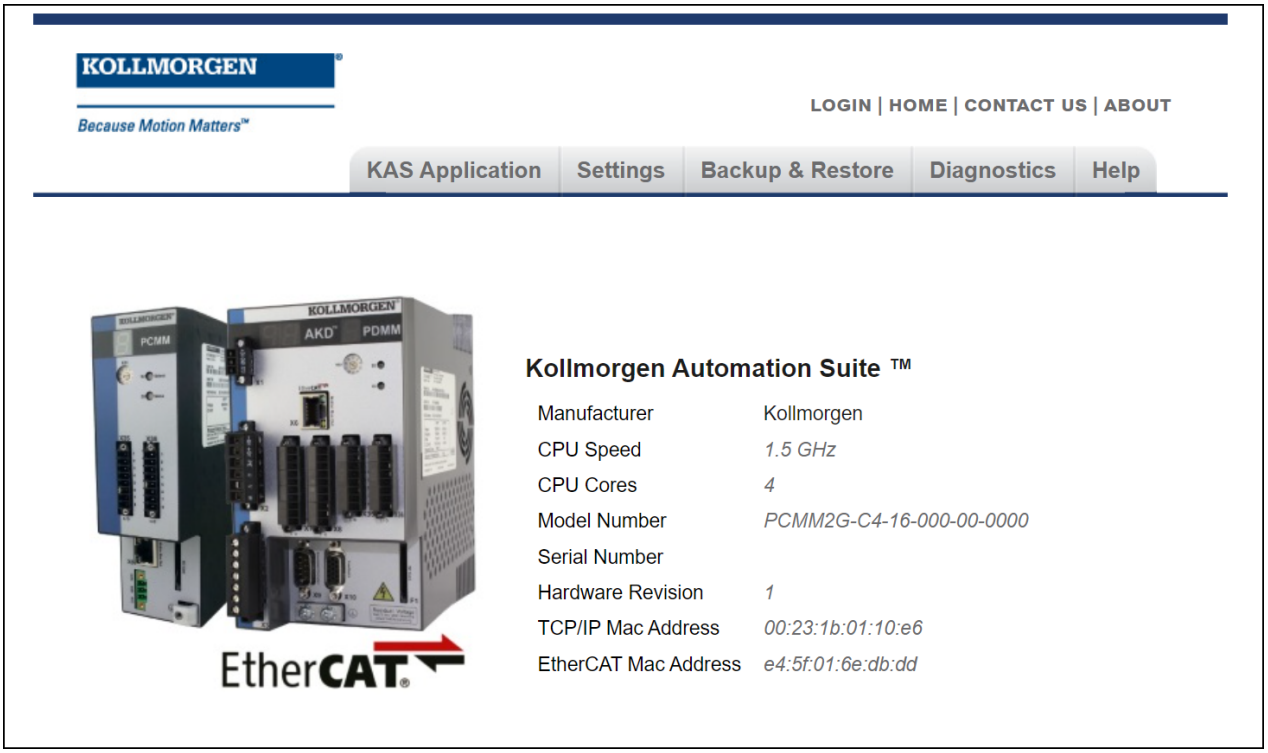


Figure 15-52: KAS Home window

3. Login to the controller.
4. Click the **Settings** tab.
5. Click the **Firmware** tab.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

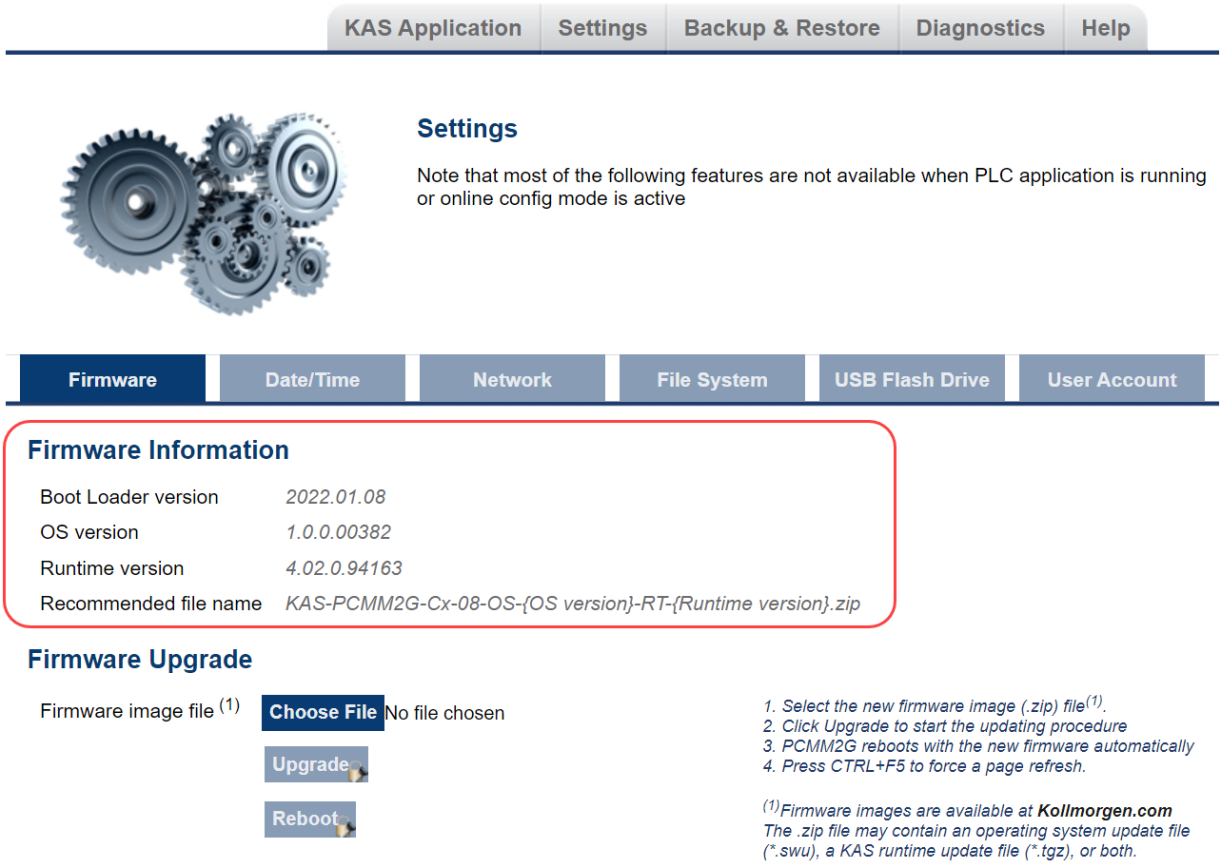
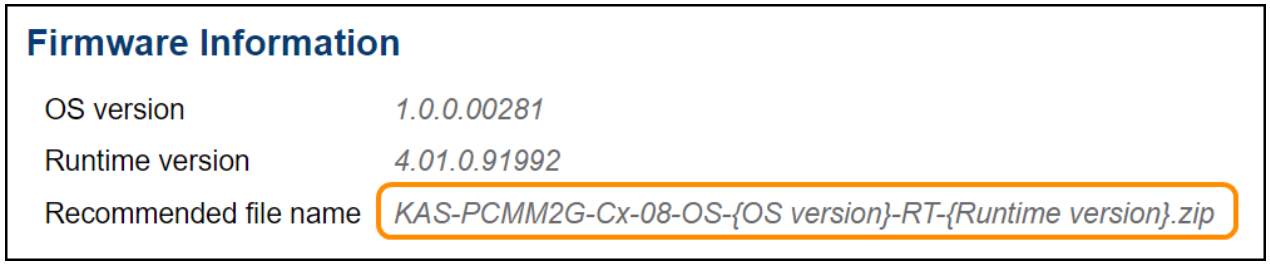


Figure 15-53: Firmware tab - Upgrade Firmware area

6. In the **Firmware Information** area, verify the **Recommended file name** for the upgrade file to download and use. (Figure 15-53)

   PCMM2G

   NOTICE

   - The PCMM2G firmware image is common across all available controller models.
   - Model-specific functionality and axis count limits are determined by the controller configuration and licensing, not by separate firmware images.

7. Open the applicable controller product web page:
   - PCMM2G Programmable Motion Controller
   - AKD PDMM Programmable Drive, Multi-Axis Master
8. In the **Software** tab, under the **Title** column, locate the appropriate firmware upgrade file.
9. Click the link for the upgrade file.
   The file automatically downloads to the location designated for the browser.
10. Close the browser window.
11. Return to the KAS the **Firmware** tab.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

12. In the **Firmware Upgrade** area, click the **Choose File** button. (Figure 15-54)
    A Windows Explorer window opens.

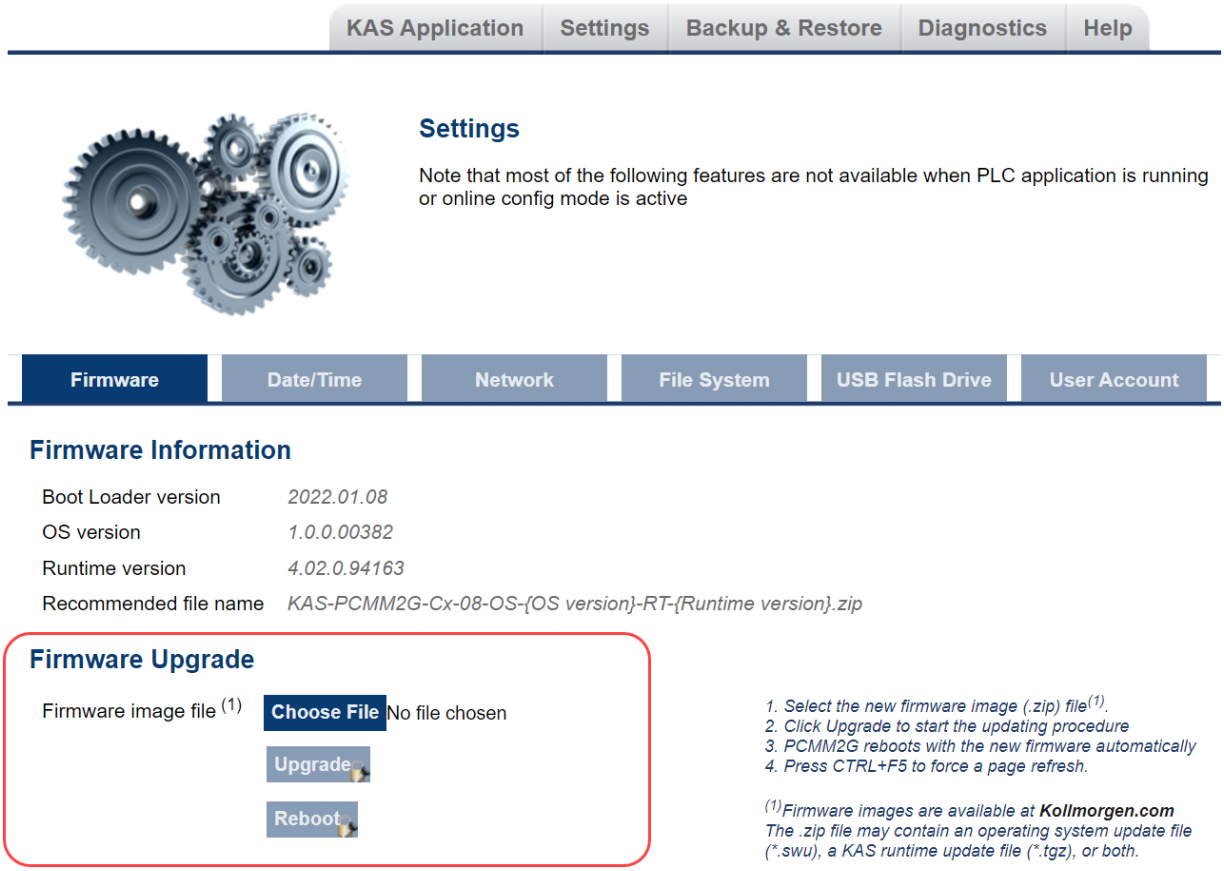


Figure 15-54: Firmware Upgrade area

13. Search for and select the firmware file downloaded in Step 9.
14. Click **Open** to select the file an close the window.
15. Click the **Upgrade** button to start the process.

TIP
If the **Upgrade** button is disabled, log into the Web server.
Click **Login** at the top of the web page and enter the password.

A message and a throbber are shown across the web page, indicating that maintenance is in progress.

The 7-segment display on the controller animates with chasing lights. 

IMPORTANT
Do **not** refresh the web page until firmware upgrade is completed.
If performing a boot loader upgrade, do **not** power-off or restart the PCMM2G until the boot loader upgrade is completed.

Successful Upgrade: The controller automatically reboots with new firmware.

When the reboot is complete, the user is logged out of the controller.
The webserver displays a message to indicate the user has been logged out.

Unsuccessful Upgrade

The Web server indicates the reason for the upgrade failure.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Example: Incompatible Firmware

AKD PDMM and PCMM Error Message

An error message, similar to this, appears if the wrong firmware file was downloaded:

```
The file provided is not compatible with this device.
The file name should be...
"KAS-PDMM-M-MCEC-{version}.img"
```

PCMM2G Runtime Error Message

```
Minimum runtime version requirement not met.
```

16. Press **<CTRL+F5>** to force the controller's web browser to refresh the page.
    The new firmware version appears in the **Firmware Information** area.

See Also

- "Booting the AKD PDMM or PCMM Controller" (→ p. 593)
- "Booting the PCMM2G Controller" (→ p. 593)
- "Controllers" (→ p. 767)
- "Recover Using the Web server" (→ p. 672)
- "PCMM2G - File Naming Conventions" (→ p. 1052)

11.10.7.3.2 View Firmware Information

The firmware installed on the controller is shown in the **Firmware Information** area. (Figure 15-55)

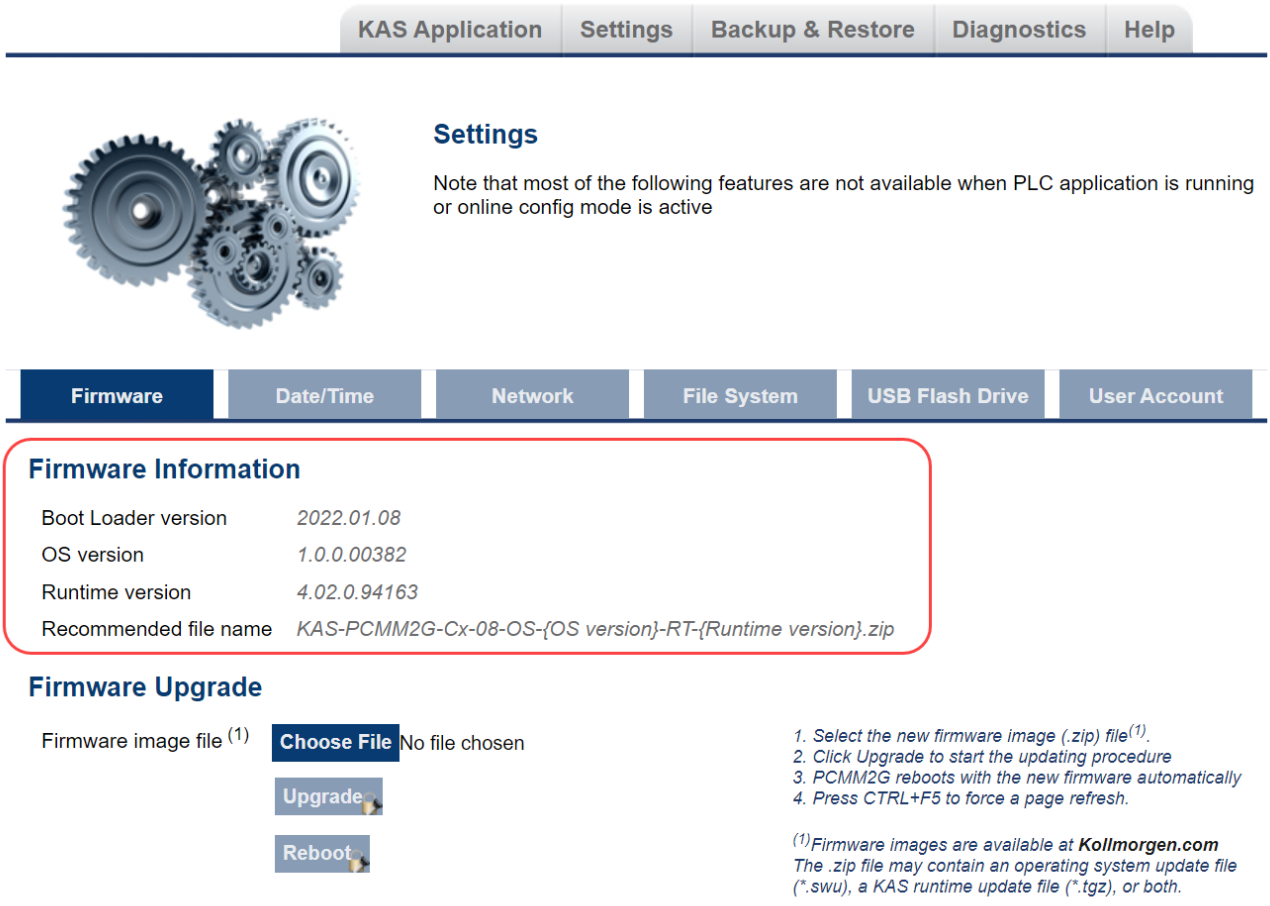


Figure 15-55: Firmware tab - Firmware Information area for PCMM2G

11.10.7.3.3.1 File Types

The firmware file types are different between the controllers.

- "PCMM2G - File Naming Conventions" (→ p. 1052)

See Also

- "Booting the AKD PDMM or PCMM Controller" (→ p. 593)
- "Booting the PCMM2G Controller" (→ p. 593)
- "Recover Using the Web server" (→ p. 672)

11.10.7.3.4 Recover Using the Web server

If the controller detects a problem in the firmware, the 7-segment display shows  .
This means a firmware recovery is needed.

These are the Recovery Modes:

- "PCMM2G Web server Recovery Mode" (→ p. 672)

⚠WARNING

- The Recovery Mode allows any firmware image file to be loaded into the controller's flash memory.
- The Recovery Mode does NOT verify the selected firmware file is compatible with the hardware model.
- Verify the **Recommended File Name** matches the selected firmware file.
- If an incompatible firmware file is loaded into the AKD PDMM's or PCMM's flash memory, the AKD PDMM or PCMM fails to boot into the Runtime image and fails to automatically boot into Recovery Mode.
- To recover from this situation requires manually booting into Recovery Mode.
- See "Boot from Recovery Image (B2 Button)" (→ p. 602).

11.10.7.3.5.1 PCMM2G Web server Recovery Mode

ⓘIMPORTANT

Recovery Mode is **not** automatic.
It requires manual intervention using a specially prepared USB flash drive.

11.10.8 Recovery USB Flash Drive

There are two kinds of Recovery USB flash drives:

- "Production Image Recovery" (→ p. 672) - restores the controller to a factory production state.
- "Customer Image Recovery" (→ p. 674) - restores the controller to a previously saved, customer configuration.

11.10.9 Production Image Recovery

This procedure is used to recover a PCMM2G controller using the official recovery firmware image.

- "Step 1: Prepare the PCMM2G Recovery Image" (→ p. 673)
- "Step 2: Flash the Image to USB Flash Drive" (→ p. 673)
- "Step 3: Boot and Recover the PCMM2G" (→ p. 673)

Pre-requisites

Hardware Requirements

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- A computer running Windows 10/11 or Linux with administrative privileges.
- A USB flash drive with at least 8GB capacity.
- A stable internet connection.

Software Requirements

- PCMM2G Recovery firmware image file (.wic.bz2).
  - Download the file from here: PCMM2G Recovery Operating System (OS) + Runtime (RT)
- Archive extraction software:
  - 7-Zip - for Windows or Linux.
  - WinZip - for Windows.
- Flashing tool (choose one):
  - balenaEtcher – cross-platform and user-friendly interface.
  - Rufus - Windows-only, with advanced options.

IMPORTANT

- Flashing erases all data on the USB flash drive.
- Back up any important files before proceeding.

11.10.10 Step 1: Prepare the PCMM2G Recovery Image

1. Download the PCMM2G recovery firmware **.wic.bz2** image file from here: PCMM2G Recovery Operating System (OS) + Runtime (RT).
   - It is recommended to download the same version of the recovery image as the existing Operation System and Runtime image.
   - See "Recovery Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (➔ p. 1054).
2. Extract the image using 7-Zip, WinZip, or another compatible archive tool:
   a. Right-click the .wic.bz2 file downloaded in Step 1.
   b. Select either **Extract Here** or **Extract to folder**.
   The result is a .wic file.

   IMPORTANT

   - Verify the extraction tool supports the .bz2 format.
   - 7-Zip is recommended for Windows and Linux users.

11.10.11 Step 2: Flash the Image to USB Flash Drive

1. Insert the USB flash drive into the computer.
2. Launch either the **balenaEtcher** or **Rufus** software.
3. Select the extracted .wic file as the source image.
4. Choose the USB flash drive as the target device.
5. Click **Flash** (balenaEtcher) or **Start** (Rufus) to begin.
6. Wait for the flashing process to be completed and safely eject the USB flash drive.

11.10.12 Step 3: Boot and Recover the PCMM2G

1. Insert the Recovery USB flash drive into X37 USB Port of the PCMM2G.
2. Hold the B2 button and power-cycle the PCMM2G.
3. Release the B2 button anytime after the 7-segment display shows  .
   - See "Booting the PCMM2G Controller" (➔ p. 593).
   - If a  7-segment display appears while bootup with the B2 button held down, there is not a bootable drive inserted in the X37 USB Port.
4. When the PCMM2G is in Recovery Mode, connect to the controller from a web browser using the PCMM2G's IP address (e.g., http://192.168.0.101).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

5. On the Recovery Mode page, click the **Recover** button to start the recovery procedure. (Figure 15-56)

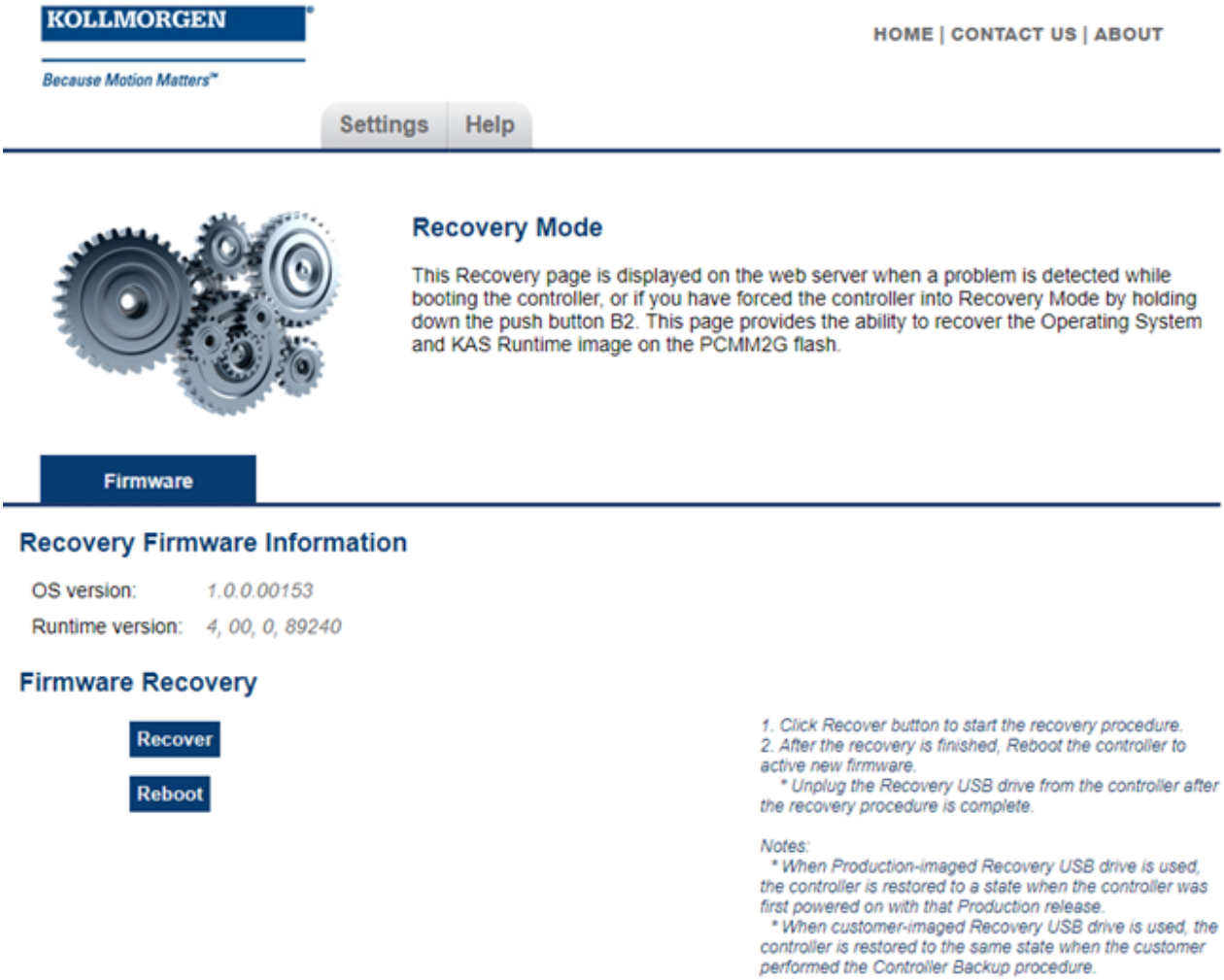


Figure 15-56: Recovery Mode tab

6. After the recovery is completed, reboot the controller.

Recommendations

- Connect to the controller using the web browser.
- Verify both the firmware OS and Runtime versions are as expected.
- Unplug the Recovery USB drive from the controller after the recovery procedure is completed.

If you used the:

- Production-image Recovery USB flash drive:
  - You must reprogram the controller with your configuration.
- Customer-imaged Recovery USB flash drive:
  - All settings are restored and the controller is fully operational.

When using the Production-image Recovery USB flash drive to perform Recovery, the controller is restored to a state when the controller was powered on with that Production release.

11.10.13 Customer Image Recovery

The Customer-imaged backup Recovery USB flash drive is generated using the "Controller Backup and Restore" (➔ p. 604) procedure.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

IMPORTANT

It is recommended to create the Recovery USB flash drive using the "Controller Backup and Restore" (➔ p. 604) procedure every time the firmware on the controller is updated.
This procedure creates an exact backup of the controller unlike the Production-image Recovery USB flash drive.
This is an important difference while recovering the controller.

When using the customer-imaged Recovery USB flash drive to perform the Recovery, the controller is restored to the same state when the customer performed the Controller Backup procedure.

See Also

- "Controller Boot Startup Script" (➔ p. 598)
- "Boot from Recovery Image (B2 Button)" (➔ p. 602)
- "Booting the PCMM2G Controller" (➔ p. 593)

11.10.13.1 Network tab

This tab is used to:

- View the current rotary switch position of the controller and its MAC address.
- Manually change the controller's IP address.

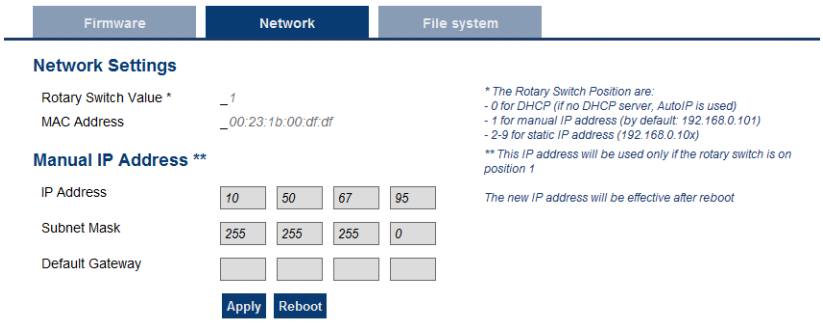


Figure 15-57: Example: Kollmorgen controller with a manually defined IP address.

11.10.13.1.1 Rotary Switch

The rotary switch on a Kollmorgen controller is set from a 0 to 9 position.

Switch Position	Description
Position 0	• The controller tries to get an IP address from a DHCP server. • If the DHCP fails, the controller uses AutoIP to get a usable IP address.
Position 1	Either: • The default static IP address of 192.168.0.101. • A custom IP address.
Position 2-9	• The controller is pre-configured with static IP addresses. • The IP addresses range is from: • 192.168.0.102 (Position 2) to 192.168.0.109 (Position 9)

NOTE

If a DHCP server is not present, the drive assumes an Automatic Private IP Address of the form 169.254.x.x.

11.10.13.1.2 Manually Change the IP Address

To connect and use your controller within your computer network, you may configure its IP address by using the web server.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. Verify the RS1 on the controller is set to position 1.
2. Open a web browser and enter the controller's IP address.
   The controller's web page opens.
   See "View the Controller IP Address" (➜ p. 602) to find this address.
3. Click the **Settings** tab.
4. Click the **Network** tab.
5. In the **Manual IP Address** area:
   a. Enter the **IP Address**.
   b. Enter the **Subnet mask**.
      The default is 255.255.255.0.
   c. Optional: Enter the **Default gateway** address if the controller is outside the local network.
6. Click **Apply**.
7. Click **Reboot**.
8. Verify the new IP address shows in the 7-segment display on the controller.

TIP

It is recommended to NOT use leading zeroes when entering an IP address.
The webserver automatically configures numbers with leading zeroes as octal values.
Example: Use **10.1.1.10** instead of **010.001.001.010**.

11.10.13.2 Security tab

This tab is used to:

- Change the user's Web server account password.
  See "Change the User's Web server Account Password" (➜ p. 679).
- Manage remote access to the controller via SSH.
  See "SSH Connection" (➜ p. 677).
- Manage non-secure connection access via HTTP.
  See "HTTP Configuration" (➜ p. 678).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

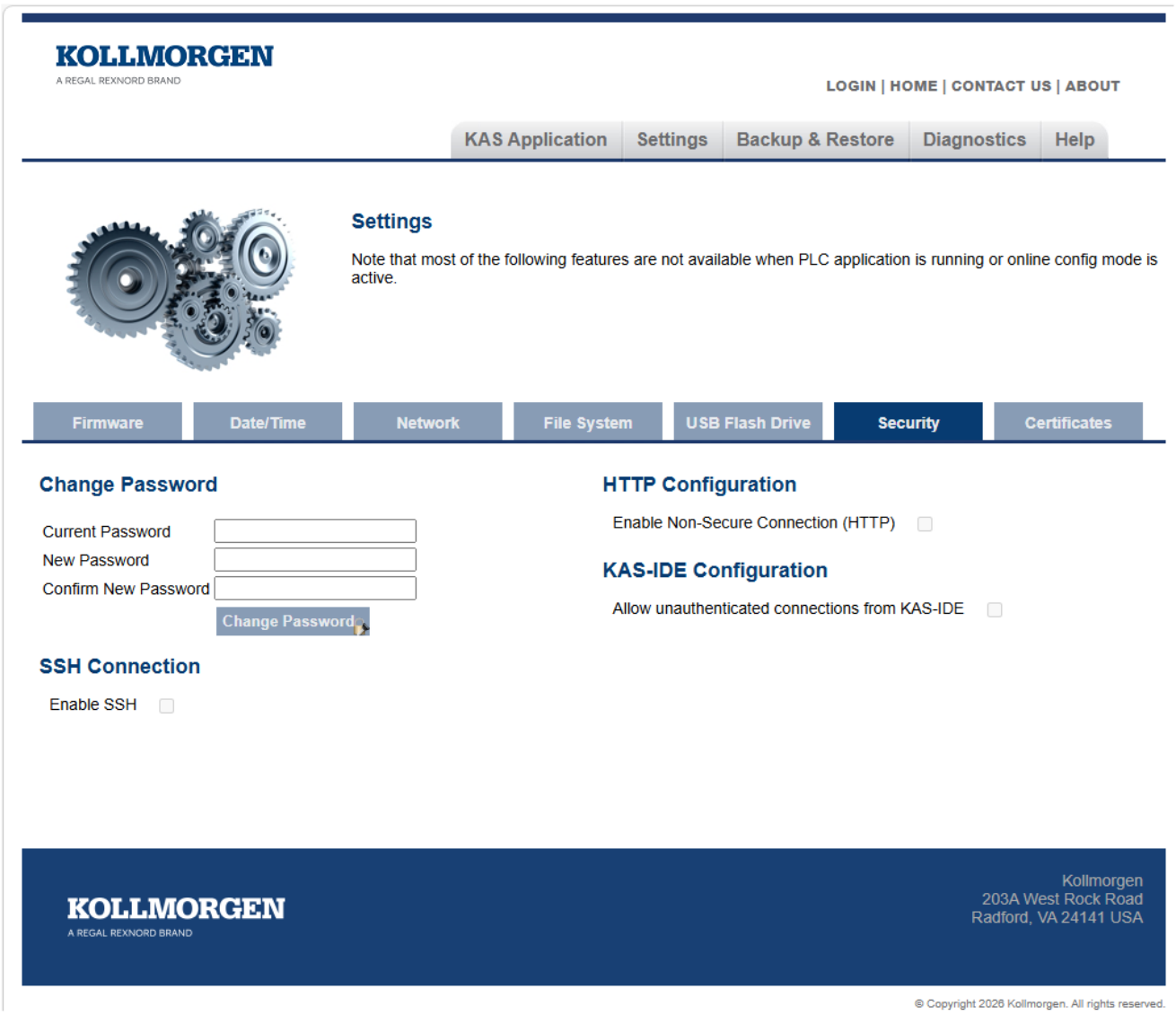


Figure 15-58: Security tab

11.10.13.2.1 Change the Password

This area is used to "Change the User's Web server Account Password" (➔ p. 679).

11.10.13.2.2 SSH Connection

This area is used to manage the Remote Access to the PCMM2G controller via SSH (Secure Shell).

- SSH is a network protocol that gives the users a secure way to access an external system remotely.
  - It provides a text-based interface by spawning a remote shell.
- By default, the SSH access is disabled on the PCMM2G to enhance security.
  - By keeping SSH access off, the PCMM2G controller minimizes the risk of external attacks.
  - SSH may need to be enabled in certain advanced troubleshooting scenarios.
  - SSH access allows diagnostic commands and access to system logs.
- See "Using SSH" (➔ p. 617).

Setting	Description
Enable SSH check box	Changing the state of the check box immediately changes the configuration on the controller. • A checked box indicates that remote access to the controller via SSH is supported. • An unchecked box indicates the controller can not be accessed remotely via SSH. **IMPORTANT** All existing SSH connections are terminated when SSH access is disabled. Save your work and complete any necessary tasks before disabling SSH.

11.10.13.2.3 HTTP Configuration

This area is used to manage the non-secure connection to the PCMM2G using Hypertext Transfer Protocol (HTTP).

- HTTP is a communication protocol that transfers unencrypted data between the user's browser and the webserver.
- Hypertext Transfer Protocol Secure (HTTPS) is a secure extension of HTTP that provides encrypting and verification to the data transferred between the user's browser and the webserver.
- By default, HTTP connections are disabled and will redirect to HTTPS on the PCMM2G to enhance security.
  - By keeping HTTP disabled, the PCMM2G minimizes the risk of vulnerable data being exposed.
  - HTTP may need to be enabled in certain legacy configurations that do not support HTTPS.

Setting	Description
Enable Non-Secure Connection (HTTP) check box	Changing the state of the check box changes the HTTP configuration on the controller and reboots the controller. • An unchecked box indicates the controller: • Has disabled non-secure (HTTP) connections. • Automatically redirects connections to an HTTPS. • A checked box indicates that non-secure (HTTP) connections on the controller are enabled.

11.10.13.2.4 KAS-IDE Configuration

This area is used to configure and manage the KAS-IDE connection settings with the controller.

- When authentication is enabled, KAS-IDE requires valid user credentials to connect to the controller.
- By default, unauthenticated connections from KAS-IDE to the controller are disabled to improve system security.
  - Disabling unauthenticated access helps reduce the risk of unauthorized or unintended controller access.
  - Allowing unauthenticated connections may be useful in specific, controlled, troubleshooting scenarios and should be enabled only when necessary.
- See Security - Authentication Access and Control - Authenticated KAS-IDE Access to PCMM2G
  - for a detailed description of the KAS-IDE operations that are protected by this authentication mechanism.

Setting	Description
Allow unauthenticated connections from KAS-IDE check box.	Changing the state of the check box immediately changes the configuration on the controller. • A checked box indicates that unauthenticated KAS-IDE connections to the controller are supported. • An unchecked box indicates the controller must require account authentication to connect to the KAS-IDE.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Important Notes

Security

- To maintain the security of the device, always disable SSH access, HTTP connections, and unauthenticated connections when they are no longer needed.

Access Control

- Ensure that only authorized personnel have access to the web interface and change security settings.

SFTP Support

- File transfer using SFTP (Secure File Transfer Protocol) is not supported when SSH is disabled.
- SFTP relies on SSH for secure file transfer, so you will need to enable SSH to use SFTP.

11.10.13.3 USB Flash Drive tab

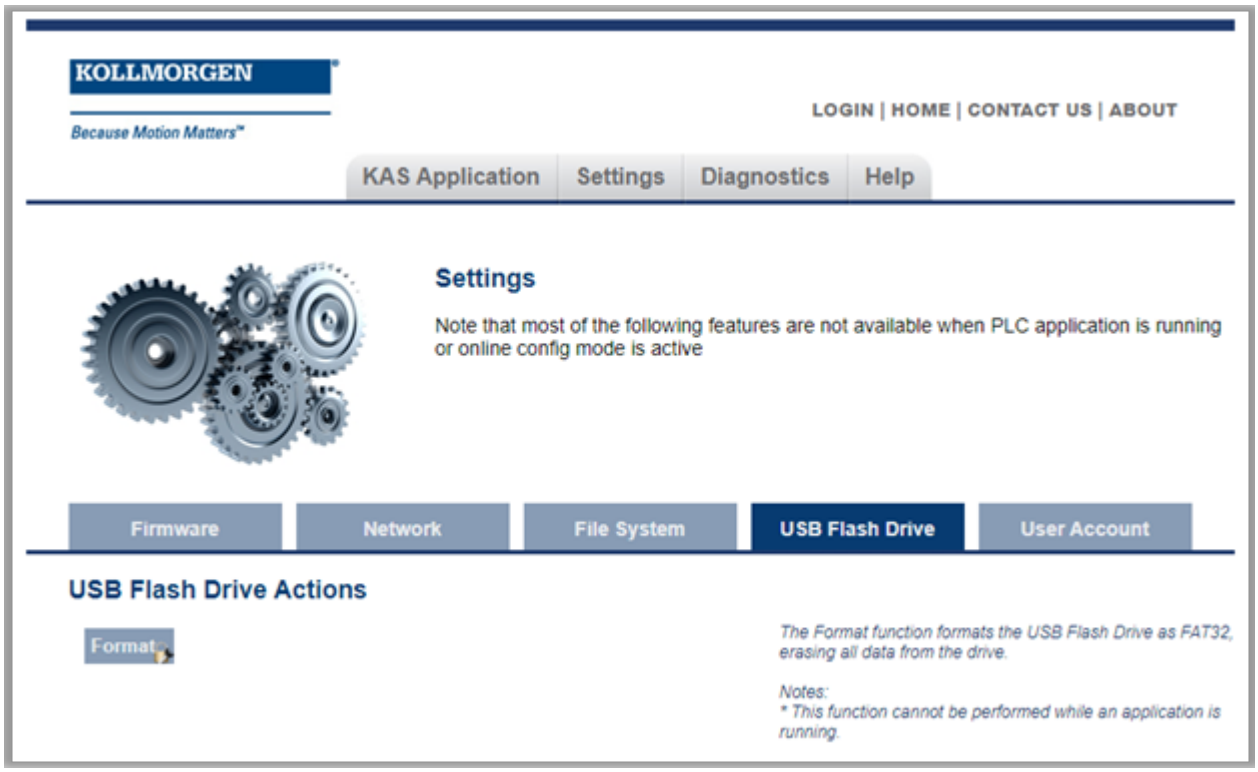


Figure 15-59: USB Flash Drive tab

This tab is used to format the USB flash drive as FAT32.

NOTE

PCMM2G supports only one USB flash drive.
If a USB hub is used to expand the USB ports on the PCMM2G, use only a single USB flash drive in the hub.

IMPORTANT

All data is erased from the USB flash drive.
The Format function cannot be performed while an application is running.

11.10.13.4 Change the User's Web server Account Password

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. Open a web browser and enter the controller's IP address.
2. Login to the controller as an Administrator.
3. Click the **Settings** tab.
4. For these controllers, click the applicable tab:

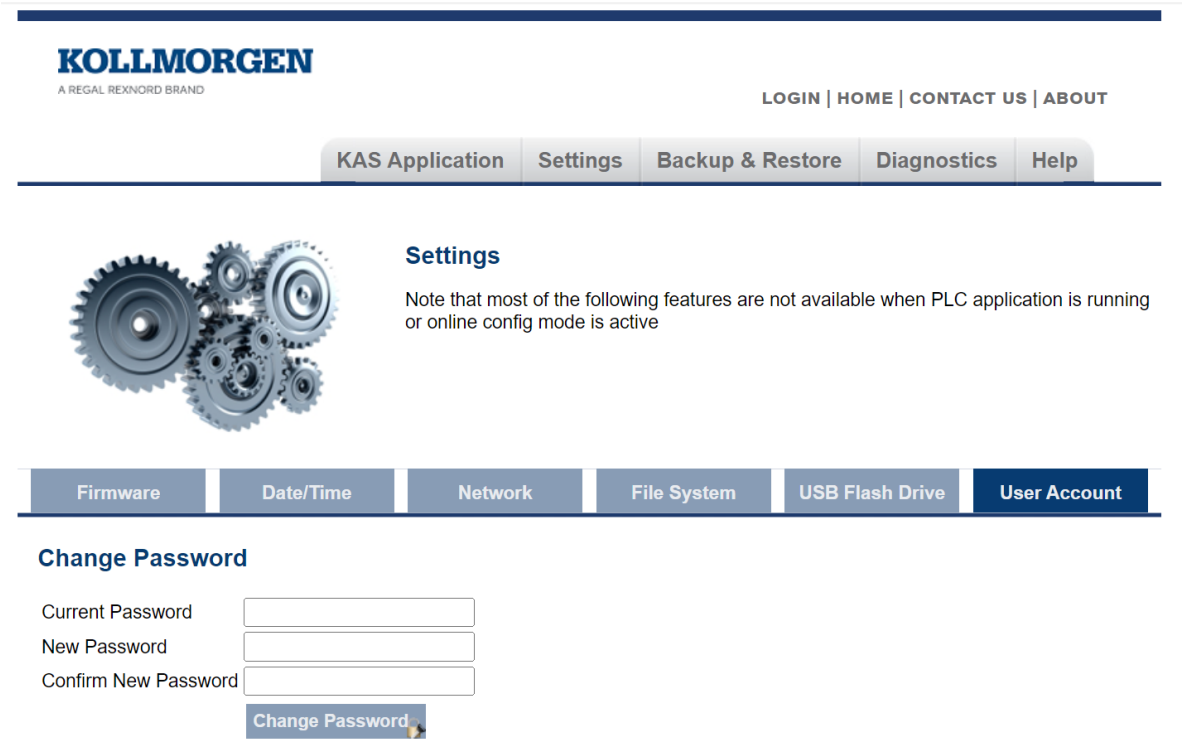


Figure 15-60: User Account tab

- For PCMM2G, click the Security tab. (Figure 15-61)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

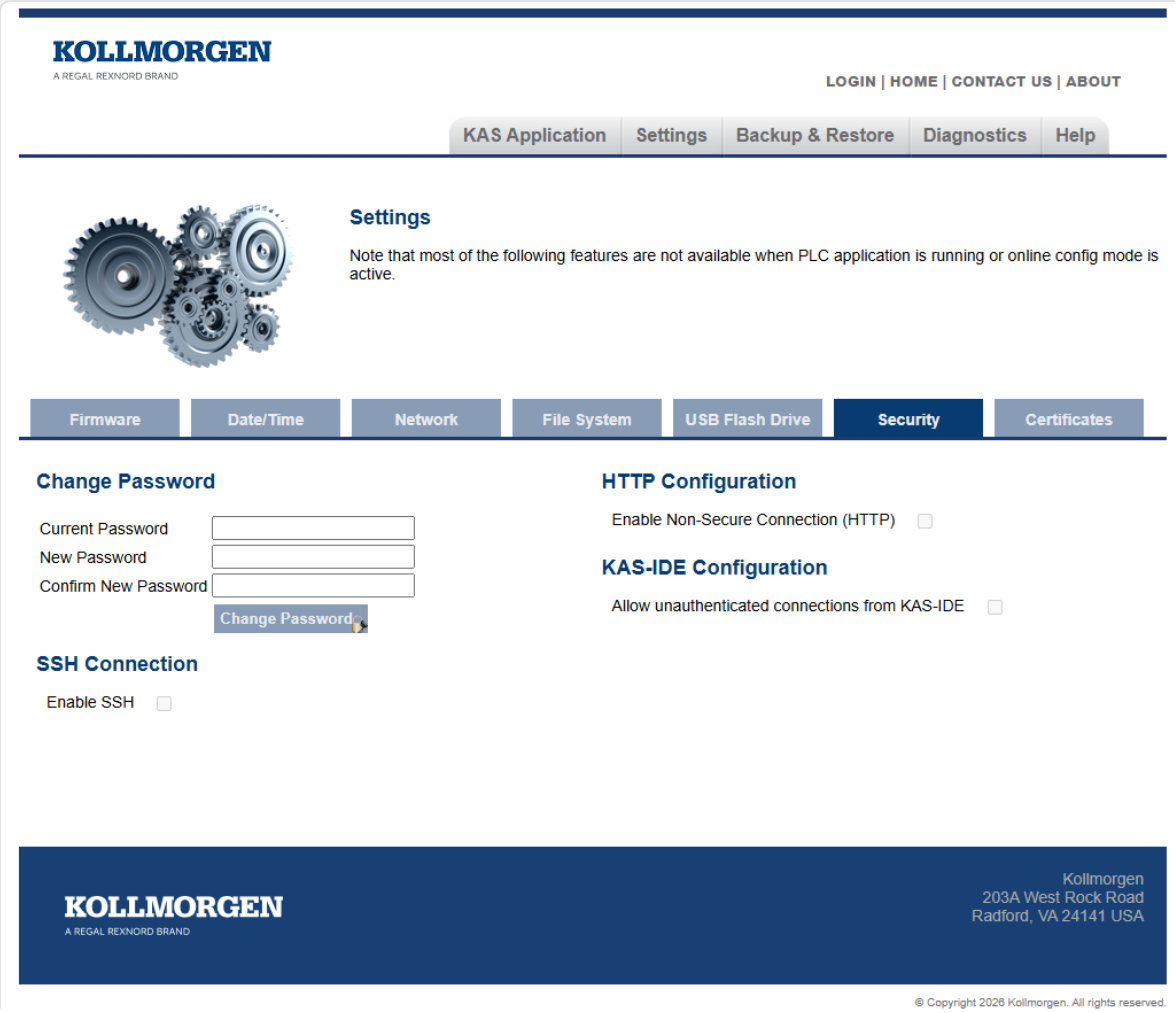


Figure 15-61: Security tab

5. Enter the **Current Password** in the text box.
6. Enter the **New Password** in the text box.

   Click for Password Conditions

   The new password must meet these conditions:

   - The password must be 6-20 characters long.
   - The password cannot contain:
     - Ampersands (&)
     - Number sign(#)
     - Quotes (' and ")
     - Semicolons (;)
     - Slashes (/ and \)
     - Spaces
7. In the **Confirm New Password** text box, enter the new password.
8. Click the **Change Password** button to save the changes.

11.10.13.4.1 I Forgot My Password

When this happens, set a new password from the IDE.

1. Open the Web server from the IDE.
2. Click the **Settings** tab.
3. Click the **User Account** tab.
4. Enter and confirm a new password.

This procedure is used to create a new password without entering the current one.

11.10.14 Backup and Restore from the Web server

The Backup and Restore functions are used to replicate a Kollmorgen controller.

The controller data is replicated on the PCMM2G USB flash drive.

11.10.14.1 All Controllers

The elements that are backed up or restored for all controllers are:

- Controller firmware.
- Network configuration.
- PLC Application.
- "Retain Variables" (➔ p. 292).
- Shared Directory settings.
- User data files.
  - See "User Data tab" (➔ p. 631).
- Web server and SSH user accounts configuration.

11.10.14.2 PCMM2G Backup / Restore Only

11.10.14.2.1 Backup

- Date/Time - "Network Time Protocol (NTP)" (➔ p. 665) and "Set Time Zone" (➔ p. 665) settings.
- SSH Connection Enable/Disable configuration.
- AKD / AKD2G / Kollmorgen Essentials Drive firmware packages and their respective drive parameters.
- EtherCAT network topology.

11.10.14.2.2 Restore

- The drive's data and firmware are copied from the USB flash drive to the controller flash when the **Restore Controller** function is selected.
- The **Restore EtherCAT Devices** function should be selected for restoring the data and the firmware of any replaced Kollmorgen AKD Series Servo Drives.

⚠WARNING

- A PCMM2G backup created from any PCMM2G axis model can be restored on a different axis model.
- The restored KAS project does not run if the physical axes need for the project is more than the restored model can support.

The Backup and Restore processes:

- Cannot be performed while an application is running.
- Takes several minutes to complete.
  - Do not power off the controller once started.
- The controller is rebooted after a Restore.

IMPORTANT

AKD PDMM or PCMM do not allow restoring the controller from the PCMM2G backup.
PCMM2G does not allow restoring the controller from the AKD PDMM or PCMM backups.

The AKD PDMM and PCMM are available with different CPU variants (800MHz, 1.2GHz single core, and 1.2GHz dual core).

- The AKD PDMM and PCMM models have the same functionality but, because of CPU speed differences, an application designed for the 1.2GHz model may not execute as expected on the 800MHz model.
  - An application designed for a single core CPU model may not execute as expected on the dual core model.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The Restore:
  - Does **not** allow a higher speed model to be replaced with a lower speed model.
  - Does allow a lower speed model to be replaced with a higher speed model if the Backup contains the model-specific Runtime firmware file.

Backup File Source	PCMM2G Restore
PCMM2G Backup	Yes
800MHz Backup	No
1.2GHz Single Core Backup	No
1.2GHz Dual Core Backup	No

TIP

This section provides an overview of the backup and restore processes.
See "Backup and Restore tab" (➔ p. 606) for more information.

11.10.14.3 Backup tab

Backup either the:

- "PCMM2G Controller and EtherCAT Devices Backup" (➔ p. 683)
- "EtherCAT Devices Backup" (➔ p. 685)

11.10.14.3.1 PCMM2G Controller and EtherCAT Devices Backup

All firmware packages are available to download from the Kollmorgen Support Network site.

- This function is used to replicate a PCMM2G.
- Backup Controller and EtherCAT Devices are used to replicate a PCMM2G.

The elements that are backed up or restored for all controllers are:

- Controller firmware.
- Network configuration.
- PLC Application.
- "Retain Variables" (➔ p. 292).
- Shared Directory settings.
- User data files.
  - See "User Data tab" (➔ p. 631).
- Web server and SSH user accounts configuration.

11.10.14.3.2.1 PCMM2G Backup / Restore Only

11.10.15 Backup

- Date/Time - "Network Time Protocol (NTP)" (➔ p. 665) and "Set Time Zone" (➔ p. 665) settings.
- SSH Connection Enable/Disable configuration.
- AKD / AKD2G / Kollmorgen Essentials Drive firmware packages and their respective drive parameters.
- EtherCAT network topology.

11.10.16 Restore

- The drive's data and firmware are copied from the USB flash drive to the controller flash when the **Restore Controller** function is selected.
- The **Restore EtherCAT Devices** function should be selected for restoring the data and the firmware of any replaced Kollmorgen AKD Series Servo Drives.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

⚠WARNING

- A PCMM2G backup created from any PCMM2G axis model can be restored on a different axis model.
- The restored KAS project does not run if the physical axes need for the project is more than the restored model can support.

Notes

- This operation cannot be performed while an application is running.
- Backup Controller takes several minutes to complete.
  - **Do not** power off the controller once started.

Backup EtherCAT devices save the network topology, the Kollmorgen AKD Series Servo Drive's (AKD / AKD2G / Kollmorgen Essentials Drive) firmware and parameters.

- For EtherCAT devices backup operation, a matching AKD / AKD2G / Kollmorgen Essentials Drive firmware package is required.
  - If a matching firmware package exists on the controller, it is used for future backups.
  - If a matching firmware package **does not** exist on the controller, you will be asked to provide it.

Procedure

1. Click the button to save the data to the USB flash drive. (Figure 15-62)



Figure 15-62: Backup Controller & EtherCAT Devices button

2. After starting, a confirmation message opens to confirm the backup. (Figure 15-63)

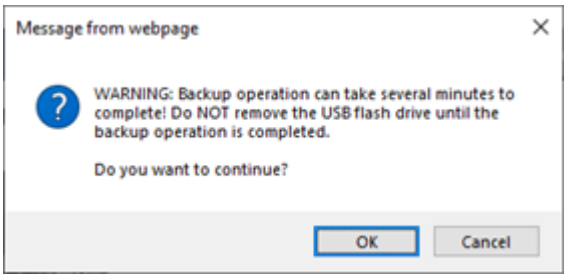


Figure 15-63: Confirmation message

3. Click **OK** to continue.
4. If a copy of the drives' firmware package is **not** present on the controller, you are prompted to provide one or more of these packages depending on the drive models present on the controller EtherCAT network. (Figure 15-64)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



Figure 15-64: Choose file

5. Provide the required firmware packages to continue with the backup function.
6. Click the **Continue Backup Controller & EtherCAT Devices** button.

See Also

- "EtherCAT Devices Backup" (→ p. 685)

11.10.16.0.1 EtherCAT Devices Backup

This function replicates:

- The network topology.
- All EtherCAT device unique IDs (Station Alias).
- The AKD, AKD PDMM, AKD-C, AKD-N, and AKD2G model FS1 dive's firmware and drive parameters.

Notes

- For AKD2G FS2 and FS3 models, only the drive parameters and the unique IDs are restored.
  - The firmware is not restored.
- A copy of the firmware package is required to accomplish the backup.

Different scenarios are possible when clicking this button:

- "Scenario 1 - Archived Firmware Copy = Firmware used on Drives" (→ p. 685)
- "Scenario 2 - Archived Firmware Copy Not = Network Configuration" (→ p. 686)
- "Scenario 3 - Only Unique ID Backed Up" (→ p. 686)

11.10.16.0.2.1 Scenario 1 - Archived Firmware Copy = Firmware used on Drives

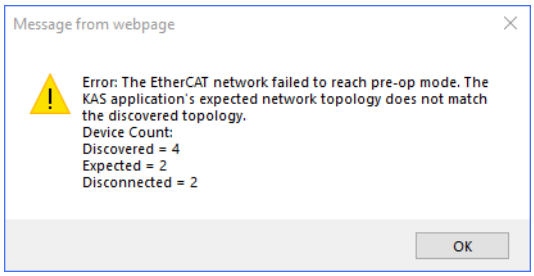
- If an archived copy of the same firmware package is on the controller as is used on the drives, clicking the button starts the backup.
  - If a copy of the firmware package cannot be found, you are prompted to browse for one or more.
- A link to the Kollmorgen website is provided; all firmware packages are on the site.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



11.10.16.0.3.2 Scenario 2 - Archived Firmware Copy Not = Network Configuration

- If an archive is found on the controller, but it **does not match** the network configuration, you are prompted to browse for one or download one from the website.
- This message appears if the EtherCAT network topology discovered during backup does not match the topology expected by the KAS application:
  - The message shows the number of discovered devices, the number that were expected, and how many are disconnected.



11.10.16.0.4.3 Scenario 3 - Only Unique ID Backed Up

- For all other EtherCAT devices (e.g., couplers, IOs, ServoStar drives, and third-party devices), only the unique ID is backed up.

TIP

The firmware package files on the Kollmorgen website are saved in a .ZIP format.
You must unzip the download to access the .TGZ file .

See Also

- "PCMM2G Controller and EtherCAT Devices Backup" (➔ p. 683)

11.10.16.1 Restore tab

IMPORTANT

AKD2G FS2 and AKD2G FS3 drive's firmware are not restored using the restore function.
Only the drive parameters and the Unique ID are restored.
See "Restore EtherCAT Devices" (➔ p. 687).

11.10.16.1.1 Restore Controller

The Backup and Restore functions are used to replicate a Kollmorgen controller.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The controller data is replicated on the PCMM2G USB flash drive.

The elements that are backed up or restored for all controllers are:

- Controller firmware.
- Network configuration.
- PLC Application.
- "Retain Variables" (→ p. 292).
- Shared Directory settings.
- User data files.
  - See "User Data tab" (→ p. 631).
- Web server and SSH user accounts configuration.

11.10.16.1.2 PCMM2G Backup / Restore Only

11.10.16.1.3.1 Backup

- Date/Time - "Network Time Protocol (NTP)" (→ p. 665) and "Set Time Zone" (→ p. 665) settings.
- SSH Connection Enable/Disable configuration.
- AKD / AKD2G / Kollmorgen Essentials Drive firmware packages and their respective drive parameters.
- EtherCAT network topology.

11.10.16.1.4.2 Restore

- The drive's data and firmware are copied from the USB flash drive to the controller flash when the **Restore Controller** function is selected.
- The **Restore EtherCAT Devices** function should be selected for restoring the data and the firmware of any replaced Kollmorgen AKD Series Servo Drives.

WARNING

- A PCMM2G backup created from any PCMM2G axis model can be restored on a different axis model.
- The restored KAS project does not run if the physical axes need for the project is more than the restored model can support.

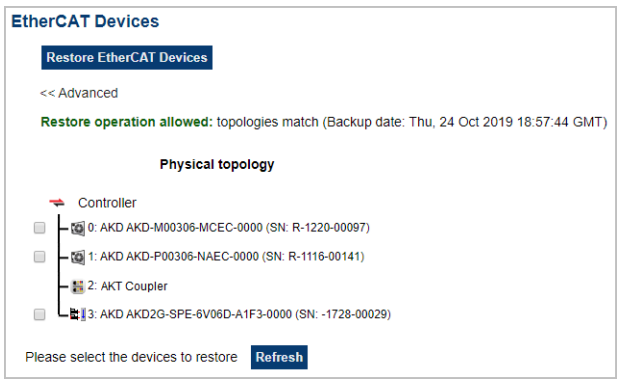
11.10.16.1.5 Restore EtherCAT Devices

This function automatically restores the data and firmware of any replaced AKD or AKD2G drives and the station alias of other EtherCAT devices.

11.10.16.1.6 Advanced

Click this button to create a map of the network and its components.

- If a valid firmware file is found, and the current topology matches the backup file, select the devices that need to be restored.
- Click **Restore EtherCAT Devices** to restore the drive's firmware.



11.10.16.2 Import / Export tab

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

These functions are used to:

- Save a copy of the controller and EtherCAT configurations backup file to a computer.
- Import the backup file to be used for restore functions.

11.10.16.2.1 Controller

Button	Description
Export Controller Backup	Click this button to save a backup of the controller firmware and configuration to a computer. The filename of the backup exported to the computer are: • PCMM2G: **kas-controller-2g-backup.zip**. • The file may be renamed to help identify the backup file with the machine
Choose File	Click this button to select a previously exported backup file to be imported.
Replace Controller Backup	Click this button to import the selected backup file, replacing any existing backup.

11.10.16.2.2 EtherCAT Devices

Button	Description
Export EtherCAT Backup	Click this button to save a backup of the EtherCAT configuration to a computer. • The filename of the backup exported to the computer is **kas-network-backup.tgz**. • The file can be renamed to help identify the backup file with the machine.
Choose File	Click this button to select a previously exported backup file to be imported.
Replace EtherCAT Backup	Click this button to import the selected backup file, replacing any existing backup.

11.10.17 Diagnostics tab

This page has information about:

- "Errors and Alarms sub-tab" (→ p. 688)
- "Hardware Status sub-tab" (→ p. 689)
- "Crash Reports sub-tab" (→ p. 691)
- "EtherCAT Network Topology sub-tab" (→ p. 691)

11.10.17.1 Errors and Alarms sub-tab

The Errors and Alarms sub-tab and the 7-segment display shows controller errors or alarms generated by the system.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

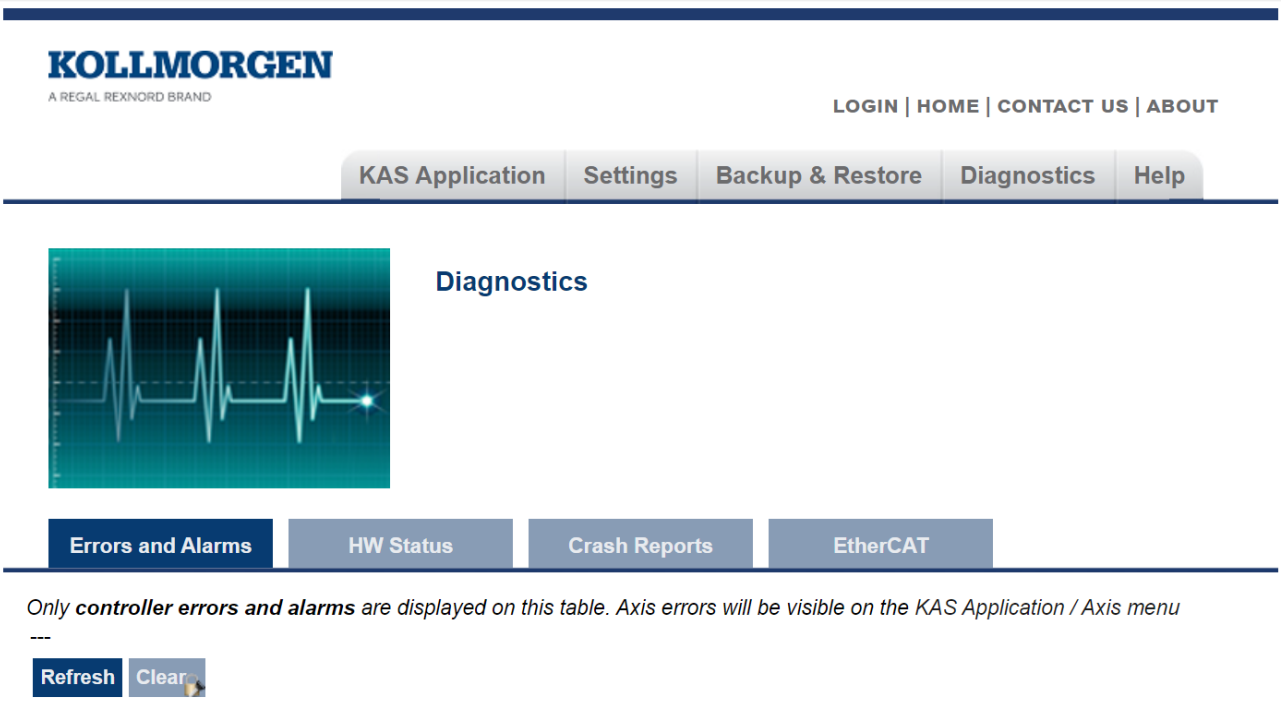
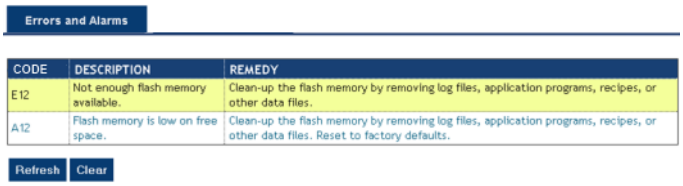


Figure 15-65: Diagnostics tab - Errors and Alarms sub-tab

- The **Refresh** button updates the list.
- The **Clear** button removes the contents of this tab.

IMPORTANT

A common error or alarm is because the flash memory is full.
This is often caused by heavy use of the PLC Advanced File function blocks.
Some errors or alarms are only cleared by powering off and restarting the Kollmorgen controller.



See "Controller Error and Alarm Descriptions" (→ p. 577) for a complete list of codes.

TIP

Axis errors are on the KAS Application **Axis** tab.

11.10.17.2 Hardware Status sub-tab

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



Figure 15-66: Diagnostics tab - HW Status sub-tab

Controller	Description
Storage Space	This is both the used and total available amount of storage space in megabytes (MB). • **Used** is the amount of file space currently being used by all files in flash memory. • **Total** is the total amount of file space available for files in flash memory.
Available Memory	This is the amount of RAM memory available on the Controller.
CPU Usage	This is the current load on the CPU. • If the load goes over 90%, the field turns red. • PCMM2G: CPU usage is measured across the CPU cores that run the motion engine and PLC program.
CPU Temp	This is the temperature of the CPU in Celsius. • If the CPU temperature is greater then the CPU warning limit, the temperature background color changes to yellow. • If the CPU temperature is greater than the CPU critical temperature, the temperature background color changes to red. • PCMM2G: The normal operating range is 0 °C to 81 °C.
CPU Fan Present	This is either **True** or **False**, depending if there is a CPU fan present in the controller.
Refresh	Click this button to refresh the Hardware Status information.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Controller	Description
Reboot	Click this button to reboot the web server. **IMPORTANT** Do not try to refresh the web page until the Web server has rebooted.

11.10.17.3 Crash Reports sub-tab

The files shown on the Crash Reports sub-tab are reports of the process that failed if there is a crash.

NOTE

These archives can be sent to Kollmorgen for analysis:
.ZIP for PCMM2G (Figure 15-67)



Figure 15-67: PCMM2G

11.10.17.4 EtherCAT Network Topology sub-tab

This sub-tab shows the EtherCAT network topology and the data for each node in the network. (Figure 15-68 and Figure 15-69)

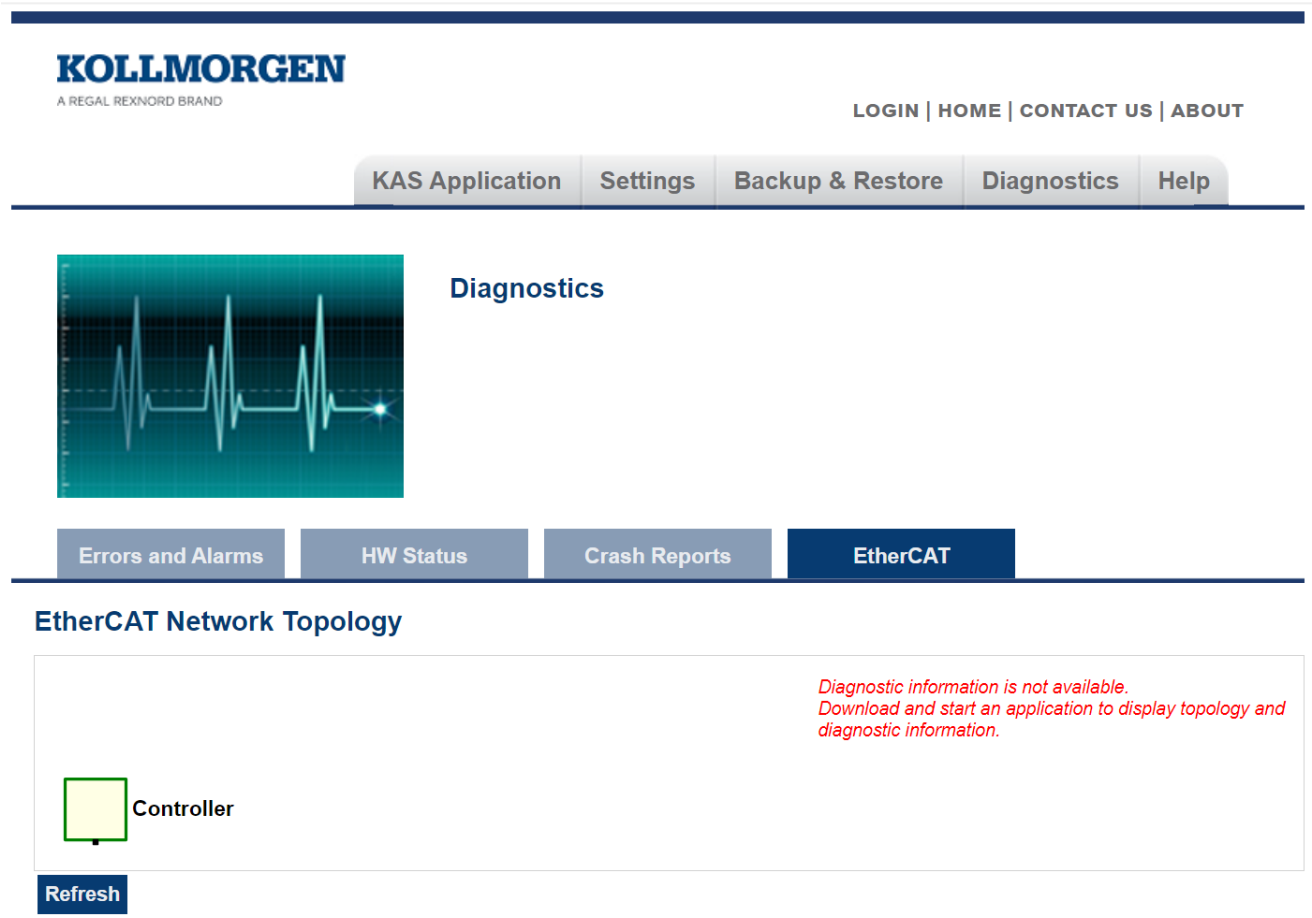


Figure 15-68: Diagnostics tab - EtherCAT sub-tab without data

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

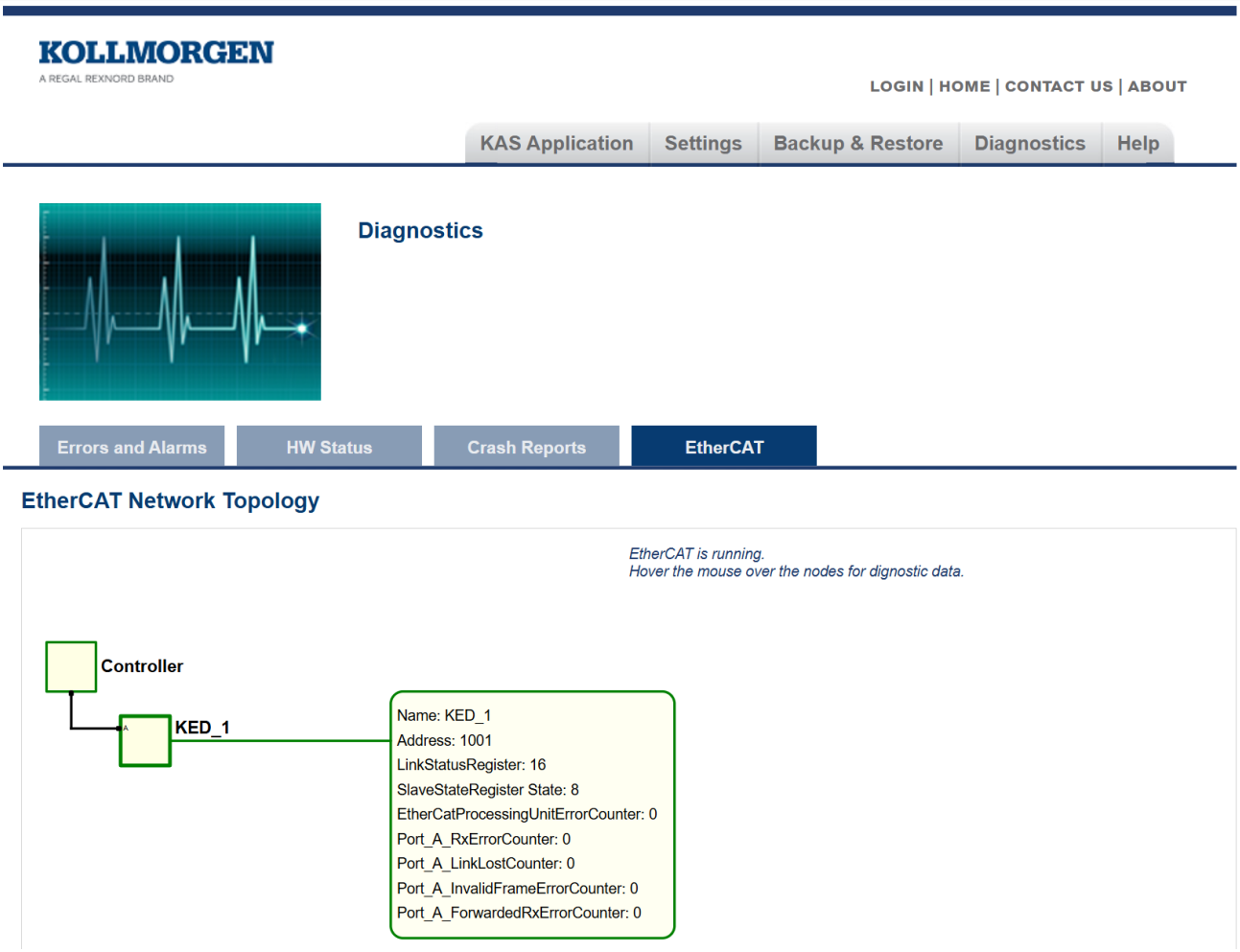
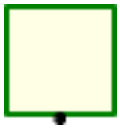
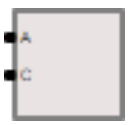


Figure 15-69: Diagnostics tab - EtherCAT sub-tab with data

11.10.17.5 EtherCAT Diagnostics tab

This tab is used to identify failed or degraded connections between nodes, (e.g., Controller, drives, and I/O) and non-accessible nodes or cables.

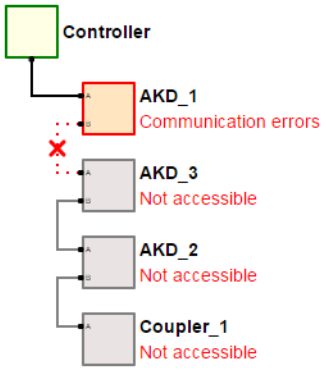
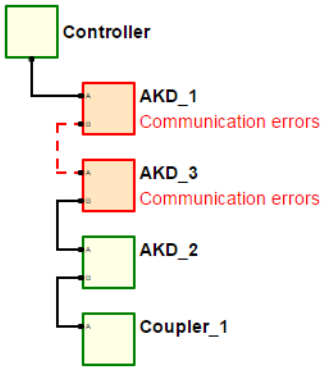
- The tab's content includes a visualization of the network topology, and it's last-known state.
- The tab contents are updated using the **Refresh** button.

Element	Description	Meaning
	Green Box	This is a node with normal communication.
	Red Box	This is a node that is experiencing communication errors.
	Gray Box	This is a node which cannot be reached, but was known to exist.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Element	Description	Meaning
	Black Line	This cable is working normally.
	Gray Line	This cable is not accessible.
	Red Dashed Line	This cable is experiencing degraded communication.
	Red X in Dotted Line	The connection between the nodes is broken.

These are examples of error situations when the application is running:

EtherCAT Stopped due to an Error	EtherCAT Network Communication Errors
	

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

12 Tools

12.1 Cam Profile Editor

Double-click a profile in the Project Explorer to open the cam profile editor in a new tab of the workspace.

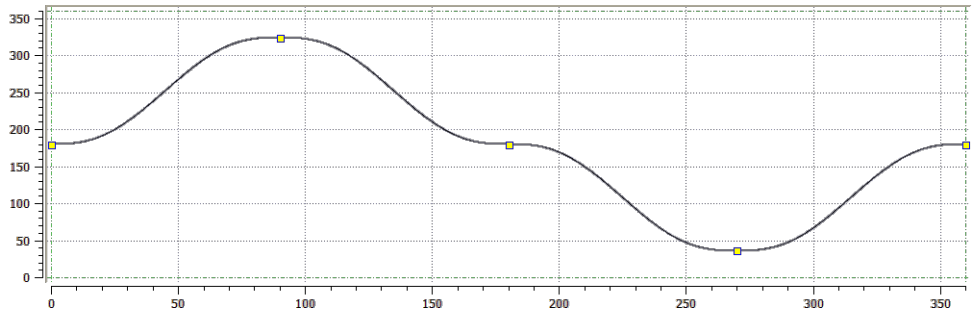


Figure 16-1: A cam profile

The cam profile editor is used to create and/or modify a profile definition that describes the position evolution of the cam.

- This evolution is displayed in a 2D graphical format.
- You can add, delete, or modify cam elements which consist of points and lines.
  - Based on those elements and some constraints, the KAS-IDE calculates a complete cam shape.
  - See "CAM Profile Segments" (➔ p. 702) for more information on the segment types.
- **Master/Input** (X-Axis) and **Slave/Output** (Y-axis) coordinates can be specified to define the position.
- It is also possible to visualize the velocity, acceleration, and jerk diagrams.

12.1.1 Windows Overview

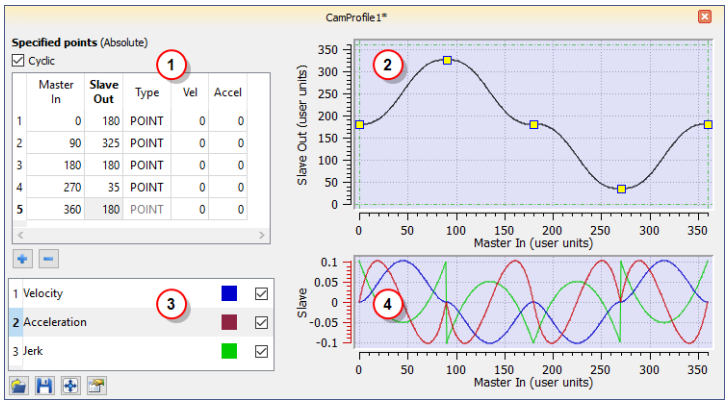


Figure 16-2: Cam Profile Editor Main Window

The cam profile editor contains these distinct areas separated by splitters:

1. The cam table (see call out 1) displays each element and allows editing of the cam.
2. The Graphical Area for the cam profile 2
   The upper graph displays a graphical representation of the cam elements
3. The Curve Selection and Color Table 3 allows you to select which plots (velocity, acceleration and jerk) are displayed
4. The Graphical Area for Curves 4
   The lower graph displays a graphical representation of the velocity, acceleration and jerk plots

TIP

Undo (Ctrl+Z) and Redo (Ctrl+Y) operations are available for any changes you make to the cam profile.

12.1.1.1 Splitters

Splitters are used to resize each part of the display.

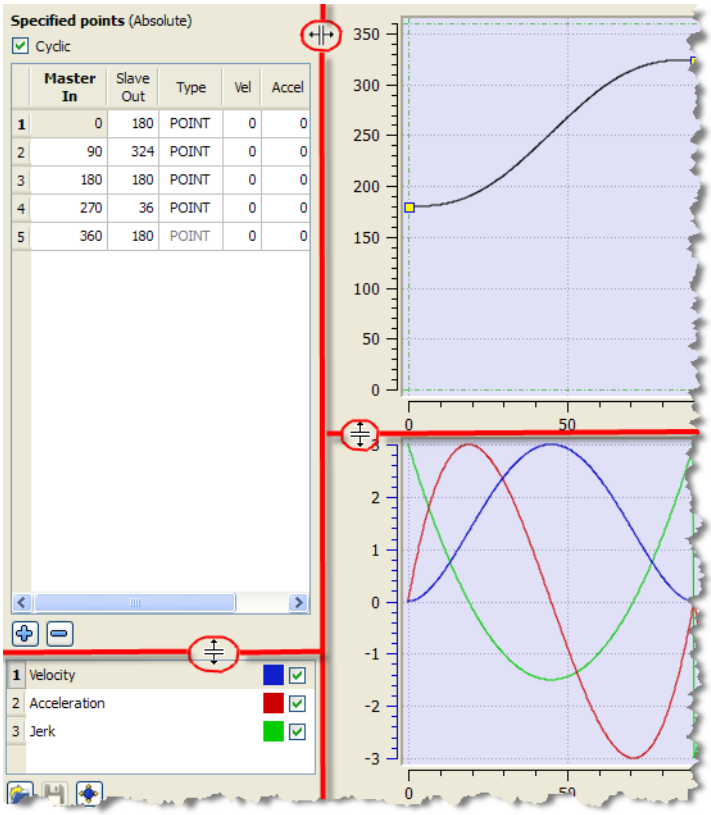


Figure 16-3: Example: Splitters

NOTE

The tables and the graphs are separated by a vertical splitter so that you can completely hide the tables to increase the graphical area.

See "Add CAM Profiles" (→ p. 247) and "Profiles" (→ p. 91) about cam profiles.

12.1.2 CAM Profile Editor's CAM Table

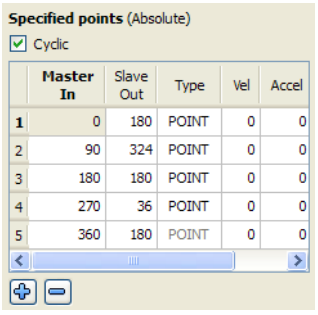


Figure 16-4: Cam Table

NOTE

By default, when a new profile is created, the CAM Profile contains five points.
These points could be different from those in the image, depending on the offsets and amplitudes specified in

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

the CAM Profile.
See "Create a CAM Profile" (➔ p. 247).

12.1.2.1 Cam Editor - Table Parameters

Column	Description
Master/In	The time is located in the **Master/In** column. It is the X-axis of the cam profile graph.
Slave/Out	The position is located in the **Slave/Out** column. It is the Y-axis of the cam profile graph.
Type	The **Type** column defines whether this element is a point or a line. • If the element is a line, In/Out specifies the start point of the line. • The next element in the table defines the end of the line. • The last element type in the table cannot be changed, since a line cannot exist as the last element.
Vel	The **Velocity** of the current element (first derivative).
Accel	The **Acceleration** of the current element (second derivative).

12.1.2.2 Cyclic Cam Element

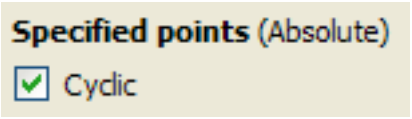


Figure 16-5: Cyclic check box

If the **Cyclic** check box is selected, the cam profile is executed cyclically.

- This means, when the axis attached to this cam runs continuously, the same profile is executed again.
  - In this case, the first and last element must have the same **Vel** and **Accel** values.
  - Changing the **Vel** or **Accel** value of the first or last elements automatically changes the other element's value.

NOTE

When **Cyclic** is first turned on, the Vel/Accel values are automatically copied from the first element to the last element when they do not match.
A warning dialog opens indicating this change has happened.
This alert cannot be suppressed until KAS is closed.

There are some combinations of points and lines where **Cyclic** is automatically turned off.

If this occurs, the **Cyclic** checkbox label is changed to **Cyclic (automatically turned off)**.

These changes to the profile automatically turn off **Cyclic**:

- The first element has been changed from a point to a line.
  - If needed, **Cyclic** can be manually turned back on.
    This will affect the velocity of the last element.
- The next to last element has been changed from a point to a line and now both first and next to last elements are lines.
  - **Cyclic** is disabled and can only be re-enabled when the first and next to last elements are not lines.
- The first element is a line and the first element is moved.
  - If needed, **Cyclic** can be manually turned back on.
    This will affect the velocity of the last element.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The first element is a line and the second element is moved.
  - If needed, **Cyclic** can be manually turned back on.
    This will affect the velocity of the last element.
- The first element is a line and the last elements velocity (or slope line) has changed.
  - If needed, **Cyclic** can be manually turned back on.
    This changes the velocity setting just made.

12.1.2.3 Use the Cam Table to Change an Element

Change a cam element by clicking in the **Master/Input**, **Slave/Output**, **Vel**, or **Accel** column and typing in a new value.

For **Type**, see "Change the Type of a Cam Element" (→ p. 698).

The graphs are updated automatically when an element changes.

Rules apply to the value entered:

- The **Master/Input** value must lie between adjacent **Master/Input** points.
- The **Master/Input** value of the first and last point cannot change.
  - These values are determined by the profile properties X offset and X amplitude.
- The **Slave/Output** value must lie between the Y offset and Y amplitude set in the profile properties.

If an invalid value is entered (due to the interpolation calculation), it is superseded with the original value without any error message.

12.1.2.3.1 Interpolation

The section between two consecutive cam elements is automatically calculated by a fifth order polynomial algorithm.

Modification of one cam element only affects the two adjacent segments.

12.1.2.4 Change the Type of a Cam Element

The type of element can either be a point or a line.

1. Double-click in the **Type** column of an element and click the down arrow.
   A list of choices is displayed as shown.
2. Select the type of element from the list.

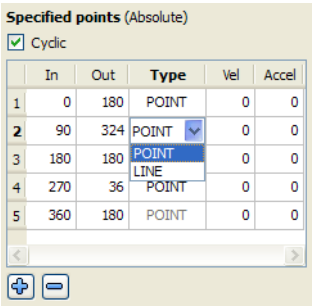


Figure 16-6: Modifying an Element Type

12.1.2.5 Cam Table Contextual Menu

Right-click an entry in the cam table to use the contextual menu.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

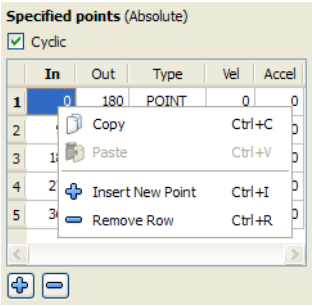


Figure 16-7: Cam Table Contextual Menu

Command	Shortcut	Description
Copy	Ctrl+C	Copy data from the selected cell in the clipboard.
Paste	Ctrl+V	Paste the data from the clipboard into the selected cell.
Insert New Point	Ctrl+I	Inserts a new row in the cam table above the highlighted entry. This command is described in "Add a Point to the Cam Table" (→ p. 699).
Remove Row	Ctrl+R	Deletes the row that contains the highlighted entry. This command is described in "Remove a Point from the Cam Table" (→ p. 700).

12.1.2.6 Add a Point to the Cam Table

You can add a point to the cam table using one of these methods:

- Use the menu in the cam table (shown in "Cam Table Contextual Menu" (→ p. 699)).
- Click the button  located below the cam table.
- Use the menu in the cam profile graph.

All of these methods open the **Add New Point to CAM Table** dialog box:

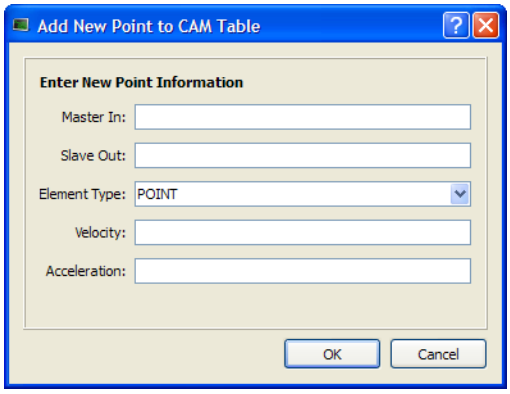


Figure 16-8: Add New Point to CAM Table dialog

Field/Command	Description
Master In	The X value of the new point.
Slave Out	The Y value of the new point.
Element Type	POINT or LINE
Velocity	The velocity of the new point (first derivative).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Field/Command	Description
Acceleration	The acceleration of the new point (second derivative).
OK	Accept the entry and verify if the point can be added.
Cancel	Cancel the dialog box – no point is added.

When you click OK, a check is performed to see if the point can be added to the cam profile.

If not, an error message opens.

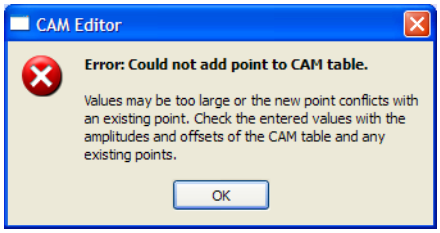


Figure 16-9: Cam Table Contextual Menu

If no problem is found, the point is added to the cam table and the graphical plots are updated.

NOTE

A new point cannot be inserted above the first element in the cam table.

12.1.2.7 Remove a Point from the Cam Table

Use these methods to remove a point from the cam table:

- Use the menu in the cam table (shown in "Cam Table Contextual Menu" (→ p. 699)).
- Click the button located below the cam table.
- Use the menu in the cam profile graph.

The selected point is removed without prompting.

IMPORTANT

The first and last points cannot be removed.

12.1.3 Cam Profile Graph

The upper graph displays the points and lines specified in the cam table along with the calculated curve. It also allows you to add, delete or modify a cam element.

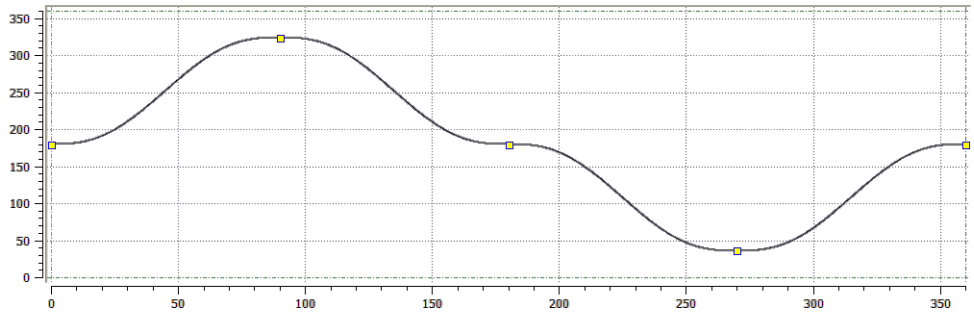


Figure 16-10: Cam Profile Graph

- Points and endpoints of lines are displayed as yellow squares ( ) in the graph.
- The profile offset and amplitude specified in the cam profile editor are displayed with a green dashed rectangle.
  - See "Create a CAM Profile" (→ p. 247).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The yellow squares are always contained within the green dashed rectangle.
  - Calculated points can extend outside it.

12.1.3.1 Modifying an Element

Change the profile by moving a point with the mouse.

1. Move the mouse over a yellow square.
   The cursor becomes ✥ indicating that the point can be selected.
2. Click to select the point and hold down the mouse button (left-click).
   When the mouse is moved, the point follows the cursor.
   Graphical curves and In/Out values are dynamically updated.

- When a point is selected, a slope line is drawn over the point.
- This line is dashed purple with two additional grips (◆) attached to it.
- The slope line can be used to change the velocity of the selected point.

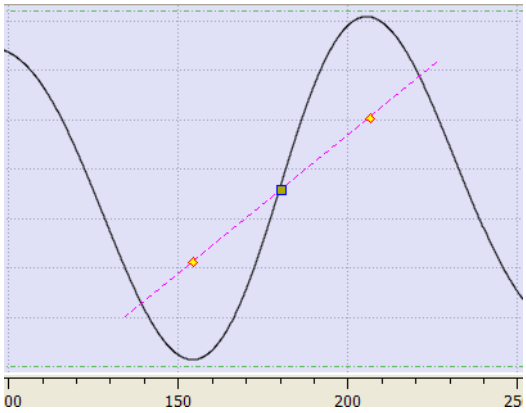


Figure 16-11: Cam Profile Graph - Slope Line

12.1.3.2 Change the Velocity of a Point

Change the velocity of the selected point.

1. Move the mouse over a slope grip (◆).
   The cursor changes to an open hand.
2. Click to select the grip and hold down the mouse.
   The cursor changes to a closed hand.
3. When the mouse is moved, the slope line follows the cursor.
   The selected point is rotated and causes the velocity of the point to change.
   Graphical curves and Vel value are dynamically updated.

12.1.3.3 Cam Profile Graph Contextual Menu

A right-click on the cam profile graph displays a contextual menu.

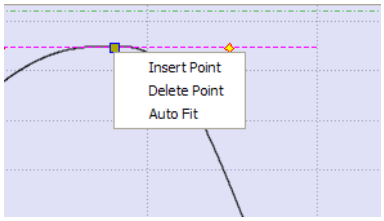


Figure 16-12: Cam Profile Graph - Contextual Menu

Command	Description
Insert Point	Inserts a new point at the X-Y location of the cursor.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Command	Description
Delete Point	Deletes the highlighted point. If the mouse is not near enough to a point, no point is highlighted and this command remains grayed-out.
Auto Fit	Adjusts the zoom and pan settings so that the entire graph is displayed in the graphical area.

12.1.3.4 Zoom In and Out

In the cam profile graph, you can zoom in or out.

1. Move the cursor in the graphical area.
2. Turn the mouse wheel forward or backward.

The current cursor becomes the center point of the zoom function and the area under the cursor remains stationary on the graph.

12.1.3.5 Panning

In the cam profile graph, you can also pan (or move) in any direction.

1. Click on any part of the graph (but not on a yellow square) and hold down the mouse button (left-click)
2. Move the mouse to move the graph accordingly

12.1.3.6 Restoring Zoom and Pan Settings

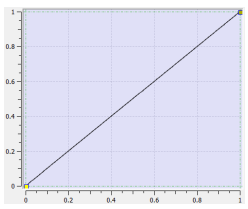
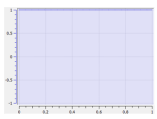
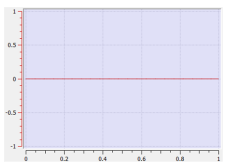
Click on the Auto Fit button  or select the Auto Fit command in the cam profile graph menu.

The entire curve is now shown in the graphical area.

12.1.4 CAM Profile Segments

- "Line Segment Type" (→ p. 702)
- "Parabolic Segment Type" (→ p. 703)
- "Point Segment Type" (→ p. 703)
- "Spline Segment Type" (→ p. 704)

12.1.4.1 Line Segment Type

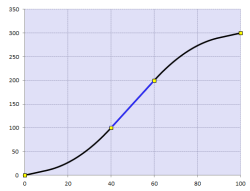
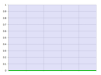
Segment Curve	Velocity	Acceleration	Jerk
			

Supported by	• KAS-IDE Profile Editor • MLProfileBuild
Continuous Velocity	✗
Continuous Acceleration	✗

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Interpolation Method	Linear function: f(x) = Ax + B
Advantages	• This segment type easily defines constant-velocity segments. • It can be used in very large numbers to define profiles of any size without having to specify the velocity or acceleration at the segment endpoints.
Disadvantages	Profiles can result in discontinuous velocities.

12.1.4.2 Parabolic Segment Type

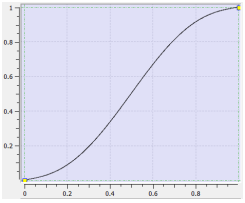
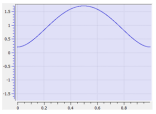
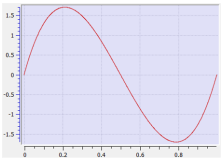
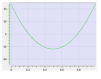
Segment Curve	Velocity	Acceleration	Jerk
			

In the example:

- The blue line represents the linear (constant velocity) part of the segment.
- The black lines represent the parabolic (constant acceleration) parts of the segment.

Supported by	• KAS-IDE Profile Editor • MLProfileBuild (parabolic option)
Continuous Velocity	✓
Continuous Acceleration	✓
Interpolation Method	Linear function: f(x) = Ax + B 2nd order polynomial: $f(x) = Cx^2 + Dx + E$
Advantages	• This segment type is used to define constant acceleration portions of a profile. • This minimizes the peak acceleration needed to move from one cam point to another. • This can be useful when the motors cannot support the accelerations used by other segment types.
Disadvantages	Acceleration is discontinuous which can lead to additional electrical stress on the drives and motors.

12.1.4.3 Point Segment Type

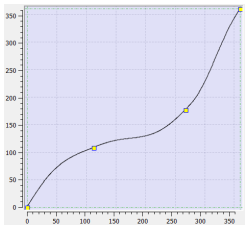
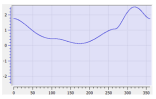
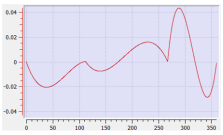
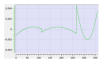
Segment Curve	Velocity	Acceleration	Jerk
			

Supported by	• KAS-IDE Profile Editor • MLProfileBuild (default option)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Continuous Velocity	✓
Continuous Acceleration	✓
Interpolation Method	5th order polynomial: $f(x) = Ax^5 + Bx^4 + Cx^3 + Dx^2 + Ex + F$
Advantages	With only a few segments, this type can be used to define profiles with continuously changing accelerations. Example: Sinusoidal profiles can be emulated with 6 to 12 point segments.
Disadvantages	• Specify the velocity and acceleration at the endpoints for each segment. • It is difficult to use the point segment type to define constant acceleration or constant velocity segments.

12.1.4.4 Spline Segment Type

Segment Curve	Velocity	Acceleration	Jerk
			

Supported by	• KAS-IDE Profile Import. • The points are created by separate software and is imported into KAS-IDE project.
Continuous Velocity	✓
Continuous Acceleration	✓
Interpolation Method	3rd order polynomial: $f(x) = Ax^3 + Bx^2 + Cx + D$
Advantages	• With only a few segments, this segment type can be used to define profiles with continuously changing velocities. • Only the positions of the master and slave need to be specified. • This produces smoother profiles than using line segments.
Disadvantages	Since only positions are specified, the user has less control over the velocities and accelerations that occur throughout the profile.

12.1.5 Curve Selection and Color Table

Velocity (first derivative), acceleration (second derivative), and jerk (third derivative) plots are displayed in the lower graph.

- If the element is a line, the velocity is constant and acceleration is 0.
- With the check boxes in the Curve selection table (Figure 16-13), you can select or clear each individual curve to be displayed.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- When a curve is selected (see blue highlighted row in Figure 16-13), the Y-scale of the Curves graph is adjusted to display the Y-scale of the selected curve.
- The color of the tick line of the scale is changed to match the color code of the selected curve.

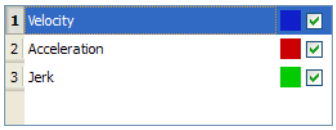


Figure 16-13: Curve Selection Table

12.1.5.1 Change the Color

1. Double-click on a colored square shown in the Curve Selection Table.
   The Select color dialog box opens. (Figure 16-14)

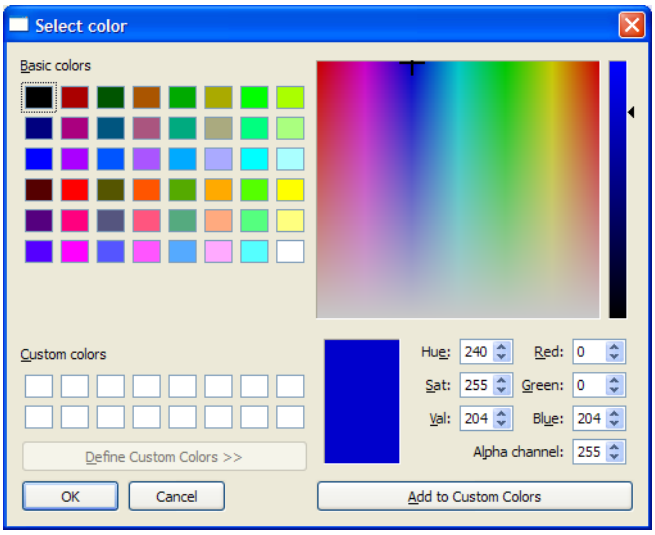


Figure 16-14: Select color dialog box

2. Click an existing color square to select it, or specify the numerical values for a color.
   You can move the black indicator on the right side until the desired color appears in the large colored rectangle.

12.1.5.2 Curves Graph

Velocity (the first derivative), acceleration (the second derivative), and jerk (the third derivative) curves are displayed in the lower graph.

All plots are displayed by default.

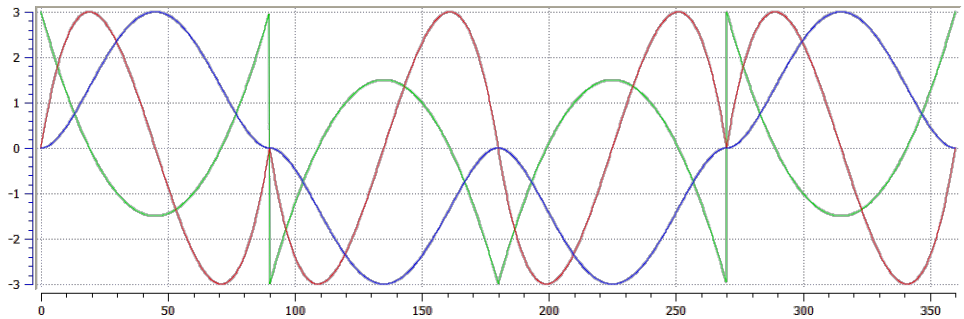


Figure 16-15: Example: Curves Graph

- With the check boxes in the Curve selection table shown in the "Curve Selection Table" (→ p. 705), you can select or clear each individual curve that you want to be displayed.

- The Y-scale of the Curves graph is adjusted to display the Y-scale of the selected curve in the Curve Selection Table.
- The color of the Y-axis scale tick lines is also changed to match the color code of the selected curve.

Y axis	Unit	Description
Vel	Units/Time	Being the rate of change of position, the velocity is the ratio between the slave and master derivatives
Accel	Units/Time^2	Rate of change of velocity with time
Jerk	Units/Time^3	Rate of change of acceleration; more precisely, the derivative of acceleration with respect to time

- The numbers relate to how the Y-axis positions (Cam Output) change with respect to the X-axis positions (CAMinput).
- The zoom and pan functions, when performed on the cam profile graph, are duplicated in the Curves graph.
  Zoom and pan functions are not available when the cursor is in the curves graph.

12.1.6 Reload, Save, Auto Fit, and Properties Buttons

These buttons are provided:

Icon	Description
	Reload the saved profile. If unsaved changes have been made to the profile, a dialog box asks you to confirm that you want to discard the changes.
	Save a modified profile.
	Adjusts the zoom and pan settings so that the entire graph is displayed in the graphical area.
	Open the Cam Profile Properties dialog box to modify the Master/Input and Slave/Output Offset and Scale values.

12.1.7 Import a CAM Profile

The KAS-IDE can import CAM profiles from several sources:

- CAM profiles (.CAM file format) exported from another KAS project.
- User-defined CAM profiles (in .CSV format).
  - When imported, the .CSV file is converted to a .CAM file.

Row	Syntax
1	`CYCLIC;YES;`
2	`TABLE_BEGIN;;`
3	`0;0;SPLINE`
4	`X;Y;SPLINE`
:	`X;Y;SPLINE`
N	`1000;1000;SPLINE`
N+1	`TABLE_END;;`

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Each row from 4 to N specifies the successive points that are part of the cam profile. The X and Y coordinates can be specified as floating-point values with sufficient digits after the decimal point (example: 995.2514255). To be valid, a CSV file must have **at least 4 spline segments** in it. See "CAM Profile Segments" (➔ p. 702) for more information on the segment types.

NOTE

Splines are the only segment type that is currently supported when importing CSV data into KAS-IDE. Other segment types are supported by importing and exporting .CAM files.

When a CSV file is imported the X, Y values are normalized with respect to maximum X, Y values present in the CSV file. The normalized X, Y values are scaled with respect to Master/Input scale and Slave/Output scale. They are added with Master/Input Offset and Slave /Output Offset respectively and will be displayed in the Specified points (Absolute) section of the cam profile.

Example:

CSV file X,Y Values:

```
0;0;SPLINE
100;111;SPLINE
200;222;SPLINE
300;333;SPLINE
```

Max Value in CSV is

```
300;333;SPLINE
```

Normalized values:

```
0; 0;
0.333333333333333; 0.333333333333333;
0.666666666666667; 0.666666666666667;
1;1;
```

Offset:

```
10      20
```

Scale:

```
300     360
```

Value displayed in profile:

```
10;20;
109.9999999999999; 139.99999999999989;
210.00000000000009; 260.00000000000011;
310;380;
```

12.1.7.1 The Import

A quick validation is performed when the profile is first imported.

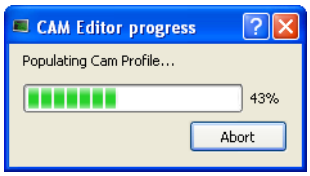
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The profile name is checked.
  - If it is in use, you are promoted to provide a new name.
- The data format is validated.
  - The profile is verified it can be compiled without error.

If cyclic is on, and the Vel/Accel values of the first element do not match the Vel/Accel of the last element, the first elements Vel/Accel are copied to the last elements. A warning message is posted to the log if this change takes place.

12.1.7.2 Displaying the Imported CAMProfile

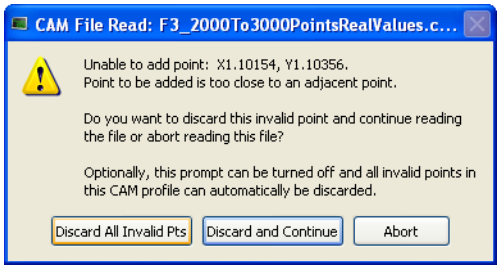
When you display an imported profile, a dialog box indicates the progression of the import process.



Click the **Abort** button to abort the process, then a default CAM profile is created.

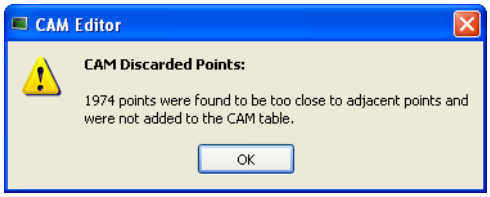
12.1.7.3 Invalid Data

When you display a CAM profile where two points are too close, a dialog box indicates the error.



Click the **Discard All Invalid Pts** button to discard all additional invalid points found in this CAM profile.

A summary appears when the process is finished.



12.2 Custom Input/Output Editor

NOTE

This tool is reserved for PROFIBUS fieldbuses only.

The Input/Output Editor (hereafter I/O Editor) is a tool used to declare and set up I/O devices, and establish the link between the application variables and physical equipment. It shows a list of the currently defined I/Os.

To open the I/O Editor select **Tools**>**Custom IO Editor** from the menubar.

Figure 16-16: Input/Output Editor

For the **Description** field, see format explanations.

12.2.1 Add an Input/Output

Drag-and-drop a variable from the dictionary to the I/O editor, then modify it.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

12.2.2 Delete Input/Output

1. Click somewhere on the I/O's row (or go to the row with the up/down arrow keys)
2. Press the delete key
3. Confirm the deletion.

12.2.3 Modify an Input/Output

1. Double-click the cell to edit.

   TIP

   You can also use the arrow keys to select the cell and press the **F2** key to start edition.
2. Set its driver name to the one of your choice, for example: **CIFDriver** (column 3).
3. Set its description to the corresponding driver address (column 4).

The **Description** field format has these characteristics:

1. It begins with a "**%**" character.
2. Followed by the type of I/O.

   I: input, **Q**: Output
3. Followed by the size of I/O.

   X: Boolean (1 bit), **B**: byte (8 bits), **W**: word (16 bits), **D**: double word (32 bits), **L**: long word (64 bits)
4. Followed by its address on the selected bus.

   The address has the following format: "**deviceId.slaveId.moduleId.bitOffset**", where deviceId, slaveId, moduleId and bitOffset are integers ranging from 0 to 65535.

   NOTE

   - set **deviceId** to 0
   - set **slaveId** to the id of the I/O node
   - set **moduleId** to the id of the slice
   - **bitOffset** must always be 0 for non-Boolean I/Os

IMPORTANT

The size of the variable and the I/O must be the same.

Example

%IX0.1.2.4 is an input Boolean located on deviceId=0, slaveId=1, moduleId=2 at bitOffset=4

%QB0.1.2.0 is an output byte located on deviceId=0, slaveId=1, moduleId=2

TIP

If you enter invalid text, the table cell becomes red and an explanation appears in the **information** column.

See Also

"Map Input and Output to Variables" (→ p. 526)

12.3 Human-Machine Interface Editor

This section covers these tools you can use to design your HMI panels:

- The Kollmorgen Visualization Builder to control your application
- The internal Control Panel editor to debug your application with the KAS Simulator

12.3.1 Use the Kollmorgen Visualization Builder (KVB)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



To work with Kollmorgen Visualization Builder:

1. Tag the PLC variables to export and map with the HMI (see "Map Variables to HMI" (→ p. 216) for more information).
2. Compile your project to generate the Modbus mapping file.
3. Create a KVB project [1] in the KAS-IDE and open it.
4. "Design the HMI Panel with KVB" (→ p. 714) .
5. Save and close KVB.

   NOTE

   Important! Be sure to use **Save** and not **Save As**.
   The KVB is self-contained within the KAS archive and the Save As function moves the KVB out of the archive.
6. Save your KAS project.

NOTE

When you create the KVB panel with the KAS-IDE, all the creation and mapping procedure is done automatically after compiling your project.
You can directly go to "Design the HMI Panel with KVB" (→ p. 714).

IMPORTANT

Be aware that as soon as you change the PLC variables exported for the HMI, the mapping file must be re-imported in Kollmorgen Visualization Builder to have an up-to-date version.

Related Documents

For further information on Kollmorgen Visualization Builder, see this documentation:

KVB Manuals	Description
Quick Start Guide	The Quick Start covers these important points to: • Install and use Kollmorgen Visualization Builder. • Configure HMI Panels and PC operated control applications.
User Manual	Contains all the content to help with Kollmorgen Visualization Builder.

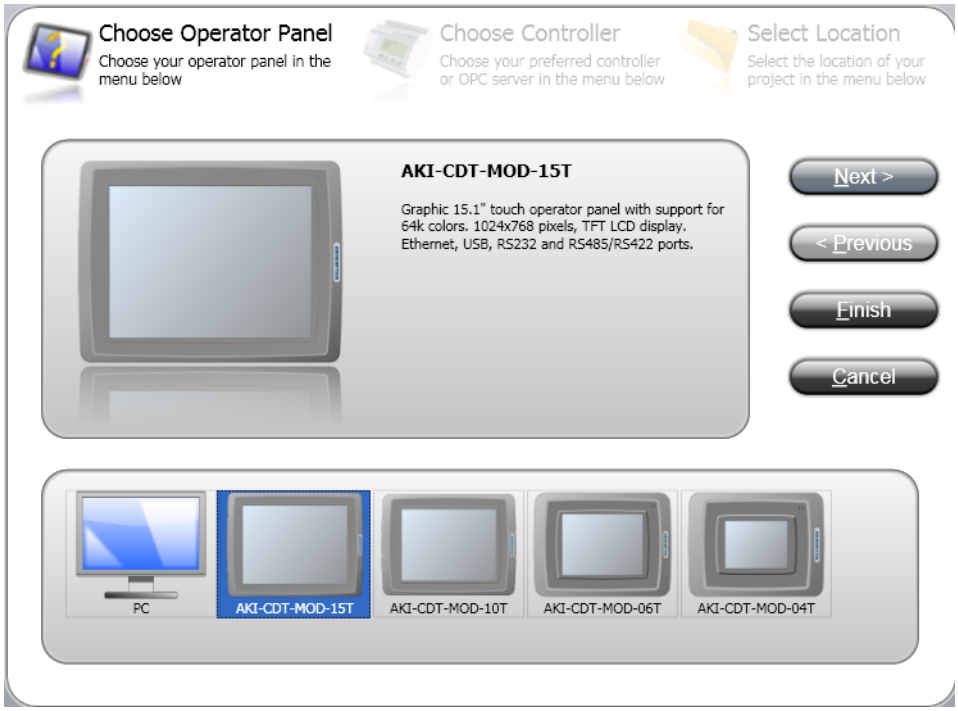
12.3.1.1 Create a New KVB Controller

[1] There is no built-in feature to import/export KVB projects.

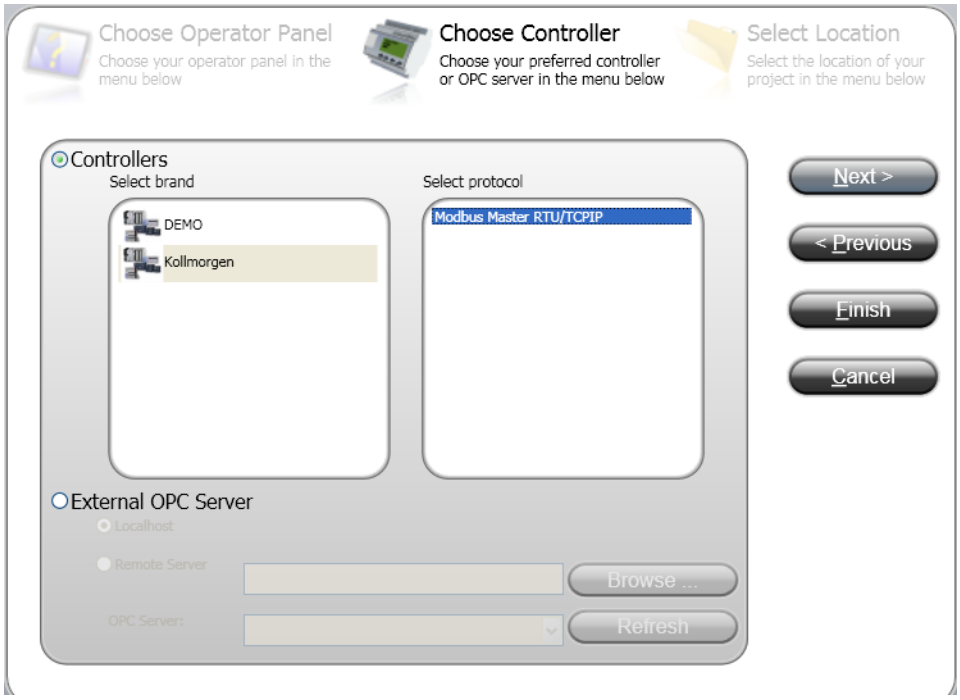
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This procedure is applicable when you use Kollmorgen Visualization Builder externally.

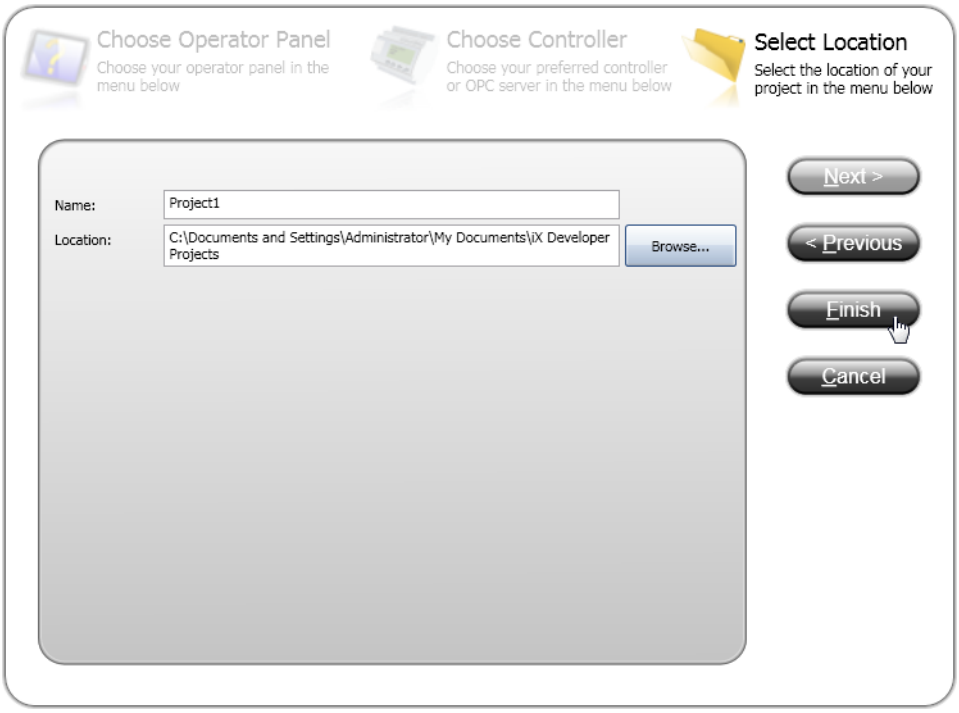
- After choosing to create a new project, select the type of operator panel to be used



- On the next dialog, select the **Kollmorgen** controller with the Modbus protocol, then click the **Next** button



- Enter the name of the project and where you want to create the project. Then click the **Finish** button



12.3.1.2 Import Variables into the Project

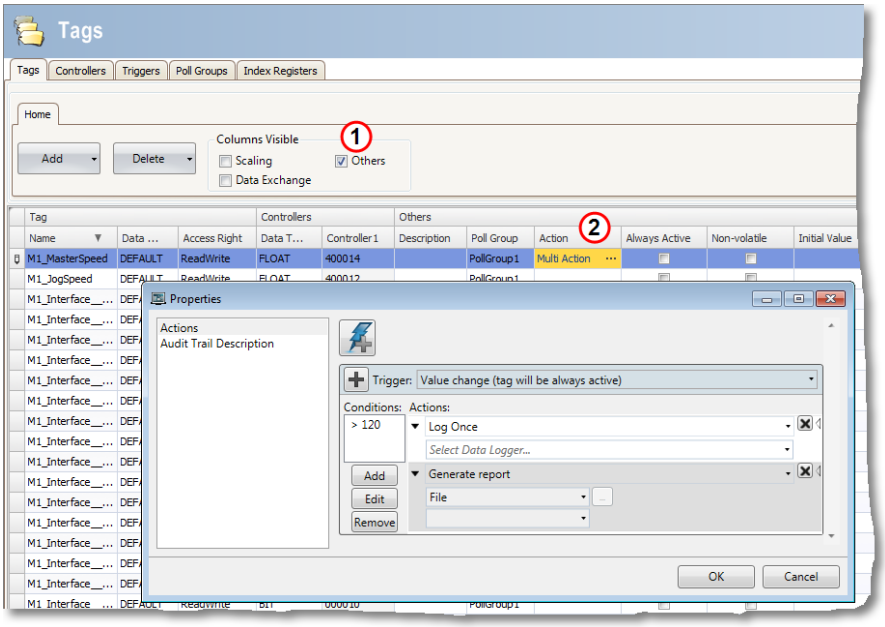
When you open the Kollmorgen Visualization Builder with your KVB panel (by double-clicking the KVB panel from the project explorer), all the variables tagged into the Dictionary at the time of compiling are automatically imported into Kollmorgen Visualization Builder.

Once the file is imported, all PLC variables are available for use within Kollmorgen Visualization Builder.

TIP

To reimport variables, (if they were changed or more were added), simply ensure they are tagged as "KVB" in the Variables list, recompile your KAS project, and open the KVB from the KAS-IDE.

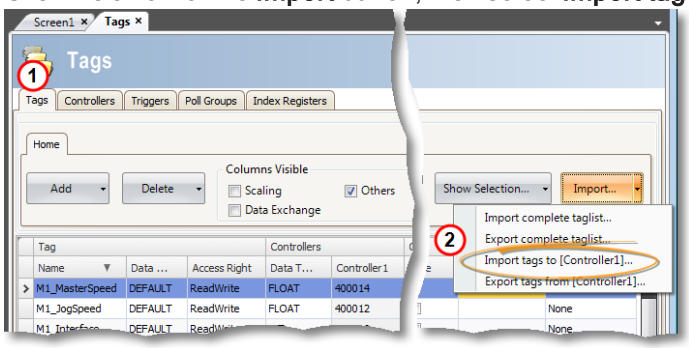
1. Select **Others** to display the Action column
2. You can edit the tag actions



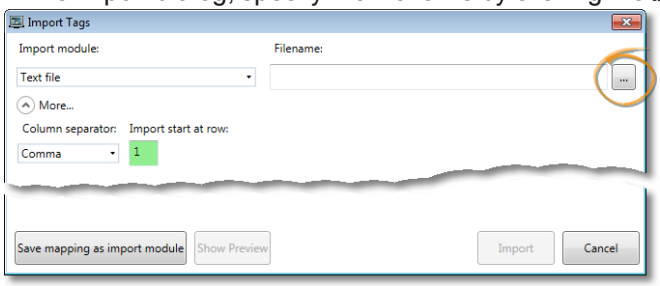
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

To manually export/import the variables (tags) of your project use the import procedure as follows:

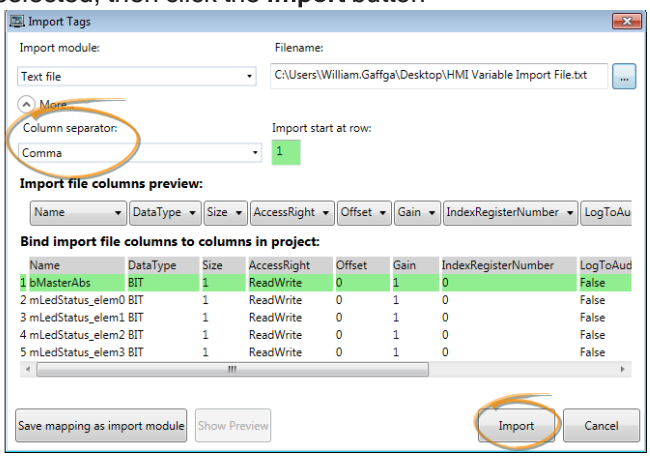
- Select the **Tags** tab
- Click the arrow of the **Import** button, then select **Import tags to [Controller1]...** in the drop-down menu



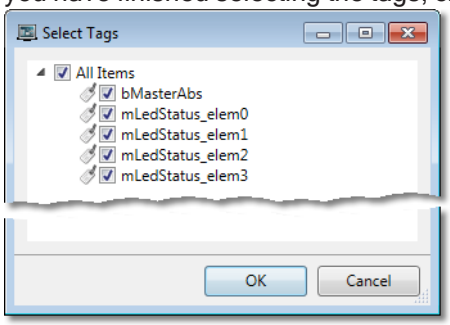
- In the import dialog, specify the filename by clicking the **...** button



- Then use the open file dialog to find the **.txt** file
- Once the file is specified, click the **Open** button
- Back in the import dialog, make sure the **Column separator** is set for Comma, and leave all options selected, then click the **Import** button

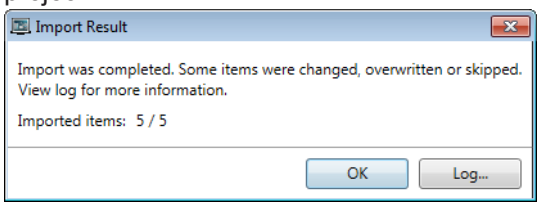


- Specify which tags (variables) you want to import. To select all tags, click the **Select All** option. When you have finished selecting the tags, click the **OK** button



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

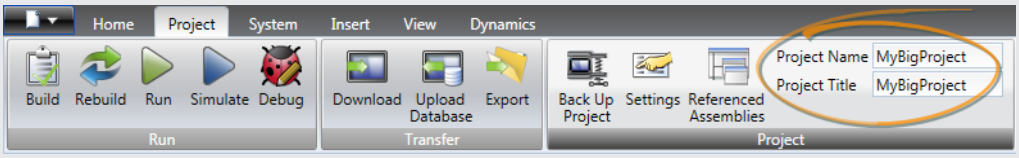
- You are notified of how many items are successfully imported. Click the **OK** button to return to the project.



12.3.1.3 Design the HMI Panel with KVB

IMPORTANT

Do not modify **Project Name** and **Title** to keep consistency between Kollmorgen Visualization Builder (KVB) and the KAS-IDE.



12.3.1.3.1 Add Object

You can drag-and-drop predefined objects from the library to the screen. The library is located in the **Home** tab of Kollmorgen Visualization Builder.

12.3.1.3.2 Customize Object

Select an object and click the **General** tab to customize:

- its settings in the Settings section
- its style to a different template in the Style section

12.3.1.3.3 Map Variable to the Object

In the **General** tab, you can set the Variable or Tag that maps to the current object in the Tag/Security section.

NOTE

Click the **F1** key to open the Kollmorgen Visualization Builder online help (or use the Help button in the ribbon tab heading)

IMPORTANT

Be aware that as soon as you change the PLC variables exported for the HMI, the mapping file must be re-imported in Kollmorgen Visualization Builder to have an up-to-date version.

12.3.1.4 Download the KVB Panel

To download your panel you have to use the Project ribbon in Kollmorgen Visualization Builder (KVB) that contains the **Transfer** command.

- To download, you must be connected directly to the HMI panel through an Ethernet cable.
- The IP address is already defined so nothing special has to be done before transferring your panel to the graphic operator terminal.
- See "Configure the Controller" (→ p. 156).



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Button	Description
Download	The Download control sends the project to an HMI-panel, or a controller with a Kollmorgen Visualizer RT installation, in the network environment.
Upload Database	The Upload Database command collects the database from an HMI-panel in the network environment to a file on the development PC. The panel pauses during the upload and is started automatically afterwards.
Export	The Export command builds and exports a copy of the project. All needed files to transfer the project to another location.

NOTE

- If you transfer your project on a USB stick, place it in the USB port of the AKI panel while it is booting up.
- See the online help in Kollmorgen Visualization Builder.

12.3.2 Design the HMI with the Internal Editor

This section details the Controls and Properties used to:

- Define the Control Panel to debug the application.
- "Map Variables to the Control Panel" (→ p. 719).

12.3.2.1 Create a Control Panel

Control Panels are created and managed in the KAS-IDE Project Explorer.

1. In the Project Explorer tree, right-click the Controller and select **Add Control Panel**. (Figure 16-17)

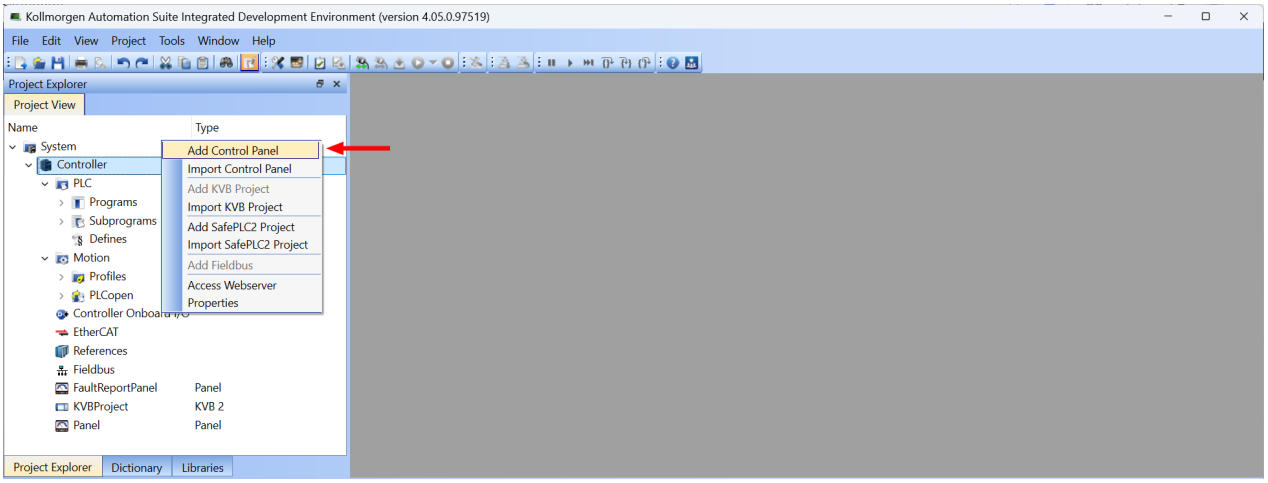


Figure 16-17: Add Control Panel option

2. Right-click the new Panel and select **Rename**. (Figure 16-18)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

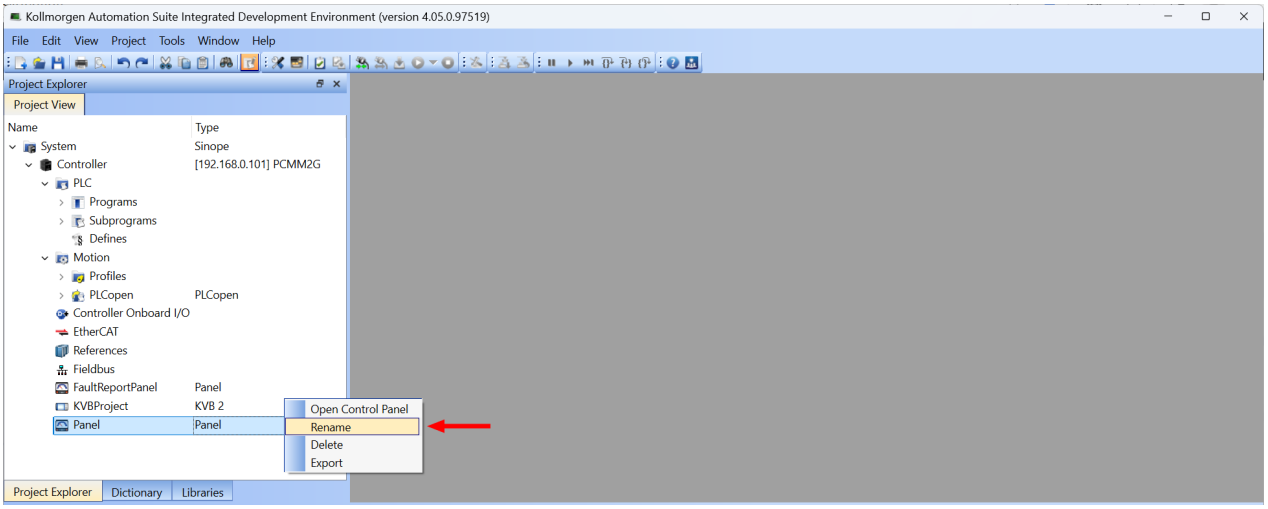


Figure 16-18: Rename option

The **Rename**dialog box opens.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

3. Change the Panel name. (Figure 16-19)

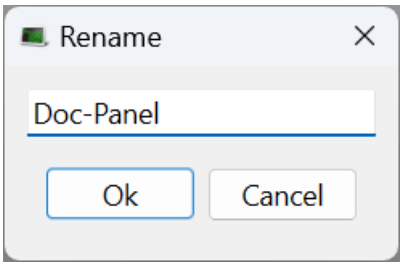


Figure 16-19: Rename dialog box with changed name

4. Click **OK** to save the changes or selections and close the dialog box.
5. In the Project Explorer tree, double-click the new Control Panel to open it in the KAS-IDE graphical editor. (Figure 16-20)

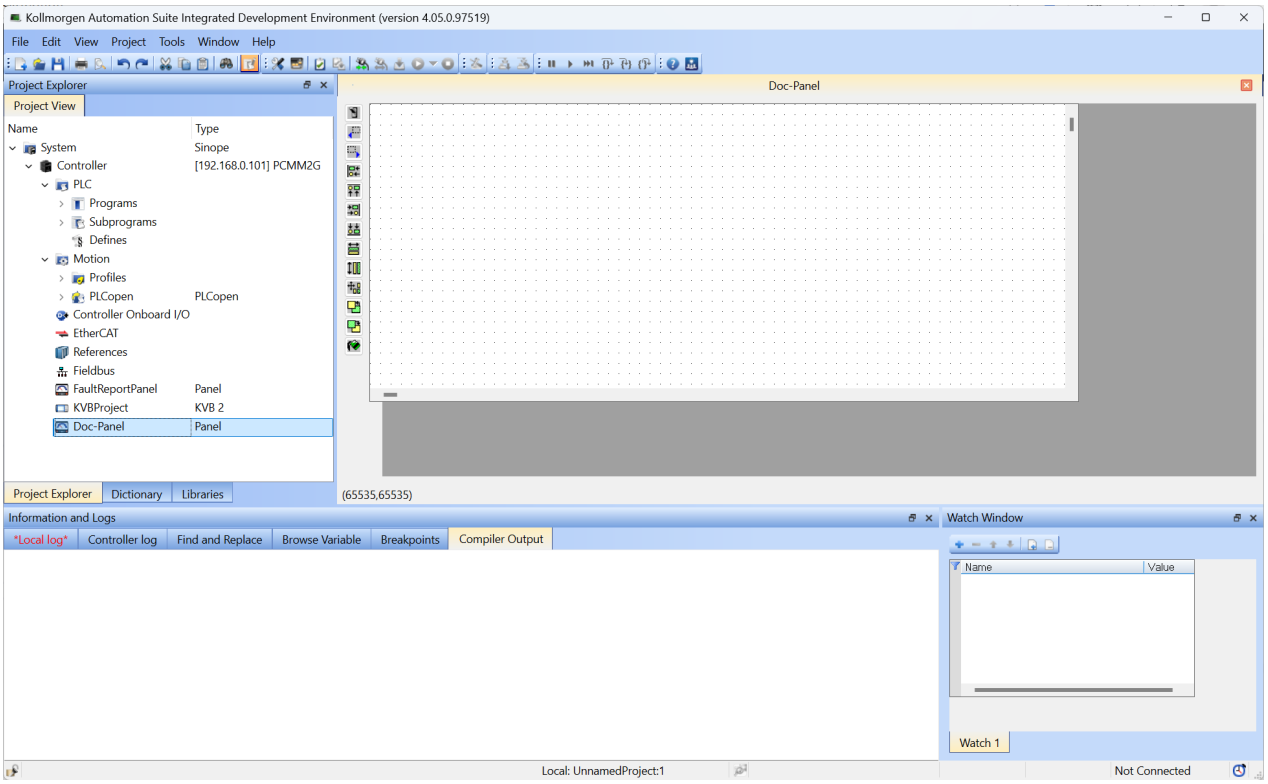


Figure 16-20: New Control Panel

12.3.2.2 Use the Control Panel Control Library

- Select a control in the Libraries toolbox (Controls tab) and drag-and-drop it in your Control Panel.

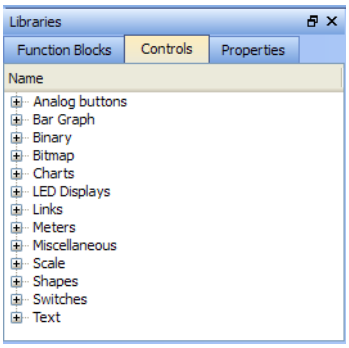


Figure 16-21: Control Panel Control Library

For a list of controls, see "Graphic Objects" (→ p. 720).

12.3.2.3 Edit the Control Panel

When a control is selected, you can change its properties (displayed in the Libraries toolbox) by double-clicking the **Value**.

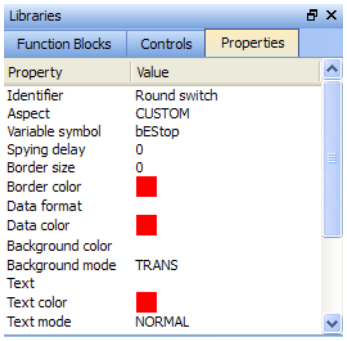


Figure 16-22: Control Panel Control Properties

See "Graphic Objects Properties" (→ p. 727).

NOTE

- You can perform multi-selection with the mouse (all the controls that are even partly inside the selection area are selected).
- Add controls to your selection either with the **Ctrl** or **Shift** keys.
- Use the arrow keys to move the Control Panel page up, down, and sideways.
- Use the <Shift + Arrow> keys to move the selected Control up, down, and sideways.

TIP

To duplicate all the selection, hold down **Ctrl** and click the right mouse button while performing your move operation (remember to release the mouse button first, before the **Ctrl** key).

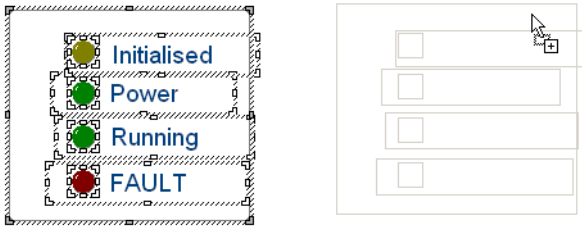


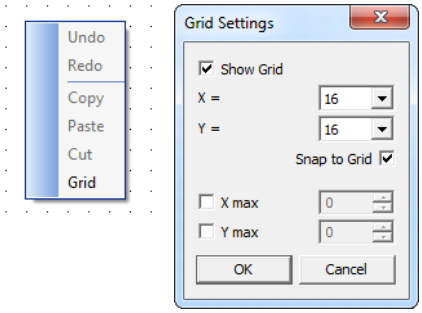
Figure 16-23: Control Panel - Selection of Controls

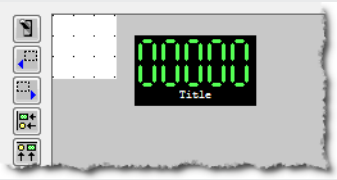
12.3.2.3.1 HMI Grid Settings

Right-clicking in the Control Panel's graphical editor provides access to the Grid Settings by selecting Grid.

- This may also accessed by pressing <Ctrl+G>.
- The settings allow you to control the appearance of the grid as well as forcing objects to snap to the grid.
- Settings are per panel and are saved with the project.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



Parameter	Description
Show Grid	Display or hide the grid in the Control Panel.
X= / Y=	Space between each grid dot in X/Y directions.
Snap to Grid	Controls are forcibly aligned with the grid when they are placed/moved.
X max / Y max	Set a maximum area in which the grid is shown. The editor is grayed out beyond these dimensions. Objects may still be placed within the grayed out area as it is merely a visual aid. 

12.3.2.4 Map Variables to the Control Panel

To link your Control panel with the PLC programs, some controls contain a property called **Variable symbol**. (Figure 16-24)

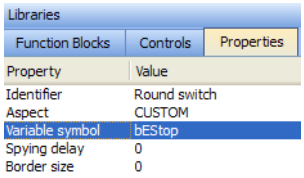


Figure 16-24: Map variables to a Control Panel control

Procedure

1. Select the variable in the Dictionary toolbox.

2. Move it to the control to be linked in the Control panel editor using drag-and-drop. (Figure 16-25)

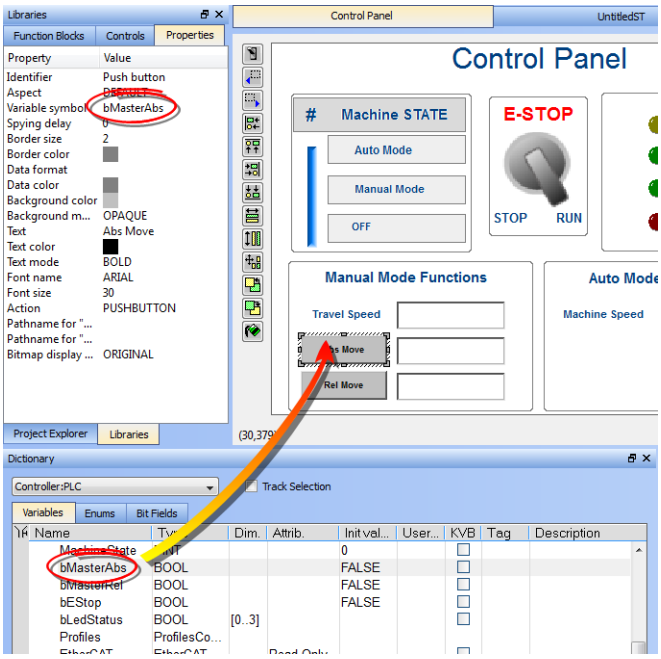


Figure 16-25: Map Variables to a Control Panel Control in the Graphical Editor

The Variable symbol is automatically updated in the **Properties** tab.

NOTE

A warning is generated when the program is compiled if the control is mapped to a variable not in the Dictionary. Double-click the warning to highlight the control object and open the editor so the variable can be defined.

12.3.2.5 Graphic Objects

These are the available basic objects you can insert in your graphics:

12.3.2.5.1 Basic Shapes



A collection of basic drawings is available. Each object can be either static, or linked to a variable used to enable its visibility (show/hide).

Properties

- "Aspect (shapes)" (→ p. 728)
- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)
- "Color when not connected" (→ p. 730)
- "Data format" (→ p. 730)
- "Direction (basic shapes)" (→ p. 731)
- "FALSE color" (→ p. 731)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Spying delay" (→ p. 731)
- "Text" (→ p. 732)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)
- "TRUE color" (→ p. 732)
- "Variable symbol" (→ p. 732)

12.3.2.5.2 Bitmaps

Bitmap file (BMP, GIF, JPG) can be inserted in the graphic area.

Properties:

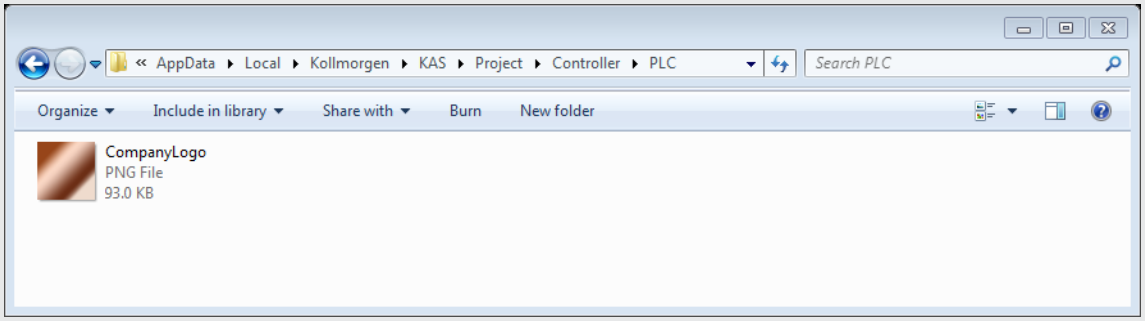
- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Bitmap display mode" (→ p. 729)
- "Bitmap pathname" (→ p. 730)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)
- "Border style" (→ p. 730)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)

NOTE

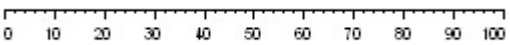
Large bitmaps are time-consuming during animation and can lead to poor performance, mainly if they have the "STRETCH" display mode or the "TRANS" (transparent) background mode.

TIP

Graphics need to be made portable with a project, otherwise the KAS-IDE attempts to point to the original location of the graphic files.
To make graphics portable, copy them to `C:\Users\[user folder]\AppData\Local\Kollmorgen\Project\Controller\PLC`.
After this the graphics are a part of the .KAS file.



12.3.2.5.3 Scales



Scales are static drawings representing an X or Y axis, generally used to document other objects such as trend charts or bargraphs.

Properties:

- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Border style" (→ p. 730)
- "Color when not connected" (→ p. 730)
- "Direction (scale)" (→ p. 731)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Maximum value" (→ p. 731)
- "Minimum value" (→ p. 731)
- "Nb divisions (main)" (→ p. 731)
- "Nb divisions (small)" (→ p. 731)
- "Placement (scale)" (→ p. 731)
- "Scale color" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)

12.3.2.5.4 Text boxes



Static, animated or edit text boxes are available for displaying / forcing variables. For edit boxes at runtime, double-click on the object to enter the value and then hit ENTER to validate the input.

Properties:

- "Action (text)" (→ p. 728)
- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)
- "Border style" (→ p. 730)
- "Data format" (→ p. 730)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Spying delay" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)
- "Variable symbol" (→ p. 732)

12.3.2.5.5 Switches and 2-state displays



Buttons, switches and 2-state displays are used for control or display of a Boolean variable.

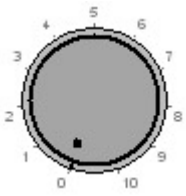
Properties:

- "Action (switch)" (→ p. 728)
- "Aspect (switches)" (→ p. 729)
- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Bitmap display mode" (→ p. 729)
- "Bitmap for "FALSE" state" (→ p. 729)
- "Bitmap for "TRUE" state" (→ p. 729)
- "Border color" (→ p. 730)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Border size" (→ p. 730)
- "Data color" (→ p. 730)
- "Data format" (→ p. 730)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Spying delay" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)
- "Variable symbol" (→ p. 732)

12.3.2.5.6 Analog buttons



Analog buttons are used for setting the value of an integer or real variable. The mouse is used for setting the value.

Properties:

- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)
- "Border style" (→ p. 730)
- "Data color" (→ p. 730)
- "Data format" (→ p. 730)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Maximum value" (→ p. 731)
- "Minimum value" (→ p. 731)
- "Scale color" (→ p. 731)
- "Spying delay" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)
- "Variable symbol" (→ p. 732)

12.3.2.5.7 Bar Graphs



Bargraphs are rectangles filled according to the value of an analog variable. Bargraphs can be horizontal or vertical.

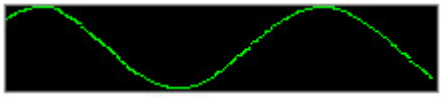
Properties:

- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)
- "Border style" (→ p. 730)
- "Data color" (→ p. 730)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Data format" (→ p. 730)
- "Direction (bar graph)" (→ p. 730)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Maximum value" (→ p. 731)
- "Minimum value" (→ p. 731)
- "Spying delay" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)
- "Variable symbol" (→ p. 732)

12.3.2.5.8 Charts

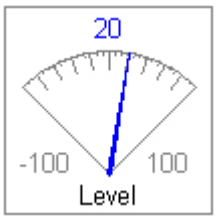


Charts enable the tracing of a variable as with an oscilloscope.

Properties:

- "Aspect (trend charts)" (→ p. 729)
- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)
- "Border style" (→ p. 730)
- "Data color" (→ p. 730)
- "Data format" (→ p. 730)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Maximum value" (→ p. 731)
- "Minimum value" (→ p. 731)
- "Nb of points (trends)" (→ p. 731)
- "Spying delay" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)
- "Variable symbol" (→ p. 732)

12.3.2.5.9 Analog meters



Analog meters provide a graphical display of an analog value.

Properties:

- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)
- "Border style" (→ p. 730)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Data color" (→ p. 730)
- "Data format" (→ p. 730)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Link" (→ p. 731)
- "Maximum value" (→ p. 731)
- "Minimum value" (→ p. 731)
- "Nb divisions (main)" (→ p. 731)
- "Nb divisions (small)" (→ p. 731)
- "Scale color" (→ p. 731)
- "Spying delay" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)
- "Variable symbol" (→ p. 732)

Sliders



Sliders are used for entering an analog value with a horizontal or vertical mouse driven cursor.

Properties:

- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)
- "Border style" (→ p. 730)
- "Data color" (→ p. 730)
- "Data format" (→ p. 730)
- "Direction (slider)" (→ p. 731)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Maximum value" (→ p. 731)
- "Minimum value" (→ p. 731)
- "Scale color" (→ p. 731)
- "Spying delay" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)
- "Variable symbol" (→ p. 732)

12.3.2.5.10 Digital meters



Digital meters (digits) display the value of a variable with the same aspect as a digital clock.

Properties:

- "Aspect (digits)" (→ p. 728)
- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Border style" (→ p. 730)
- "Data color" (→ p. 730)
- "Data format" (→ p. 730)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Maximum value" (→ p. 731)
- "Minimum value" (→ p. 731)
- "Spying delay" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)
- "Variable symbol" (→ p. 732)

12.3.2.5.11 Links



Links are mouse-driven hyperlinks that are used as shortcuts to open another graphic document. Using links enables the design of multi-page animated applications.

Properties:

- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)
- "Border style" (→ p. 730)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Link" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)

12.3.2.5.12 Connection status

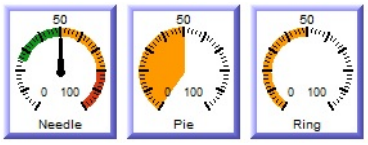
Connection status is a box actuated with the current status of the connection and the connected runtime application. It is mainly dedicated to diagnostic.

Properties:

- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)
- "Border style" (→ p. 730)
- "Data color" (→ p. 730)
- "Data format" (→ p. 730)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Spying delay" (→ p. 731)
- "Text" (→ p. 732)
- "Text color" (→ p. 732)
- "Text mode" (→ p. 732)

12.3.2.5.13 Gauges

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



Analog view meter.

Properties:

- "Background color" (→ p. 729)
- "Background mode" (→ p. 729)
- "Border color" (→ p. 730)
- "Border size" (→ p. 730)
- "Border style" (→ p. 730)
- "Data color" (→ p. 730)
- "Data format" (→ p. 730)
- "Font name" (→ p. 731)
- "Font size" (→ p. 731)
- "Identifier" (→ p. 731)
- "Maximum value" (→ p. 731)
- "Minimum value" (→ p. 731)
- "Nb divisions (main)" (→ p. 731)
- "Nb divisions (small)" (→ p. 731)
- "Needle color" (→ p. 731)
- "Spying delay" (→ p. 731)
- "Title" (→ p. 732)
- "Title color" (→ p. 732)
- "Title placement" (→ p. 732)
- "Variable symbol" (→ p. 732)

Other Properties:

- Needle aspect: defines how the dynamic area must be shown (see bitmaps)
- Background bitmap: optional bitmap to be displayed as background
- Needle X/Y position: start position from the center of the image (percents)
- Needle size: needle with (percents)
- Needle angle: angles for min/max values in degrees
- Green color: color for the "low" range
- Green to: maximum "low" value
- Yellow color: color for the "medium" range
- Yellow to: maximum "medium value
- Red color: color for the "high" range
- Red to: maximum "high" value
- Value placement: where the text value should be drawn
- Set Gain/offset

12.3.2.6 Graphic Objects Properties

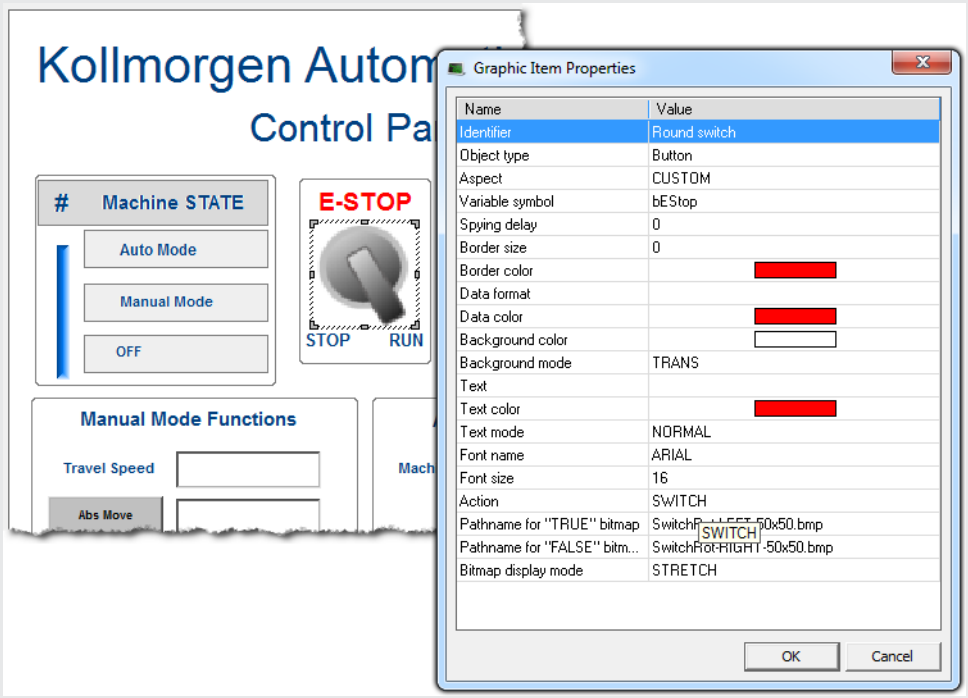
This page details all possible properties for graphic objects. Refer to the list of available objects for further information on which property is used for which object.

TIP

Double-click an object to open the **Graphic Item Properties** dialog box.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This dialog lists the properties particular to that object.



Name	Description
Action (switch)	Indicates the possible mouse action for switches. Possible values are: • STATIC = no mouse action. • PUSHBUTTON = the variable is forced to TRUE when pressed and FALSE when released. • SWITCH = the status of the variable is inverted when the button is pressed. • ONESHOTBUTTON = same as switch, but the display continues to appear released.
Action (text)	Indicates the possible mouse actions for text boxes. Possible values are: • STATIC = no mouse action. • EDIT = double-click opens an edit box for entering the variable value.
Aspect (digits)	Indicates the type of drawing for a digital meter. Possible aspects are: • DEFAULT = plain drawing. • BEZEL = all segments have a 3D effect.
Aspect (shapes)	Indicates the type of basic shape to be drawn. Possible aspects are: • CYLINDER = a 3D like cylinder. • ELLIPSE = an ellipse. • GATE = a simple vector drawing for a valve. • HALFELLIPSE = one half of an ellipse. • RECTANGLE = a rectangle. • ROUNDRECT = a rectangle with rounded corners. • TRIANGLE = a triangle

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Name	Description
Aspect (switches)	Indicates the type of switch to be drawn. Possible aspects are: • DEFAULT = a standard Windows-like push button. • CUSTOM = a button with TRUE and FALSE drawings defined with bitmaps.
Aspect (trend charts)	This property indicates the type of drawing for a trend chart. Possible aspects are: • POINT = only relevant dots are drawn. • LINE = lines are drawn from point to point. • HISTO = histogram style.
Background color	Indicates the color used for filling the background of the object. In case of a bitmap, it specifies the color that **must not** be drawn if the TRANS (transparent) background mode is specified.
Background mode	Indicates whether the background of the object must be filled or not. • If this property is: • OPAQUE, the background is filled with the specified background color. • TRANS (transparent), the background is not filled. • Transparent drawing mode can be useful in the case of overlapping objects. **IMPORTANT** Specifying the TRANS (transparent) mode for large bitmaps is time-consuming and affects the real-time performances of graphic updates.
Bitmap display mode	For bitmap-based objects, this indicates whether the attached bitmap must keep its original aspect or be stretched to the actual size of the object. Possible values are: • ORIGINAL = keep the original aspect of the bitmap (cut if too large). • STRETCH = stretch or shrink the bitmap for fitting the actual size of the graphic object. **IMPORTANT** Large bitmaps with STRETCH display mode are time-consuming during animation and can lead to poor performance.
Bitmap for "FALSE" state	For two-state objects with the CUSTOM aspect, this specifies the pathname of the bitmap to display when the value of the attached variable is FALSE (or zero for analogs). • BMP, GIF, and JPG formats are supported. • If no directory is specified, the specified file name is searched: • in the project folder. • in the \BITMAP\ folder of the KAS-IDE.
Bitmap for "TRUE" state	For two-state objects with the CUSTOM aspect, this specifies the pathname of the bitmap to display when the value of the attached variable is TRUE (or not zero for analogs). • BMP, GIF, and JPG formats are supported. • If no directory is specified, the specified file name is searched: • in the project folder. • in the \BITMAP\ folder of the KAS-IDE.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Bitmap pathname</td><td>For bitmaps, this specifies the pathname of the bitmap to be displayed.
BMP, GIF, and JPG formats are supported.
If no directory is specified, the specified file name is searched:
in the project folder.
in the \BITMAP\ folder of the KAS-IDE.</td></tr>
<tr><td>Border color</td><td>Indicates the color of the border drawn around the object.</td></tr>
<tr><td>Border size</td><td>Indicates the width of the border drawn around the object, expressed as a number of pixels.
If this property is 0 (zero), no border is drawn.</td></tr>
<tr><td>Border style</td><td>Indicates the possible 3D effect used for drawing the border around the object.
Possible values are:
3D = default 3D effect
3DDOWN = pressed 3D effect.
3DUP = depressed 3D effect.
FLAT = no 3D effect.</td></tr>
<tr><td>Color when not connected</td><td>For shapes, this indicates the color used for filling shapes when no variable is attached to the graphic object.</td></tr>
<tr><td>Data color</td><td>Indicates the color used to represent the value of a connected variable within the object (e.g., the filled part of a bar graph).</td></tr>
<tr><td>Data format</td><td>If defined, this indicates that the value of the connected variable must be displayed on the graphic object.
This property requires a specified format string that indicates how the data is to be formatted.
IMPORTANT
The "text" property is ignored when a data format is specified.
Format string has the same format as the printf function of "C" language.
It can include static characters together with one of these possible pragmas that specify the value:
%s = default formatting according to IEC syntax
%d = integer (decimal)
%X = hexadecimal
%g = floating point
%.nf = decimal real (where n is the number of displayed decimal digits)
Examples<table><tr><th>Format</th><th>Value</th><th>Displayed String</th></tr><tr><td>%d</td><td>12.3</td><td>12</td></tr><tr><td>Var = %g meters</td><td>1.2</td><td>Var = 1.2 meters</td></tr><tr><td>%.2f</td><td>1.12345</td><td>1.12</td></tr></table>NOTE
Only one % pragma can be used in a string.</td></tr>
<tr><td>Direction (bar graph)</td><td>For bar graphs, this indicates the growing direction: to the left, right, top, or bottom.</td></tr>
</table>

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Name	Description
Direction (basic shapes)	For oriented shapes such as triangles, half ellipses or cylinders, this indicates the direction of the drawing: to the left, right, top, or bottom.
Direction (scale)	For scales, this indicates the direction of the axis. • If LEFT, the minimum value is on the left side. • If RIGHT, the minimum value is on the right side.
Direction (slider)	For slider, this indicates whether the slider is horizontal (RIGHT) or vertical (TOP).
FALSE color	For shapes, this indicates the color used for filling shapes when the attached variable has the FALSE state, or zero for analogs.
Font name	Indicates the name of the character font used for drawing texts in the graphic object.
Font size	Indicates the size of the character font used for drawing texts in the graphic object. • The size is expressed as a percentage of the actual height of the object. • Maximum possible value is 100. • This ensures that the ratio is kept when the object is resized.
Identifier	Freely attach a text identifier to each graphic object inserted in a document. Identifiers are useful for arranging overlapped objects as they appear in the Z-order list.
Link	Indicates the name of the target . • GRA animated document for shortcuts. • If no directory is specified in the link, the file is searched in the project folder.
Maximum value	For analog animated objects (e.g., meters, bar graphs, or trends) this indicates the maximum possible value that can be displayed. For static scales, it indicates the value of the highest mark.
Minimum value	For analog animated objects (e.g., meters, bar graphs or trends) this indicates the minimum possible value that can be displayed. For static scales, it indicates the value of the lowest mark.
Nb divisions (main)	For objects including a graphic scale, this indicates the number of main division marks to be drawn in the scale.
Nb divisions (small)	For objects including a graphic scale, this indicates the number of small division marks to be drawn in the scale, between each main division mark.
Nb of points (trends)	For trend charts, this indicates the maximum number of stored points. If the width of the object (in pixels) is less than this number, then oldest points are not visible.
Needle color	For gauges, this is the color of the needle.
Placement (scale)	For scales, this indicates the location of the scale within the object rectangle: on the left, right, top, or bottom.
Scale color	For objects including a graphic scale, this indicates the color used for drawing the axis, the division marks, and corresponding values of the scale.
Spying delay	It is the minimum period for actuating the value of the connected variable, expressed as a number of milliseconds. If the delay is not specified or equal to 0 (zero), refresh is done as fast as possible.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Name	Description
Text	If defined, this indicates the text displayed on the graphic object. **IMPORTANT** This property is ignored when a **data format** is specified.
Text color	Indicates the color used for inserting texts in the graphic object.
Text mode	Indicates the font effect used for drawing texts in the graphic object. Possible values are: • BOLD = bold text. • HIDE = text is not displayed. • ITALIC = italic text. • NORMAL = normal font. • UNDERLINE = underlined text
Title	For gauges, this is the text shown in the tile
Title color	For gauges, this is the color of the title text.
Title placement	For gauges, this is the position of the title within the gauge.
TRUE color	For shapes, this indicates the color used for filling shapes when the attached variable has the TRUE state, or non zero for analogs.
Variable symbol	This is the full name of the application variable connected to the graphic object. In case of a local variable, its symbol must be prefixed with the parent program name, separated with "/". Example: MyProg/MyVar

12.3.2.7 Operate the Control Panel

The Example program has a default control panel built-in to make it easy to start an application.

Use this procedure to operate the control panel:

1. Double-click **Control Panel** in the Project Explorer to open the form.

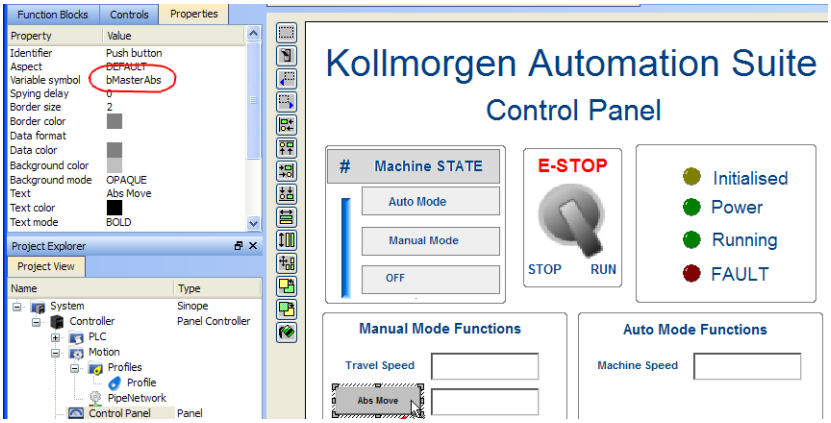


Figure 16-26: Control Panel

2. Move the vertical slider bar to select the Machine STATE as **Manual Mode**.
3. In the **Manual Mode Functions** area, double-click the text box for the **Travel Speed**.
4. Enter the numeric value for the **Travel Speed** and press <Enter>.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

12.3.2.7.1 KAS Simulator Display

The KAS Simulator displays the status and position of the axes and the log messages.

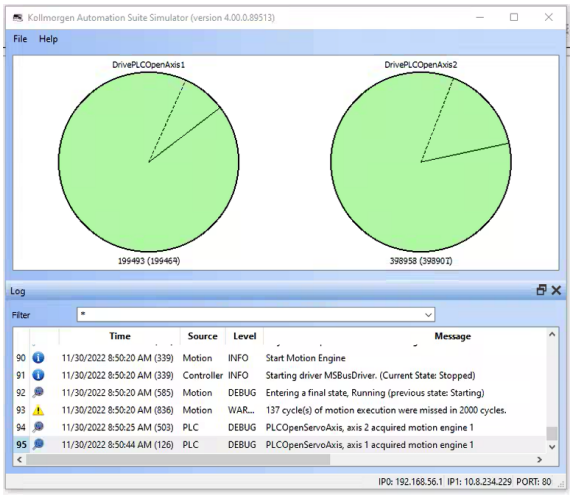


Figure 16-27: Example: Log Messages

Continue to use the Control Panel to:

- Experiment with the controls and observe the simulated output.
- Perform an absolute move by entering a position in the text box.
- Perform a relative move.

12.3.2.7.2 Exiting Simulation Mode

To exit Simulation mode, do as follows:

1. Click the **Stop Device** button .
2. Click the **Disconnect Device** button .

NOTE

For additional information about Kollmorgen Automation Suite, see:

- Online Help
- Getting Started
- Technical Reference - PLC Library
- Technical Reference - Motion Library
- User Manual

12.4 Pipe Network Editor

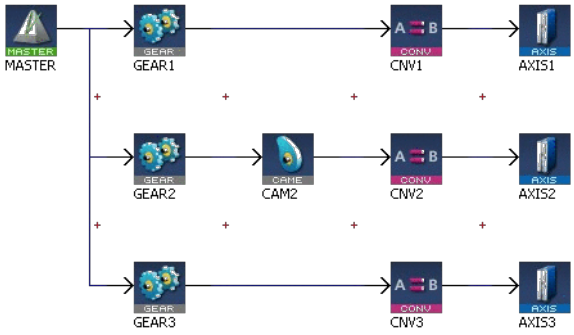


Figure 16-28: Pipe Network Structure

The Pipe Network Editor is a tool dedicated to the description of the motion part of the application ("Pipe Network Concept" (➔ p. 326)).

Functions of the Pipe Network Editor are accessed via context sensitive menus.

When the Pipe Network Editor is used, an ST file containing all the calls to the Motion Library is automatically generated during compilation, and based on the graphical description of the Pipe Network.

Pipe Network Editor is optional

Although strongly recommended, the Pipe Network Editor is optional.

Use it to graphically create a Pipe Network or to manually instantiate Pipe and Pipe Blocks by calling the appropriate functions in the Pipe Library directly from the IEC 61131-3 editors (SFC, FBD, ST, IL, FFLD).

Grid

The layout of the editor is grid oriented, which means that items (except the comments) are placed in the middle of a rectangular area called a grid unit.

NOTE

Comments are not centered in the grid unit but merely placed at the cursor position.

12.4.1 Insert Pipe Blocks or Comments

Right-click on a free grid unit and choose the corresponding command in the contextual menu.

12.4.2 Insert Pipe Network Connections

Connections are simply inserted by clicking on an adequate point and dragging the mouse to another adequate point.

See "Add Motion" (➔ p. 187).

Connections are drawn between an input and an output port of two different Pipe Blocks.

Connections can be drawn from input to output ports or vice-versa.

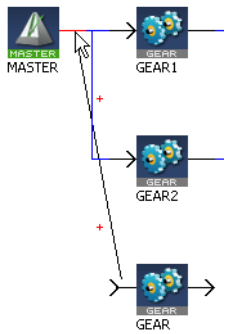


Figure 16-29: Pipe Network - Create a Link

When you try to connect two Pipe Blocks, the editor highlights the target port in red when the connection is allowed.

12.4.2.1 Relation Type for Output-Input is 1-n

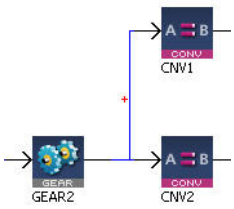


Figure 16-30: Pipe Block - Relation Type for Output-Input

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

One output can be connected to several inputs, but one input can only be connected to one output.

12.4.3 Connect a Comment to a Pipe Block

Connections are drawn between the text area of the comment (title bar is reserved for moving the comment) and the Pipe Block icon.

NOTE

The connection cannot be drawn from the Pipe Block to the comment.
Allowed target is not highlighted.

12.4.4 Pipe Network Editor Controls

12.4.4.1 Basic Functions

This section details the basic moving and editing functionality with Pipe Network blocks.

Edit Pipe Blocks or Comments

To edit Pipe Blocks or comments, double-click an item to open its **Property** dialog box.

NOTE

You can also access the property dialog box of an item through its contextual menu.

Move Pipe Blocks

Pipe Blocks are moved by dragging their center.

- When dragging a Pipe Block, a colored shadow is shown under the Pipe Block indicating where the Pipe Block is dropped.
- When the shadow fills out a complete grid unit, the Pipe Block is placed in this grid unit.

Move Connections

You can move an end-point of a connection from one item to another.

To do this, select the connection and drag an end-point to a new target.

Move Comments

You can drag-and-drop a comment by selecting its title bar.

Remove Pipe Blocks, Comments and Connections

Select one or several items (Pipe Blocks, comments or connections) and choose **Delete Selection** in the menu.

NOTE

You can select several items by clicking on them while pressing either the <Ctrl> or <Shift> keys.

Insert rows and columns

When the shadow does not fill out a whole grid unit, but is squeezed between two grid units, a row or column is inserted before placing the Pipe Block in the newly created grid unit.

When the Pipe Block is dropped on the crossing point of four grid units, a row and a column are inserted simultaneously.

NOTE

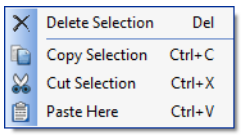
You cannot drop a Pipe Block into a grid unit which is already occupied by a Pipe Block or a comment.

Remove Rows and Columns

It is not possible to remove rows or columns.

If a row or column has been inserted by error, click the **UNDO** icon in the toolbar or press <Ctrl+Z>.

12.4.4.2 Multiple Selections



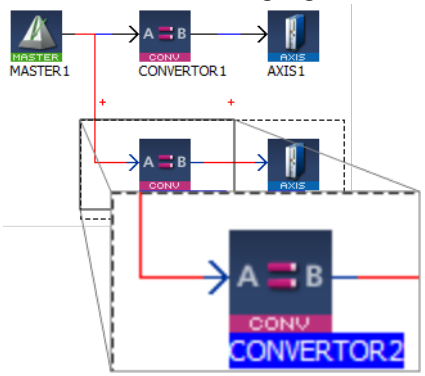
This section details selecting, moving, copying, pasting, and deleting multiple Pipe Network blocks.

Selecting multiple Pipe Network blocks.

Multiple blocks may be selected by either:

- Ctrl-clicking on individual blocks and connectors
- Click-dragging an area to select an area of blocks and connectors

Selected blocks are highlighted and connectors is red.



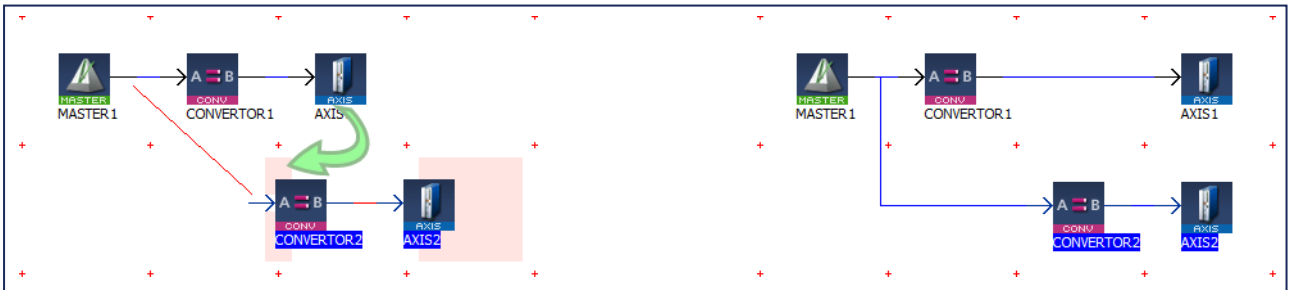
TIP

Be sure to select the connectors between blocks.
If the connectors are not selected, any copied blocks will be unconnected.

Moving multiple Pipe network blocks.

Once selected, multiple Pipe Network blocks and connectors can be moved as a group.

- Click a block and drag it to its new location.
- Light red rectangles show where the blocks will be placed.
- Narrow rectangles represent a block that is placed between two current blocks and which is moved to accommodate the repositioned blocks.



Convertor2 will be placed in a column between Convertor1 and Axis1

Results of inserting Convertor2 in a "new" column.

Copying multiple Pipe Network blocks.

Selected blocks and connectors may be copied by pressing <Ctrl+C> or by using the right-mouse menu.

Pasting multiple Pipe Network blocks.

Selections may be pasted in one of these ways:

- Right-clicking in the Pipe Network Editor and selecting Paste Here.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Clicking in an empty location places the blocks starting at that spot.
- Clicking on an existing block places the copied blocks on top of the selection.
- You will need to move the pasted blocks to see the entire network.

- Press <Ctrl+V> and click in the location you want the blocks to be placed.

TIP

Copied connectors are only pasted when they are connected to a copied block.

Deleting multiple Pipe Network blocks.

Selected blocks may be deleted using either of these methods.

- Press the <Delete> key.
- By right-clicking on a selection and choosing **Delete Selection**.

12.4.5 Plug/Unplug Channels

Right-click on a Pipe Block to plug/unplug a channel of the Softscope.

See "Plug Motion Variables" (➔ p. 274).

12.5 Soft Oscilloscope

The soft oscilloscope (commonly known as softscope or scope) is a tool used to view, in a two-dimensional graph, one or more variable's evolution (vertical axis) across the time (horizontal axis).

- The scope has a set of channels where each can acquire the evolution of a value. (Figure 16-31)
- A value may be the feedback position of an axis, the speed of a machine, or anything else that can be measured with the oscilloscope probes.
- See "Plugging Probes, Mapping Variables" (➔ p. 749) about how to assign a variable to a channel.

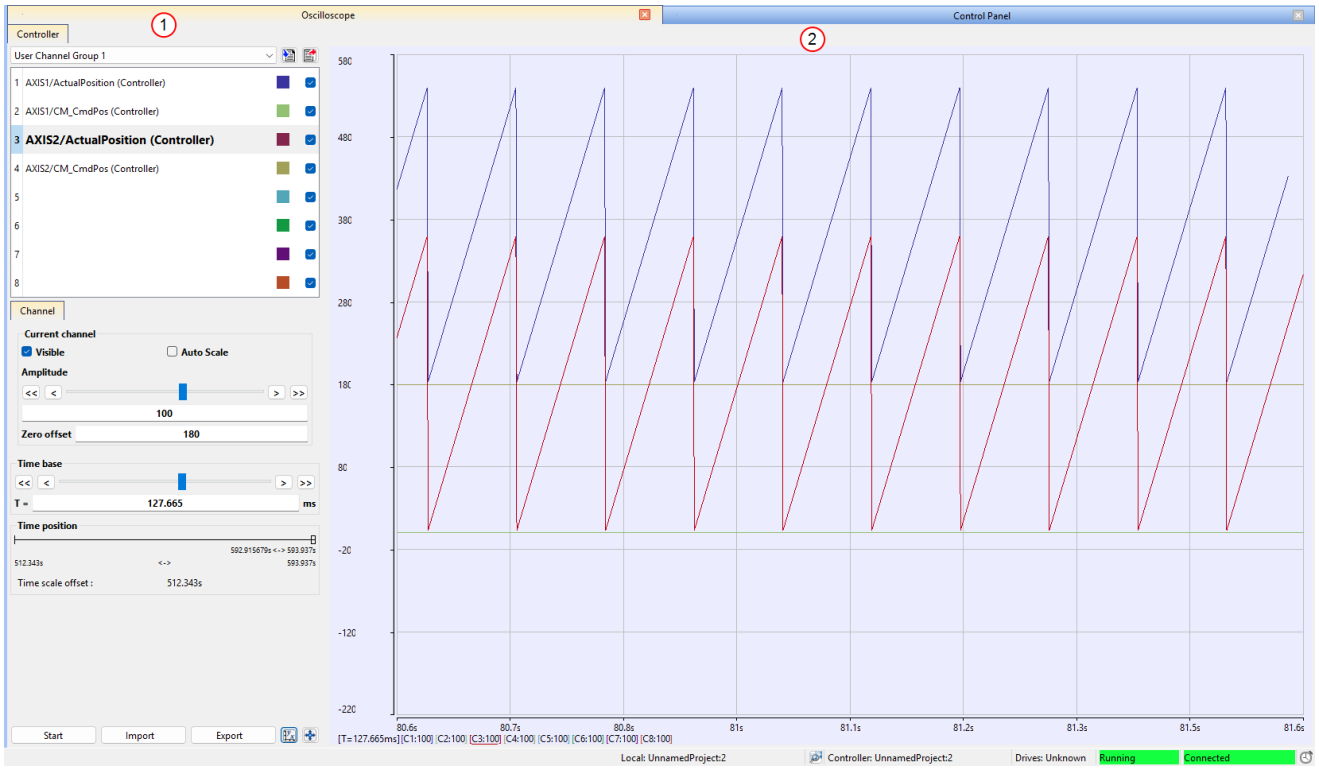


Figure 16-31: Example: Using the Oscilloscope

The soft oscilloscope is a child window in the workspace.

This workspace has two visually distinct parts separated by a splitter which allows you to change the size of the two panes.

12.5.0.1 1. Control Panel

- The "Oscilloscope Control Panel" (→ p. 739) is used to:
  - Select the Channel Group.
  - Assign variables to a channel.
  - Change the settings of the soft oscilloscope (including those of the channels).

12.5.0.2 2. Graphical Area

- The "Oscilloscope Graphical Area" (→ p. 748) shows the traces (or variable values) acquired by the channels.

TIP

You can hide the control panel by dragging the splitter all the way to the left.
To show the control panel again, click and drag the splitter to the right.

12.5.1 Access the Soft Oscilloscope

On the **Tools** menu (Figure 16-32), click the **Oscilloscope** option to open the "Oscilloscope Control Panel" (→ p. 739).

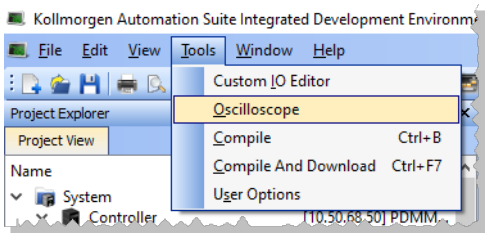


Figure 16-32: Tools menu > Oscilloscope to Access the Scope

12.5.2 About OpenGL

For performance reasons, the scope uses **OpenGL**.

IMPORTANT

Disabling most or all OpenGL accelerations is compensated by an increase in CPU consumption.
It can lead to a point where the soft oscilloscope is not very usable when limited hardware is trying to display loads of samples.
See "System Requirements" (→ p. 43).

If you encounter problems in the quality of drawings, we suggest that you consider these options before contacting our support desk:

- "Verify the Graphic Card Driver is Up-to-date" (→ p. 738)
- "Change the Settings of the Graphic Card" (→ p. 738)
- "Display a Given Amount of Samples per the Refresh Rate" (→ p. 739)

12.5.2.1 Verify the Graphic Card Driver is Up-to-date

- Newer drivers often fix the rendering bugs of OpenGL.

12.5.2.2 Change the Settings of the Graphic Card

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. Open the manufacturer-specific settings of your graphic card.
2. If there are some settings related to **Performance and quality**, try to set them to **quality** (but not high quality) instead of performance, at least for the program **kaside.exe**.

This solves many drawing problems that occur when zooming a lot in the graph.

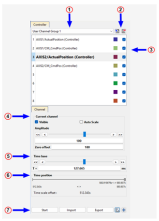
12.5.2.3 Display a Given Amount of Samples per the Refresh Rate

If your channels have acquired a large number of samples, and the refreshing of the graph does not occur frequently enough, do not display all samples at the same time either by:

- Hiding some less useful channels.
  - Use the visibility property in the "Oscilloscope Control Panel" (➔ p. 739).
- Reducing the time-base and/or restricting the time-frame in the time position.
  - In any cases, this action does not stop acquisition or lose your acquired samples.

12.5.3 Oscilloscope Control Panel

Figure 16-33 shows the control panel with these items:



"1. Channel Group Selector" (➔ p. 739)
"2. Import / Export Scope Channel Configuration buttons" (➔ p. 741)
"3. Channels List area" (➔ p. 741)
"4. Current Channel area" (➔ p. 744)
"5. Time Base area" (➔ p. 746)
"6. Time Position area" (➔ p. 746)
"7. Control Panel buttons" (➔ p. 748)

Figure 16-33: Oscilloscope Control Panel

12.5.3.1 1. Channel Group Selector

Click to view Channel Group Selection Selector information

Sometimes, there is a need to compare different types of data or look at them in isolation.

The **Channel Group Selector** drop-down is used to switch between different channel groups.

- A channel group is used to keep a set of channels together, facilitating data collection and analysis.
- With these channel groups, users can tailor their analysis by selecting the specific group that contains the relevant data channels for their current task.
- By organizing the channels in groups, users can toggle between different sets of channels.
  - This eliminates the need to manually reconfigure each time when a different set of data needs to be acquired.
- Example: Configure multiple parameters for one axis and switch to another group with the same parameters configured for another axis to assist in troubleshooting their application.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

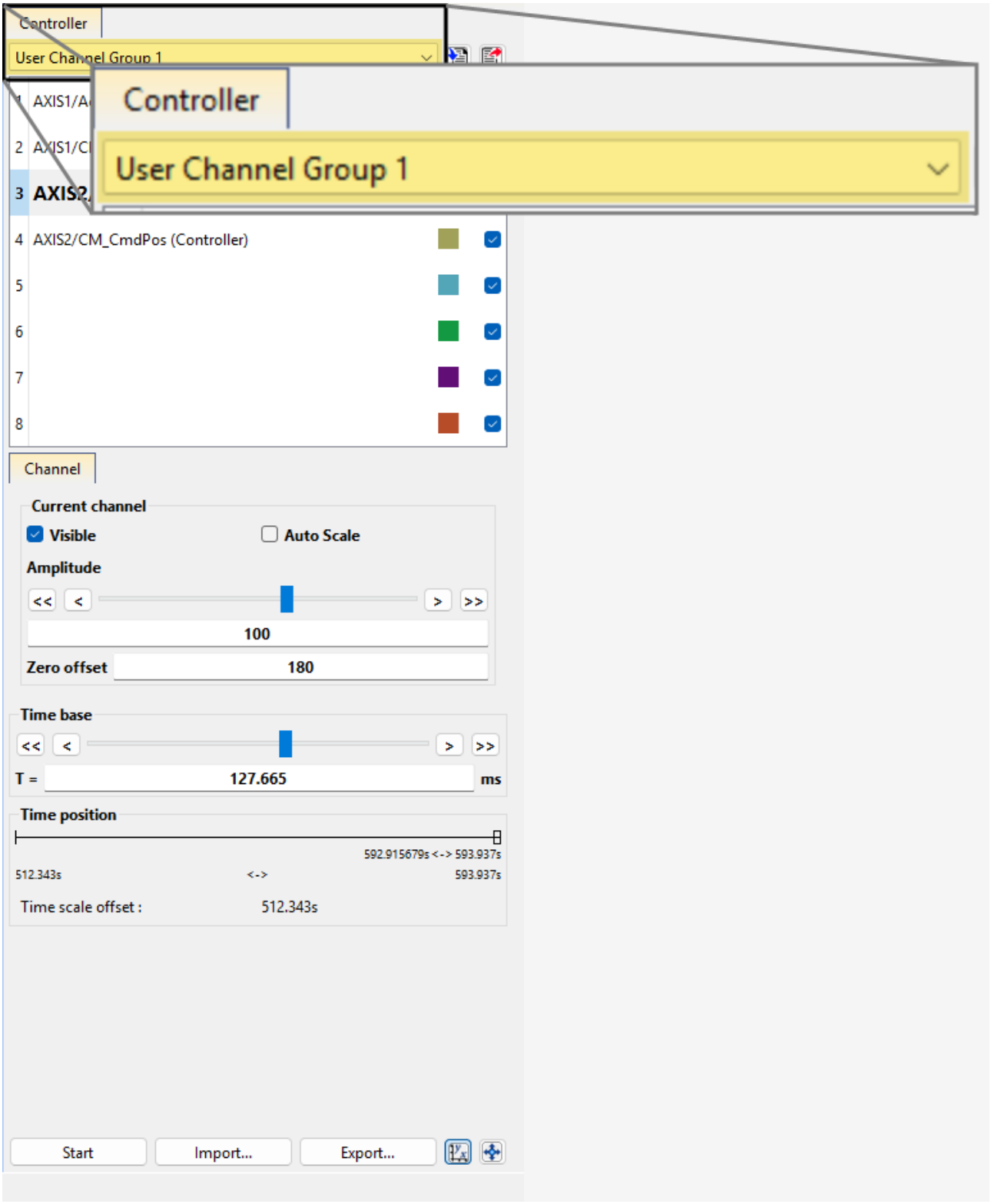


Figure 16-34: Channel Group Selector

Available Channel Groups

- There is one channel group dedicated for the Trace Time Channels.
  - Additional channels are **not** allowed assignment to the Trace Time Channels group.
- There are four channel groups available for customizing channel selections.
  - A maximum of eight channels can be assigned in each group. (see "3. Channels List area" (→ p. 741)).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

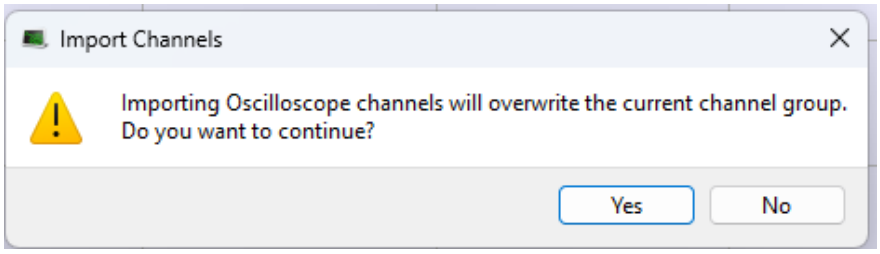
NOTE

The application **must be** connected and running to be able to change the channel group selection.
The Channel selections in the groups are automatically saved when the project is saved.

12.5.3.2 2. Import / Export Scope Channel Configuration buttons

Click to view Import / Export buttons information

These buttons are at the top of the Control Panel.

Button	Description
Import Scope Channels Configuration ( )	Used to import the active user group channels configuration to an XML file. **NOTE** If the active User Channel group is not empty, a confirmation message appears. (Figure 16-35)  **Figure 16-35:** Import Channels confirmation message
Export Scope Channels Configuration ( )	Used to export a previously exported configuration XML file to the active User Channel group. **NOTE** This button exports only the channels in the currently active user channel group.

12.5.3.3 3. Channels List area

Click to view Channels list information

This area lists the eight available channels and the variables assigned to each channel.

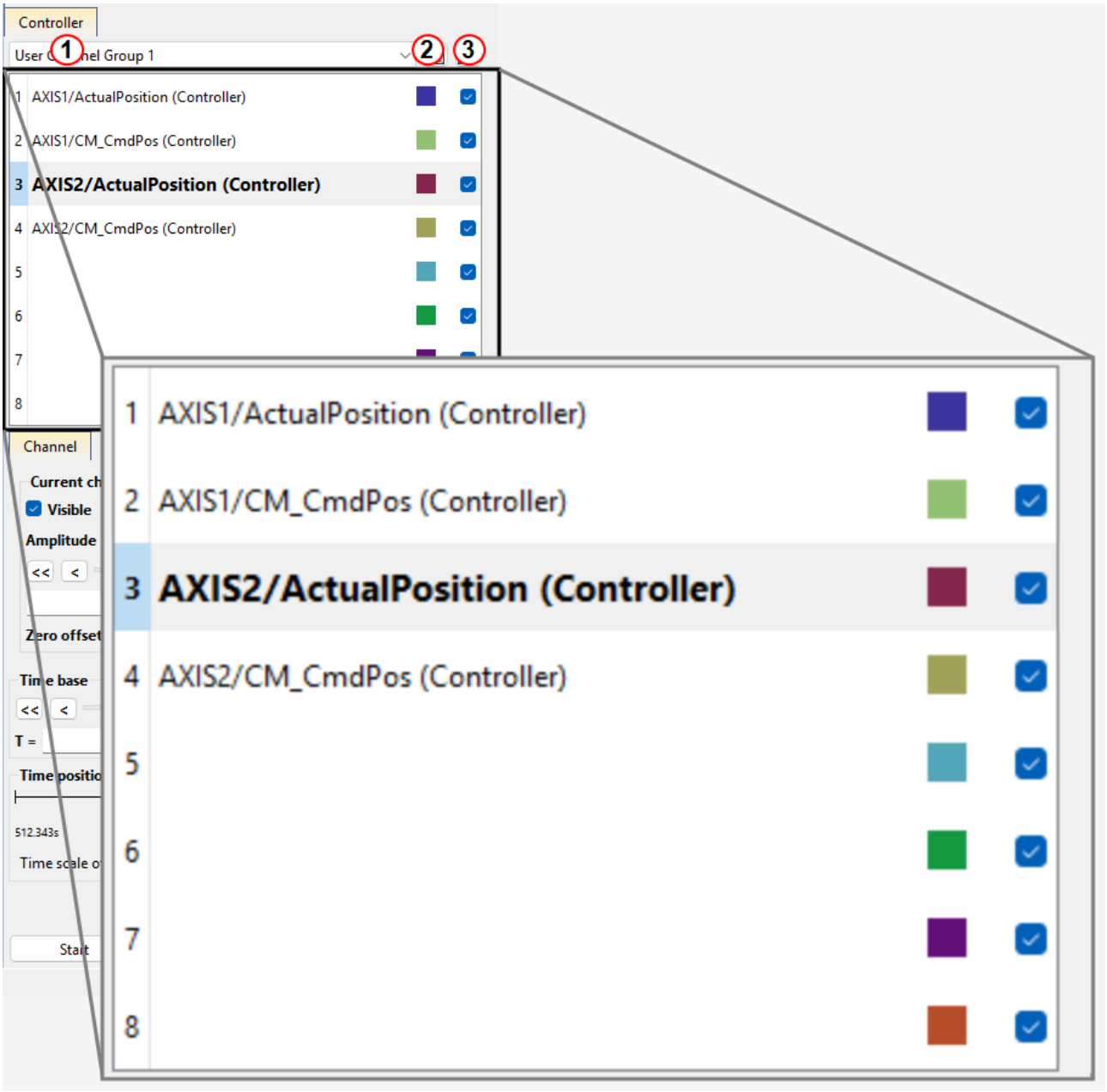


Figure 16-36: Scope Control Panel - Channels

Each channel shows:

- The **name** of the associated variable (1).
- The **color** of the associated trace in the graph with a color icon (2).
- The **visibility** of the associated curve with a check box (3).
  - This is to show or hide each captured trace.

TIP

- Double-click the curve color icon to change its color.
- Click the curve check box to change its visibility.
- Right-click a channel to map, unmap, or edit the color of a variable's trace.
- When the Trace Time channels are present, the Map or Unmap options are not available.

When selecting a channel in the channels list, it is superimposed on the existing traces.

Related information is displayed on the left and lower sides of the graph.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

12.5.3.3.1 Trace Times Channel Group

Click to view Trace Times Channel Group information

The Trace Times Channel Group entry in the Channel Group Selector is used to load and display these channels.

Channel 1: **CycleJitter** (in µs)

When the motion is started, the current cycle time remains constant on an average of several cycles, and equal to the EtherCAT cycle time which is a constant value (1000, 500 or 250 µs).

The CycleJitter is due to EtherCAT transmissions that can vary in a particular cycle (i.e., call out (1)).

The channel 1 of the scope monitors the time difference between the expected Cycle Time and the actual Cycle Time:

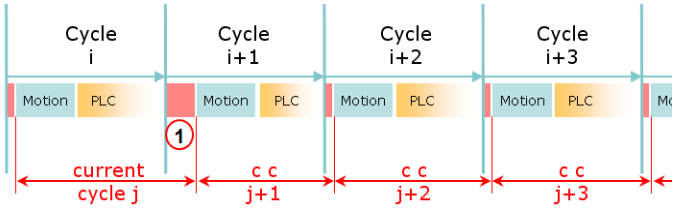


Figure 16-37: Cycle Time Calculation

Channel 2: **Motion execution time** (microseconds)

Channel 3: **PLC execution time** (microseconds)

Channel 4: **PCMM2G - PLC Margin** (microseconds)

- This channel monitors the available execution time for the PLC within a cycle:
  `Cycle Time Period - EtherCAT network execution time - PLCProgExecTime`
- The range is 0 to cycle period in microseconds.
- The measurement is calculated and updated each cycle period.
- If the PLC execution time spans multiple cycles, then the PLC Margin Time is zero.

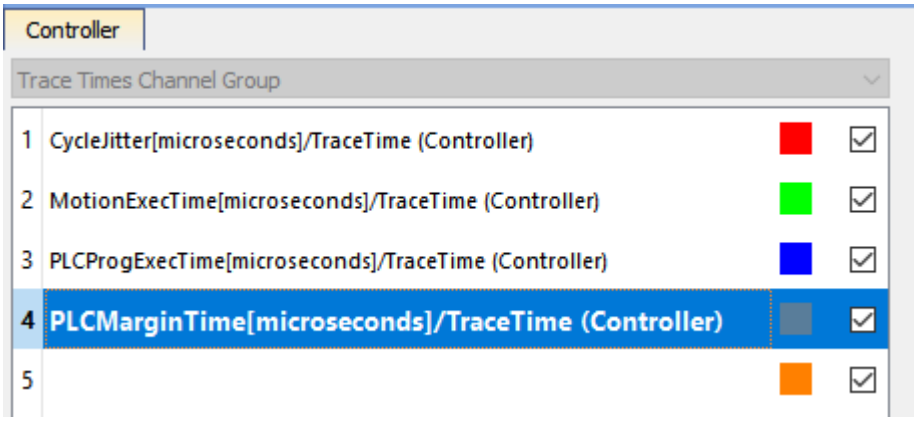


Figure 16-38: PCMM2G - Oscilloscope Trace Times Channels

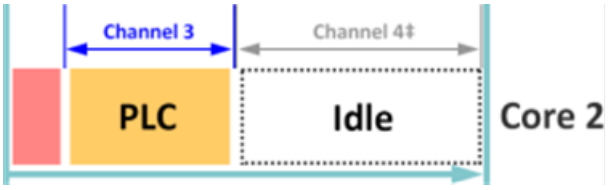


Figure 16-39: PCMM2G - Multiple Core Controller, PLC, and PLC Margin Time Calculations

12.5.3.3.2 Simulator

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Click to view Simulator information

- Channel 1: Not supported.
- Channel 2: **Motion execution time** (microseconds)
- Channel 3: **PLC execution time** (microseconds)
- Channel 4: Not supported.
  - Channel 1 (CycleJitter) and Channel 4 (RealTimeMargin) are always zero with the simulator.
  - The MotionExecTime and PLCProgExecTime traces are visible with simulated values.

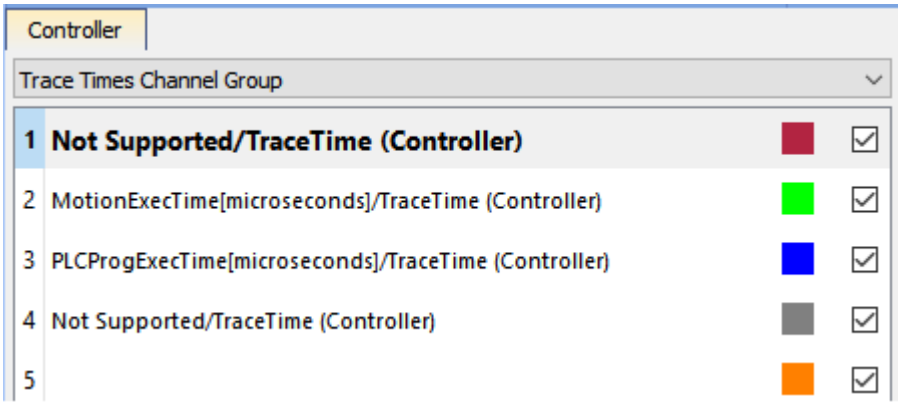


Figure 16-40: Simulator - O-Scope Trace Times Channels

See "Tasking Model / Scheduling" (→ p. 450).

12.5.3.4 4. Current Channel area

Click to view Current channel area information

This area details the properties of the currently selected channel.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

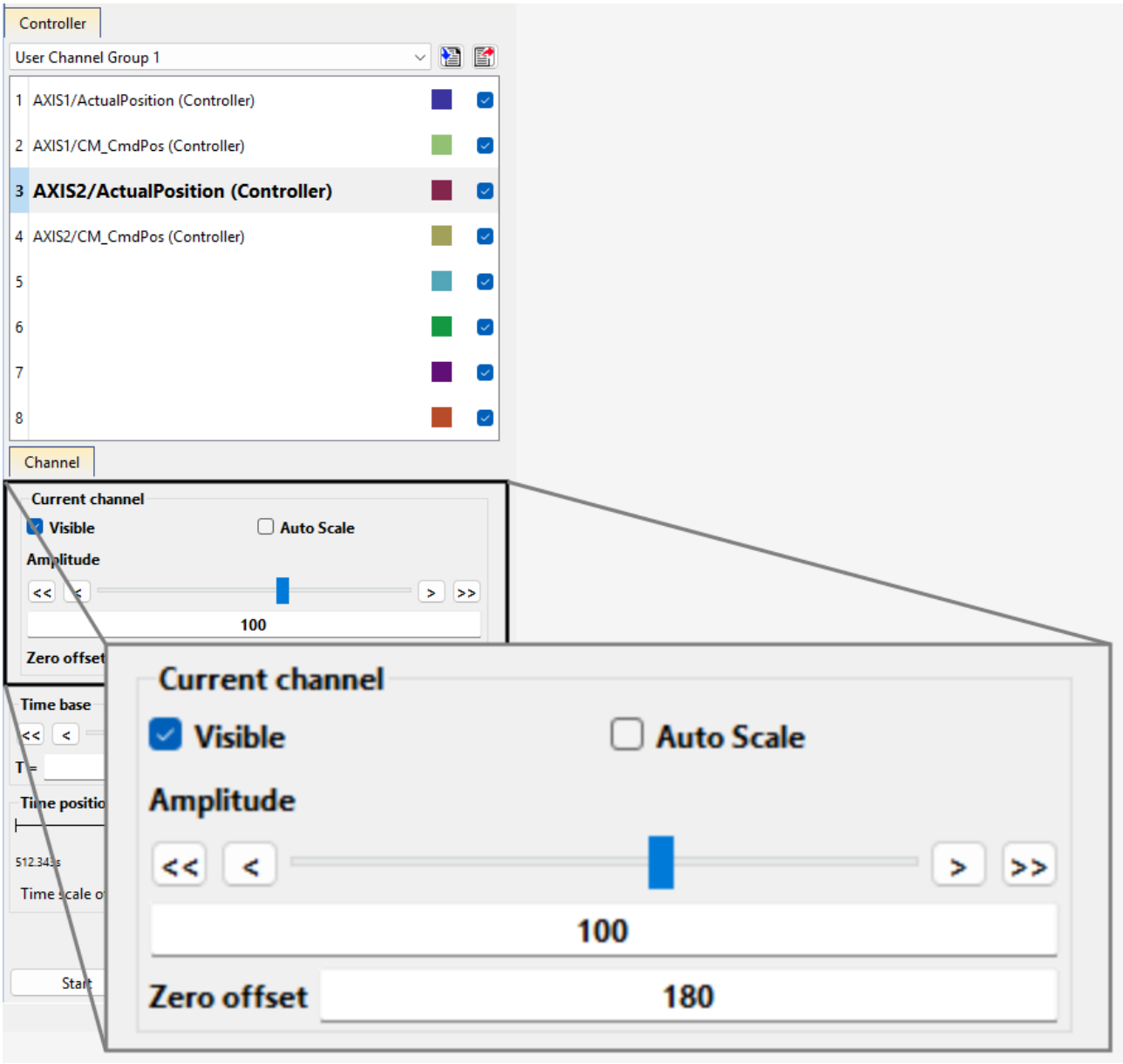


Figure 16-41: Current Channel area

Properties	Description
Visible	A channel must be visible to be drawn on the graph. Even if not visible, data is still collected by the associated probe.
Auto Scale	A channel in Auto Scale mode automatically adapts its amplitude and zero offset to display all samples. See "Unit per Division" (→ p. 754). • Setting the Auto Scale mode disables the possibility of changing the **Amplitude** and **Zero offset**. • See "Setting Scale" (→ p. 753).
Amplitude	Controls the amplitude (unit/division) of the channel. • The buttons and slider change the amplitude according to a logarithmic scale. • The dialog box allows a more precise definition of the value.
Zero offset	The curve is vertically shifted so that this value is located halfway through the graph height.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

12.5.3.5 5. Time Base area

Click to view Time base area information

This area is used to set the speed at which all the lines for each channel are drawn.

It is calibrated in milliseconds per division.

- Its usage is similar to the Amplitude property.
- The time-base can always be changed, even during sampling.
  - See "Setting Scale" (➔ p. 753).

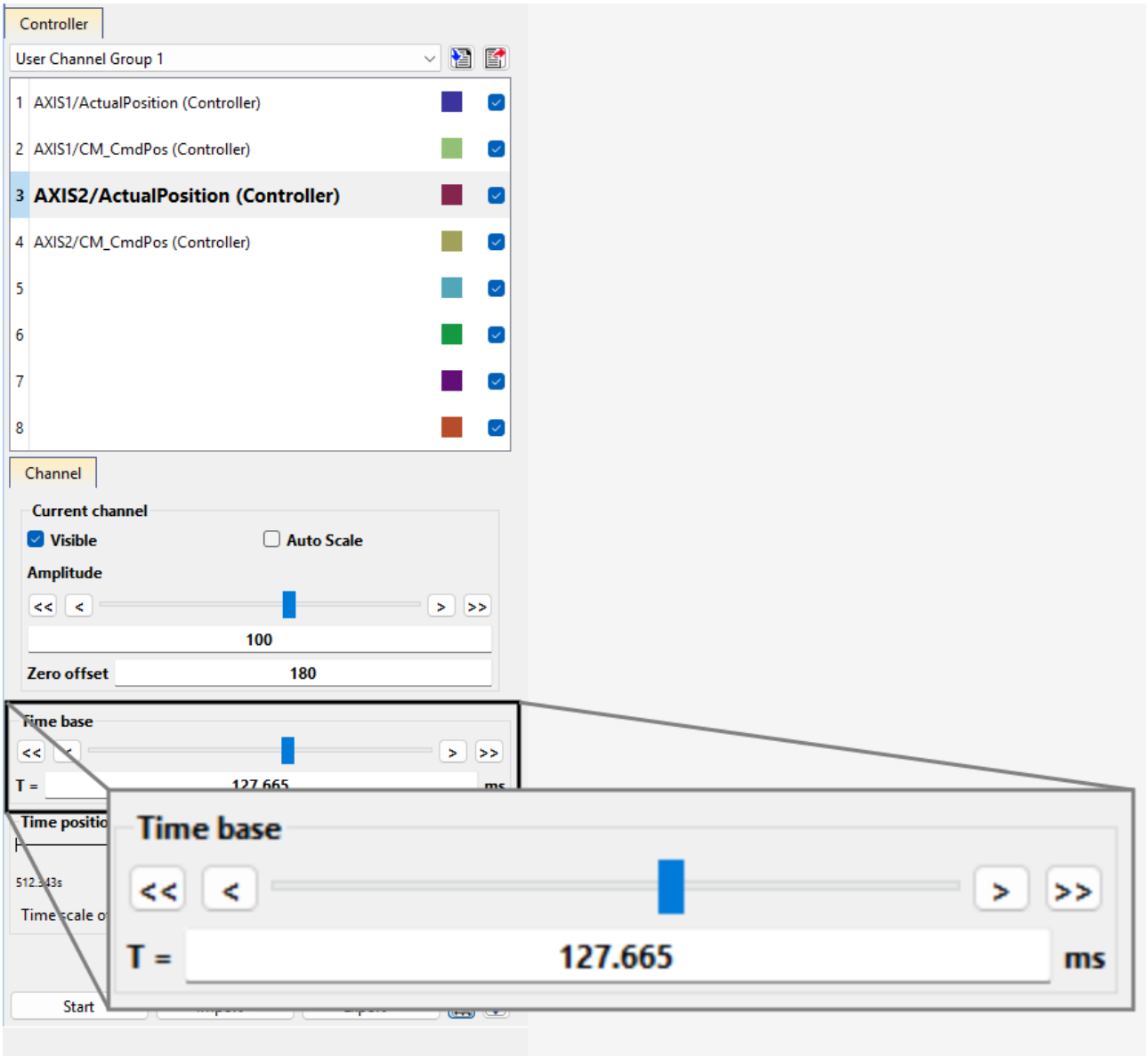


Figure 16-42: Scope Control Panel - Time-base

To setup the time-base properly, the total measurement duration and the required time resolution have to be taken into account.

12.5.3.6 6. Time Position area

Click to view Time Position information

This area is used to change the time-frame of the acquired samples shown on the graph.

It is composed of:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- A single **horizontal line** represents all the acquired samples with start and stop timings.
- A **rectangle** represents only the time slot of the acquired samples shown in the graphical area (the time-frame) with timings:

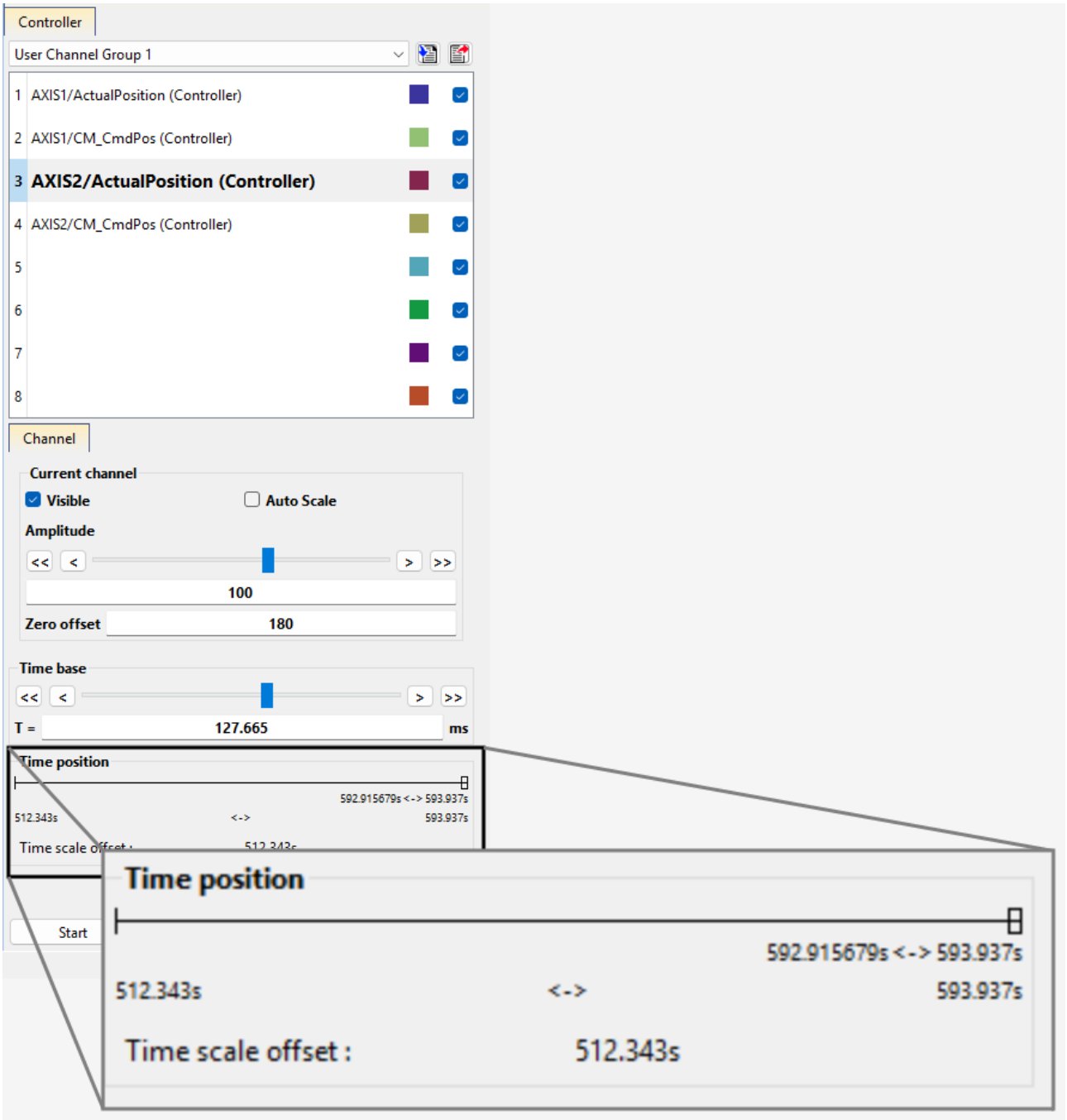


Figure 16-43: Scope Control Panel - Time Position

NOTE

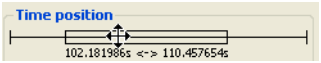
The acquisition of samples is limited to 100,000 cycles (i.e., 100 s when cycle time is set to 1000 µs, and 25 s when cycle time is set to 250 µs).
When you reach this limit:

- The first data added to the queue are the first data to be removed (FIFO queue).
- The start timing increases.

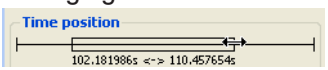
Change the time slot with the mouse by:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Moving the rectangle:



- Changing the size:



The **Time Scale Offset** is the time value of the first sample the graph when plotting is started.

- Using this as an offset, the time axis is always started at 0 seconds.
- To get the actual time value of any sample, add the time scale offset to the Time axis value.
  - `Actual Sample Time Value = Time Scale Offset + Time axis value`

12.5.3.6.1 Set the Time-frame

When clicking anywhere on the horizontal line, the time-frame is centered on the clicked point.

It is also possible to move the time-frame by clicking on its rectangle part and dragging.

Click the time-frame's left or right ends and drag to the new size.

NOTE

During acquisition, the time position item is disabled and displays the progression of acquisition.

12.5.3.7 7. Control Panel buttons

Click to view Control Panel buttons information

These buttons are at the bottom of the Control Panel.

Button	Description
Start/Stop (Start)	• Used to start or stop the acquisition of samples. • When starting acquisition, all previous samples are lost.
Import (Import)	• Used to: • Import data from a CSV file. • View the exported/saved scope data. • The saved data is plotted in the graphical area. • The channel details are available in the "3. Channels List area" (➔ p. 741). • See "Import the Oscilloscope Data" (➔ p. 755). **IMPORTANT** When the "1. Channel Group Selector" (➔ p. 739) is **Trace Time Channels**, the **Import** button is disabled.
Export (Export)	• Used to export trace / acquisition data to a CSV file. • See: • "Oscilloscope Graphical Area" (➔ p. 748) • "Export the Oscilloscope Data" (➔ p. 756)
Graduations ( )	• Used to view or remove the axis graduations of the graphical area. • Graduations are visible by default.
Autofit ( )	• Used to change the time-frame of the graph and amplitudes. • It zero offsets channels so they all fit entirely into the graphical area.

12.5.4 Oscilloscope Graphical Area

The graph displays a subset of the collected data: the **time-frame**.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

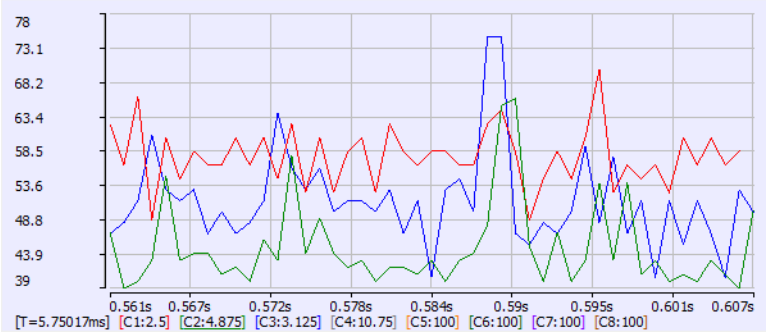


Figure 16-44: Example of the traces generated in the Oscilloscope.

The graph has these features to view and analyze the data:

- Graduations are displayed on the left and lower sides of the graph.
- Information concerning the time-frame of the graph and the amplitude of channels also appears at the bottom of the graph.
  - The current channel amplitude is underlined and the coordinates of the nearest collected sample are displayed.
- It is possible to **zoom** in the graph using various methods.
- It is possible to **move** the contents of the graph within the time-base.
- See "Trace Zoom Feature" (→ p. 754).

NOTE

Moving the contents is possible **only** when the acquisition is stopped.

See "Export the Oscilloscope Data" (→ p. 756) for capturing the trace data.

12.5.5 Traces

The trace is the resulting graph of the variable's evolution against time, with the more distant past on the left and the more recent past on the right.

NOTE

The acquisition of samples is limited to 100,000 cycles (i.e., 100 s when cycle time is set to 1000 µs, and 25 s when cycle time is set to 250 µs).
When you reach this limit:

- The first data added to the queue are the first data to be removed (FIFO queue).
- The start timing increases.

12.5.6 Plugging Probes, Mapping Variables

A probe is a virtual measurement point that can be connected to a variable.

These types of variables can be mapped:

- Pipe Block variable
- IEC 61131-3 variable, which is any other variable.
- PLCopen axis values.

  NOTE

  Your application **must be connected and running** to plug a variable to the channel.

You can connect a probe to a variable in one of these ways:

- From the Softscope, see "Map a Variable to the Oscilloscope" (→ p. 750).
- From the Dictionary, see "Plug a Probe from the Dictionary" (→ p. 751).
- From the Pipe Network, see "Plug a Probe from the Pipe Network" (→ p. 752).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

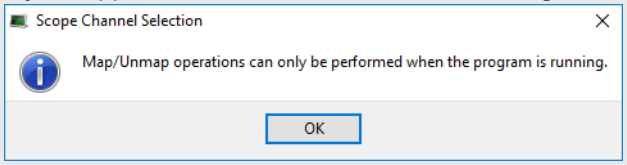
12.5.6.1 Map a Variable to the Oscilloscope

A probe (variable) can be directly plugged or mapped from within the Oscilloscope.

1. Right-click on any channel (already mapped or not) in the channels list.
2. Click the **Map** option to open the **Scope Variable Selector**.

NOTE

If your application is not connected and running, this message appears:



3. Navigate through the available variables and select the one to connect to the channel.

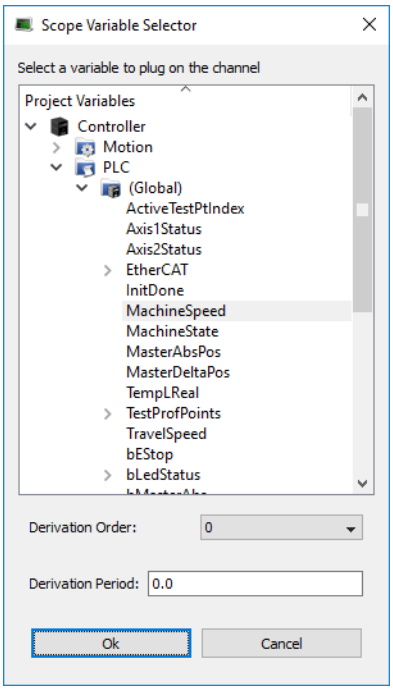


Figure 16-45: Use the Variable Selector to choose a variable to be mapped in the Oscilloscope

NOTE

The **Scope** Variable Selector contains only the PLC variables eligible for the Softscope as long as they are not in a UDFB instance.
This includes: BOOL, BYTE, DINT, DWORD, INT, LINT, LREAL, LWORD, SINT, TIME, UDINT, UINT, ULINT, USINT, WORD.
Only a subset of variables are displayed (e.g., ActualVelocity [1] is not visible), when in Simulated mode.

See:

- "Axis Pipe Block" (➔ p. 332) about Axis pipe block positions.
- "Axis Positions Data" (➔ p. 369) about PLCopen Axis positions.

4. (Optional) Set the **Derivation Order**.
   This is the number of derivatives to apply to the trace.

   If the variable selected is the axis command position, then setting the **Derivation Order** to:

[1] The measured value is the instant velocity of the axis in RPM*1000.
Some oscillations are visible because it is an instant velocity, not an average velocity.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- 1 - Generates a trace for the axis command velocity.
- 2 - Generates a trace for the axis command acceleration.
- 3 - Generates a trace for the axis command jerk.

5. (Optional) Set the **Derivation Period**.
   The value entered should be either `0.0` (no modulo) or the Modulo Period (e.g., 360.0).
   - If the selected Derivation Order is greater than zero, the Derivation Period of the selected signal can be used to remove rollover spikes in the derivative value if the variable is of a periodic nature as the result of modulo behavior.

12.5.6.1.1 Un-map Variables from the Scope

Use this procedure to disconnect a probe:

1. Right-click on the channel to remove from the channels list and select **Unmap** to disconnect the probes.
2. Select the **Unmap all Oscilloscope Channels** command from the right-mouse menu in the Dictionary or the Pipe Network.

12.5.6.2 Plug a Probe from the Dictionary

A probe (variable) can be directly plugged or mapped to the Oscilloscope from the Dictionary.

Procedure

1. In the **Dictionary**, right-click on the variable.
2. In the menu, click the **Map to Oscilloscope channel ...**. (Figure 16-46)

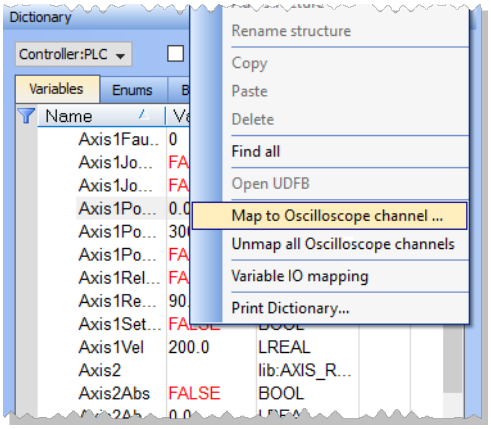


Figure 16-46: Map to Oscilloscope channel ... menu option

Notes

- This menu option is enabled if the variable "Data Types" (➔ p. 286) is eligible for the softscope, as long as they are not in a UDFB instance.
- This menu option is disabled when the Trace Times Channel Group is selected in the "Oscilloscope Control Panel" (➔ p. 739).
- The Channel selections in the groups are automatically saved when the project is saved.

3. Define the probe parameters.
   There is a difference between a Pipe Network probe or a PLC probe.
   - With Pipe Network variables, select which Data point is being used.
   - PLC variables may require navigation within the group to find the correct item.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

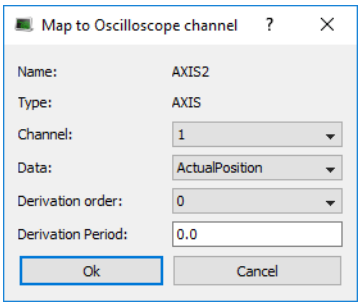


Figure 16-47: Associating a Pipe Network

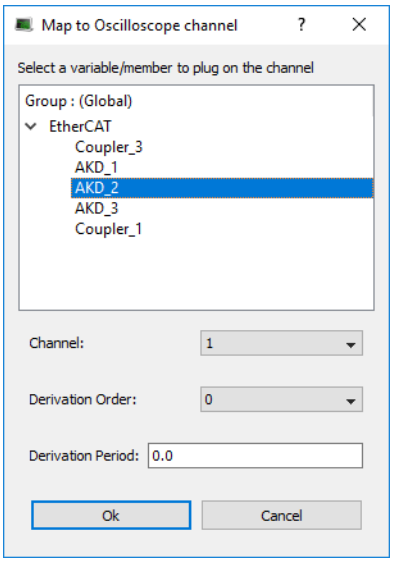


Figure 16-48: Associating a PLCopen variable to a Channel

Field	Description
Channel	Channel's number where the variable has to be plugged
Data	Desired variable information to show (the list depends on the type of Pipe Block.
Derivation Order	• Performs a derivation of the measurement of the selected variable. • If this value is different from 0 (zero), the derived value of the selected order is shown on the selected channel.
Derivation Period	• Specifies the modulo period for a periodic variable to remove spikes in the display of derivative orders greater than zero. • The value entered should either be 0.0 (No Modulo) or the Modulo Period (e.g., 360.0).
Name	Variable's name.
Type	Variable's type.

NOTE

The **Map to Oscilloscope channel ...** menu option is disabled if the KAS-IDE is not connected to the controller or if the project is not running.
This menu option is disabled when the Trace Times Channel Group is selected in the "Oscilloscope Control Panel" (→ p. 739).

12.5.6.3 Plug a Probe from the Pipe Network

A probe (variable) can be directly plugged or mapped to the Oscilloscope from the Pipe Network.

Procedure

1. Right-click on a Pipe Block.
2. In the menu, click the **Map to Oscilloscope channel ...**. (Figure 16-49)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

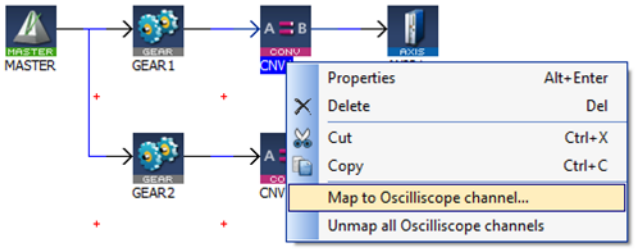


Figure 16-49: Map to Oscilloscope channel ... menu option

The **Map to Oscilloscope channel ...** dialog box opens. (Figure 16-50)

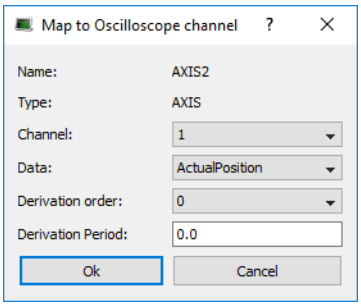


Figure 16-50: Map to Oscilloscope channel ... dialog box

3. Define the probe parameters.

Field	Description
Channel	Channel's number where the variable has to be plugged
Data	Desired variable information to show (the list depends on the type of Pipe Block.
Derivation Order	• Performs a derivation of the measurement of the selected variable. • If this value is different from 0 (zero), the derived value of the selected order is shown on the selected channel.
Derivation Period	• Specifies the modulo period for a periodic variable to remove spikes in the display of derivative orders greater than zero. • The value entered should either be 0.0 (No Modulo) or the Modulo Period (e.g., 360.0).
Name	Variable's name.
Type	Variable's type.

NOTE

The **Map to Oscilloscope channel ...** menu option is disabled if the KAS-IDE is not connected to the controller or if the project is not running.
This menu option is disabled when the Trace Times Channel Group is selected in the "Oscilloscope Control Panel" (→ p. 739).

12.5.7 Setting Scale

The soft oscilloscope graph is divided into eight units for the horizontal time scale (X-axis) and eight units for the amplitude (Y-axis).

These units can be user-defined by using these configuration panels:

- "Time Scale" (→ p. 754)
- "Variable Scale" (→ p. 754)

12.5.7.0.1 Unit per Division

This is the time-base value for the X-axis and the amplitude value for the Y-axis.

Example: If the time-base is 10ms and the amplitude is 1, each division in the soft oscilloscope grid corresponds to a time of 10ms for the X-axis and an amplitude of 1 for the Y-axis.

12.5.7.1 Time Scale

The time scale can be configured with the **Time-base** configuration panel.

- The default value is 100ms/unit with the limits being 0.1ms to 25,000ms.
- The new value can be entered by hand directly in the text field or by using the buttons:

Buttons	Description
< >	Used to divide / multiply the time-base by 2. • Performing a **division** corresponds to a **zoom in**. • Performing a **multiply** corresponds to a **zoom out**.
<< >>	Used to divide / multiply the time-base by 10.

The base time unit is 1 ms.

TIP

Modify the time scale by scrolling the mouse wheel with the cursor located in the graphical area.

12.5.7.2 Variable Scale

Variable scaling is done by modifying the amplitude and offset value of a channel.

The variable scale can be configured in different places:

- The
- The Current channel control panel.
- The Edit all channels dialog.

TIP

You can modify the variable scale by pressing down the Ctrl key while scrolling the mouse wheel with the cursor located in the graphical area.

NOTE

The changes affect only the selected channel.

12.5.8 Trace Zoom Feature

The zoom feature is used to magnify or reduce a portion of a trace.

Two zoom modes are available:

Mode	Description
Amplitude zoom	Used to have a better view of a part of a signal. This zoom operation updates the amplitude and zero offset value. **NOTE** When performing an amplitude zoom, the zero offset is automatically set by the cursor position.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Mode	Description
Time zoom	Used to expand/collapse the time-base in order to have a better view of the signal evolution through time. This zoom operation updates the time-base value.

The zoom operations can be done by either:

- Modifying the corresponding values by hand.
- Using the mouse wheel.

See "Setting Scale" (→ p. 753) about setting the amplitude, zero offset, and time-base values.

Mouse Shortcuts

Action	Result
Scroll up the mouse wheel.	Expands the time-base value.
Scroll down the mouse wheel.	Collapses the time-base value.
Press the Ctrl key while scrolling up.	Makes the amplitude value greater.
Press the Ctrl key while scrolling down.	Makes the amplitude value smaller.

12.5.9 Import the Oscilloscope Data

This procedure imports data from a CSV file.

IMPORTANT

Only CSV files can be imported.

1. Navigate to the Soft Oscilloscope.
2. Click the **Import** button.
   If the Soft Oscilloscope already has data, an overwrite confirmation message appears. (Figure 16-51)

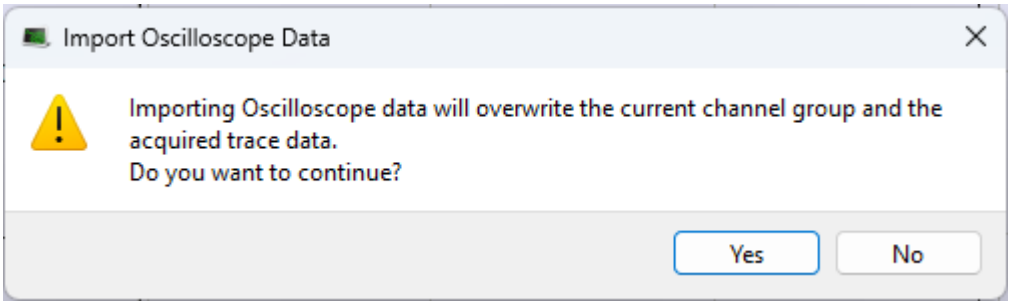


Figure 16-51: Import Oscilloscope Data confirmation message

NOTE

The **Import** button is grayed out when the application is running and channels sample data acquisition has started.

3. Click **Yes** to continue.
   The **Choose a File to Load Oscilloscope Data From** dialog opens. (Figure 16-52)

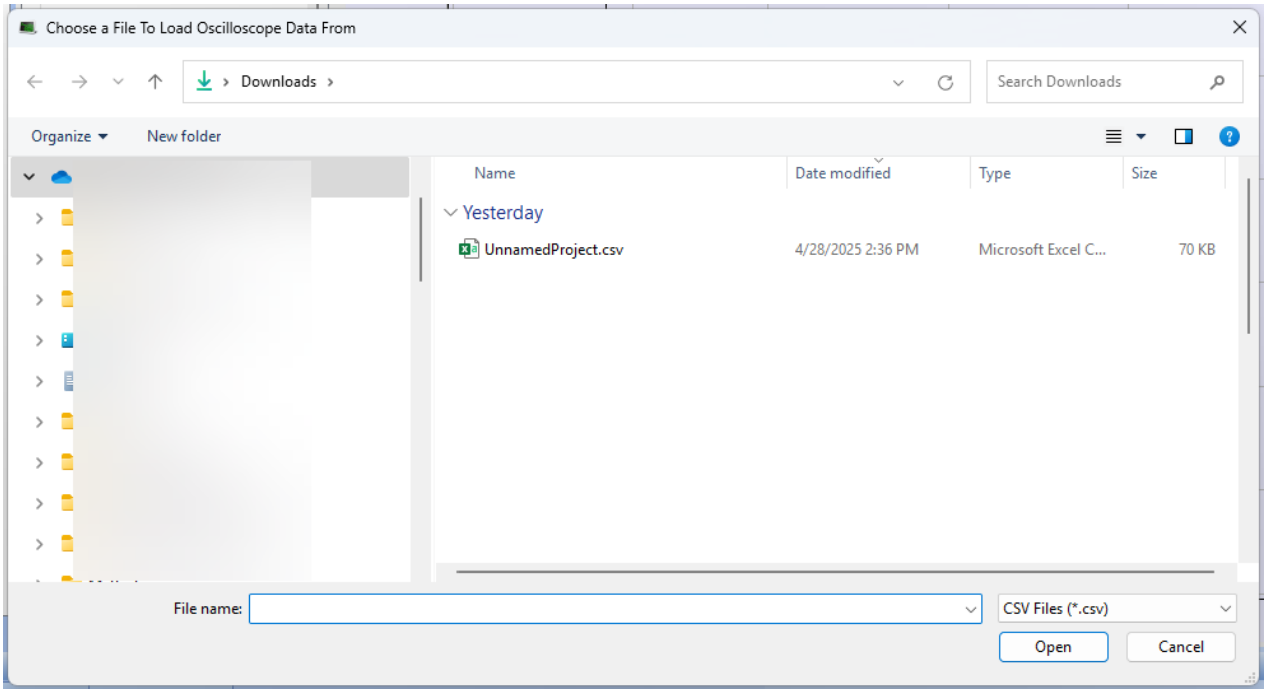


Figure 16-52: Choose a File to Load Oscilloscope Data From dialog

4. Locate and select the CSV file to import.
5. Click **Open** to continue.
   The traces are plotted in the "Oscilloscope Graphical Area" (→ p. 748).

12.5.10 Export the Oscilloscope Data

This procedure exports the trace data to a CSV file.

1. Navigate to the Soft Oscilloscope.
2. **Start** the data collection.
3. Wait for the probe data to save to be collected.
4. **Stop** the data collection.
5. Click the **Export...** button.
6. Select the location where to save the CSV file.
7. Click the **Save** button.

NOTE

- The Export operation is possible even when acquisition of samples is in progress.
  - When this happens, the latest exported data are the data collected when the CSV file is defined.
- The acquisition of samples is limited to 200 sec when the cycle time is set to 1000 µs (respectively 50 s with 500 µs, and 25 s with 250 µs).

12.6 Practical Application: Using Trace Time To Measure CPU Load

To determine the overall controller CPU usage, look at the HW Status tab on the **Diagnostics** page of the controller's web server.

12.6.0.1 PCMM2G

- The CPU usage alone may not provide an accurate indicator of the Motion and PLC CPU usage.
  - This is because the CPU usage is the average of all the CPU cores.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

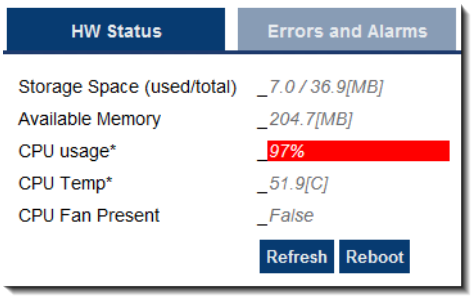


Figure 16-53: Example: Practical Application PCMM2G

12.6.1 Oscilloscope Trace Times

The IDE Oscilloscope trace times can be used to analyze the application performance on a controller or programmable drive.

This section describes techniques used to interpret the trace times to examine the real-time performance.

There are two major parts to consider when evaluating total performance:

- **Real Time**: EtherCAT + Motion Engine + PLC program.
- **Non-Real Time**: Everything else (the background tasks).

The Oscilloscope trace times provide a very good tool to examine the Real Time response.

- Although it doesn't provide the complete system picture, it is a good place to start.
- It can provide some indication about the Non-Real Time load.
  - However, the best indicator is the overall **CPU usage** and the Controller Log messages.

You need to know the Cycle Time for the system.

From the **Project View**, select the **EtherCAT** view and the "EtherCAT Master Settings tab" (➔ p. 539).

The update period for the system in this example is set to 250 microseconds.

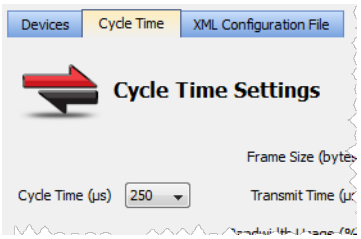


Figure 16-54: Cycle Time tab

The **Trace Times** traces are enabled by pressing the **Plug Trace Times channels** button in the Oscilloscope view when your application program is running.

This button automatically configures the Channels:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

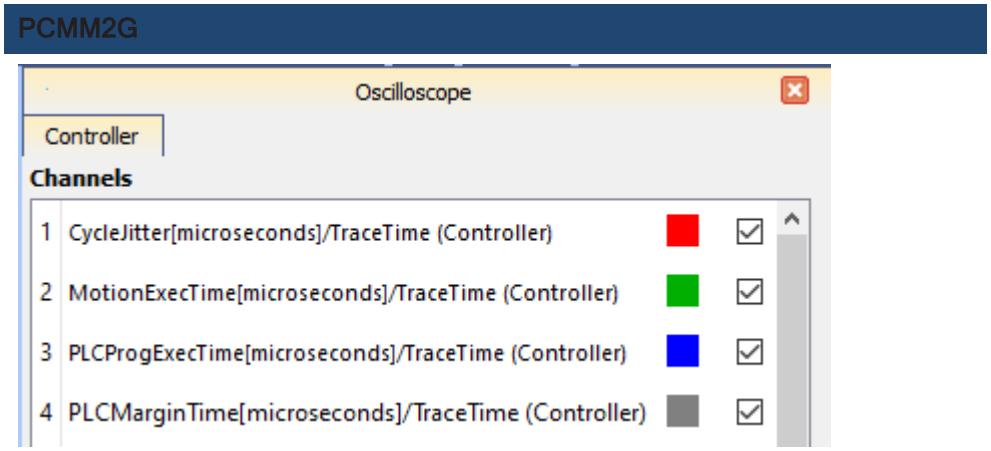


Figure 16-55: PCMM2G

12.6.1.1 Press the Start Button to Collect Some Data

The first thing to do is to collect data during the normal application operation, particularly once the system has reached a steady state.

Press **Start** and let the data collect for a few seconds and then press the **Stop** button.

The first traces to examine are the **MotionExecTime** and **PLCProgExecTime**.

Configure the **Amplitude** and **Zero offset** so you can see both traces easily.

This table shows some recommended values based on several Cycle Time values:

Cycle Time	Amplitude	Zero Offset
250ms	20	80
500ms	40	160
1000ms	80	320

TIP

Clearing check boxes for the **CycleJitter** and **RealTimeMargin** or **PCLMarginTime** traces is useful so they don't clutter the view.

Example

This example has a Cycle Time of 250 microseconds.

- The **MotionExecTime** average is about 27 microseconds.
- The **PLCProgExecTime** average is about 68 microseconds.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

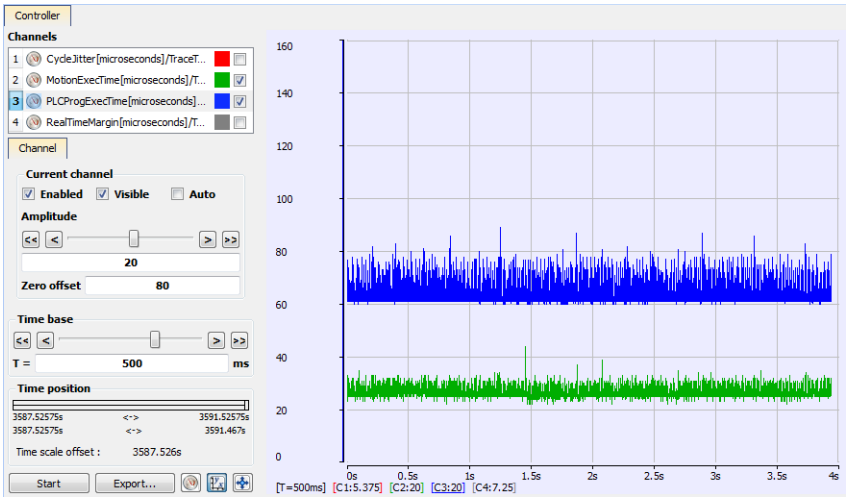


Figure 16-56: PLCProgExecTime and **MotionExecTime**

For single-core controllers, we want to investigate the combined percentage of cycle time for both the **MotionExecTime** and **PLCProgExecTime**.

- The average time for the **MotionExecTime** + **PLCProgExecTime** is `95 (27 + 68 = 95).`
  - This is about 38% of the cycle (95 / 250).
    - This is a good value.

For multi-core controllers, we want to investigate the percentage of cycle time for each **MotionExecTime** and **PLCProgExecTime** individually.

- **MotionExecTime** is about 11% of the cycle.
- **PLCProgExecTime** is about 27% of the cycle.
  - These are both good values

12.6.2 Check the Peak Times

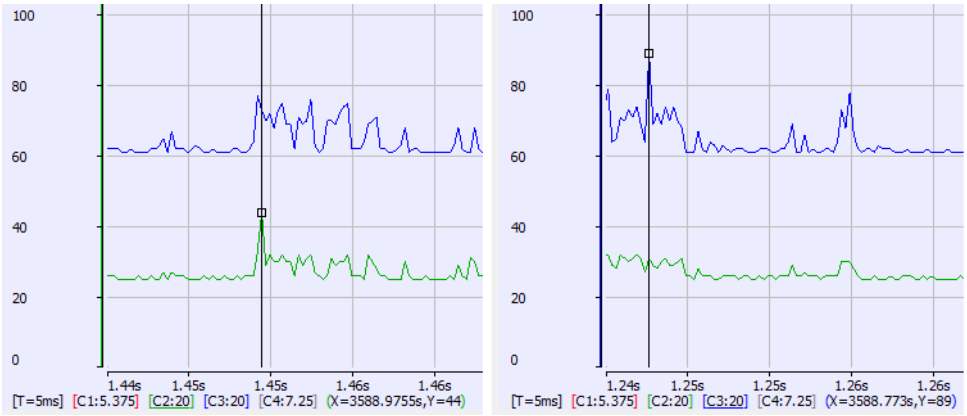
The next step is to examine the spikes.

We will examine the **MotionExecTim**, **PLCProgExecTime**, **RealTimeMargin** or **PLCMarginTime** and **CycleJitter** traces.

1. Reduce the **Time** base and move the traces left or right with the mouse while holding the left mouse button.
2. Position the cursor to measure the peak.

In this example:

- The **MotionExecTime** peak is 44.
- The **PLCProgExecTime** peak is 89.
  - This is reasonable.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Figure 16-57: Peak Times

12.6.2.1 RealTimeMargin or PLCMargin Time

PCMM2G

To view the minimum **PLCMarginTime**, configure the **Amplitude** and **Zero offset** so you can see the trace near zero.

In this example:

- The minimum (closest to zero) is 52 microseconds.
- This provides a 21% (52 / 250) **PLCMarginTime** (which is good).

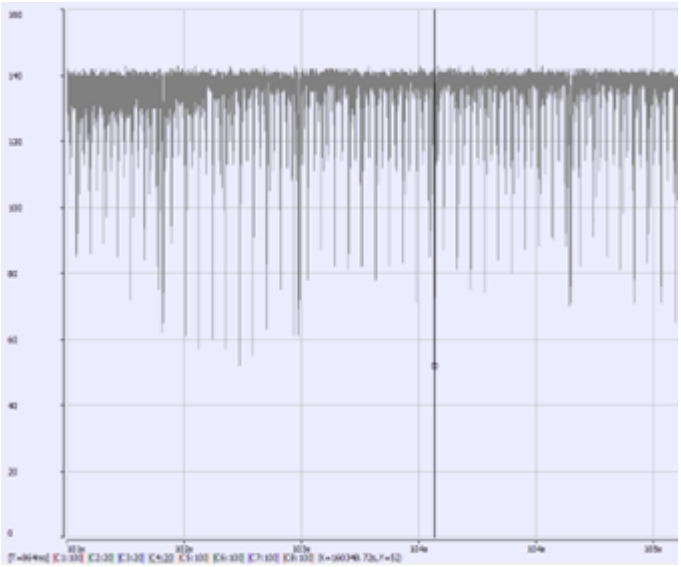


Figure 16-58: PCMM2G **PLCMarginTime**

12.6.2.2 CycleJitter Trace

For the **CycleJitter** trace configure the **Amplitude** and **Zero offset** so you can see the trace **centered** at zero.

- This trace is not too interesting unless a system is misbehaving.
- A jitter of +/-15 microseconds is acceptable.

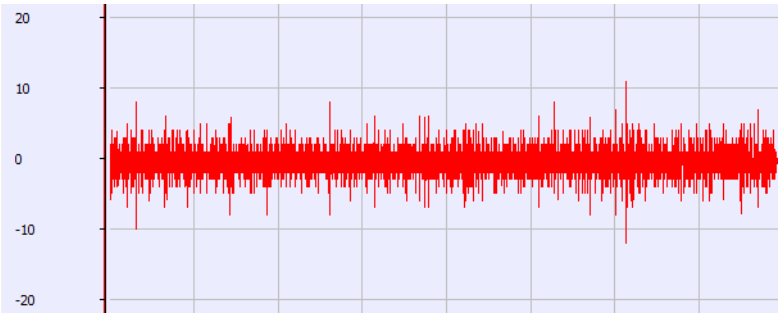


Figure 16-59: Cycle Jitter Trace

See Also

- "Example: PCMM2G Heavily Loaded CPU" (➔ p. 760)
- "Example: PCMM2G Over Loaded CPU" (➔ p. 763)

12.6.3 Example: PCMM2G Heavily Loaded CPU

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This example shows an application that is heavily loading a PCMM2G with the EtherCAT Cycle Time = 250 microseconds.

Specifically, this example looks at:

"1. MotionExec and PLCProgExec Times" (➔ p. 761)

"2. PLCProgExec Peaks" (➔ p. 761)

"3. PLCMarginTime" (➔ p. 762)

"4. Controller Log" (➔ p. 763)

12.6.3.1 1. MotionExec and PLCProgExec Times

Use the techniques described in "Practical Application: Using Trace Time To Measure CPU Load" (➔ p. 756) to examine the **MotionExec** and **PLCProgExec** times.

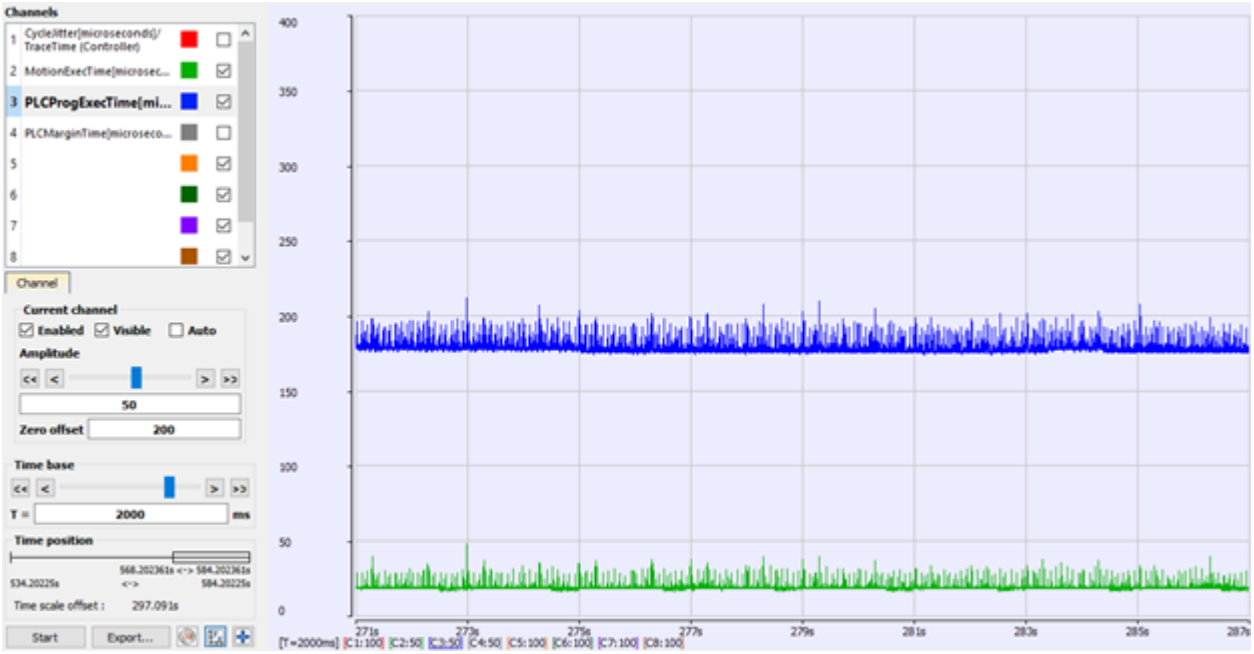


Figure 16-60: MotionExec and **PLCProgExec** Times

- The average **MotionExec** = 20 microseconds.
  - This is about 8% (20 / 250) of the cycle time.
- The average **PLCProgExec** = 178 microseconds.
  - This is about 71% (178 / 250) of the cycle time.
- Thus, the PLC programs have a much higher load.

12.6.3.2 2. PLCProgExec Peaks

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

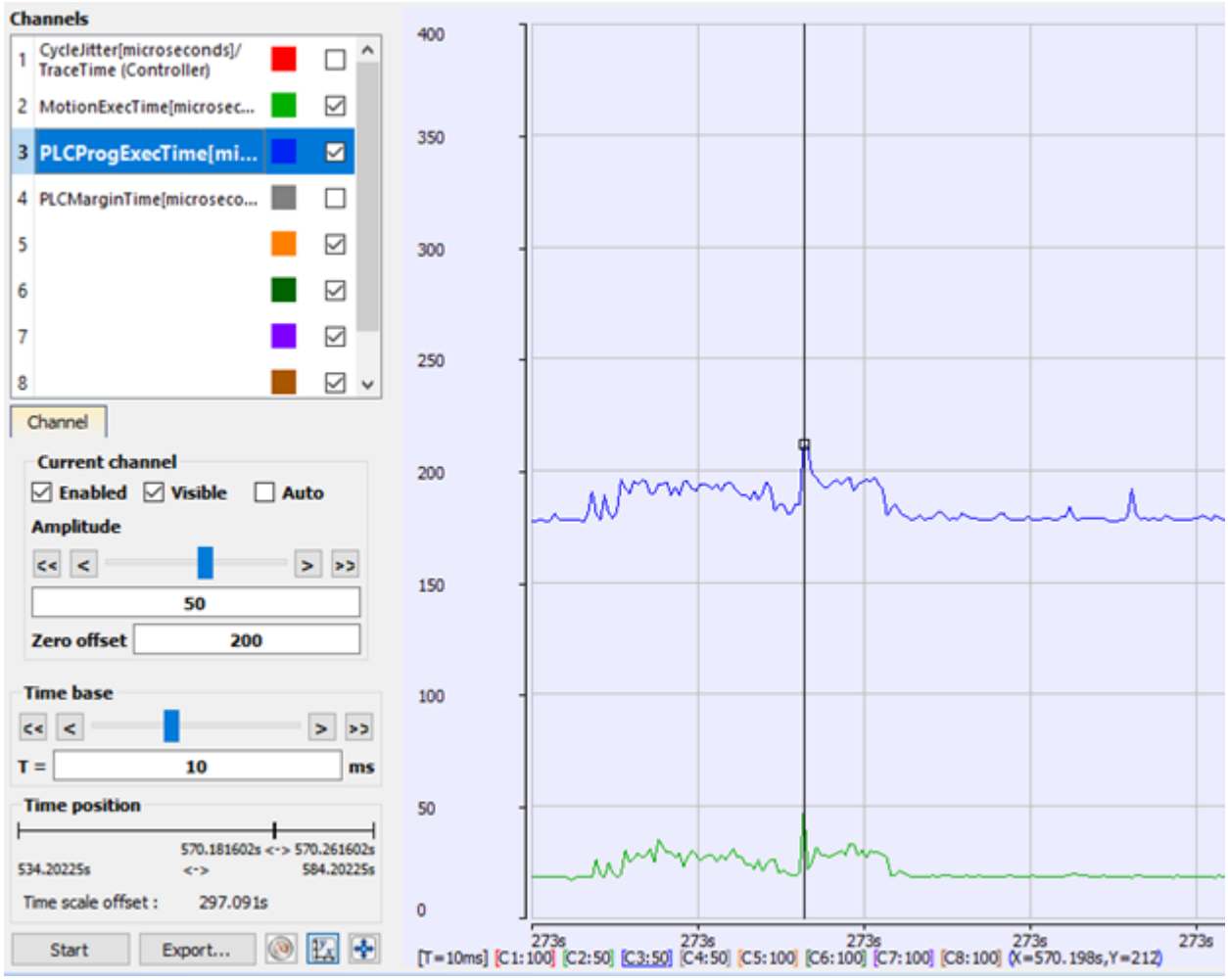


Figure 16-61: PLCProgExec Peaks

This shows there is not much time left over:

- **PLCProgExec** = 212.

12.6.3.3 3. PLCMarginTime

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



Figure 16-62: PLCMarginTime

- The **PLCMarginTime** is between 0 and 25 microseconds (0% to 10%).
- This is not a comfortable margin for deterministic PLC performance.

12.6.3.4 4. Controller Log

The Controller log confirms the occasionally missing PLC execution cycles:

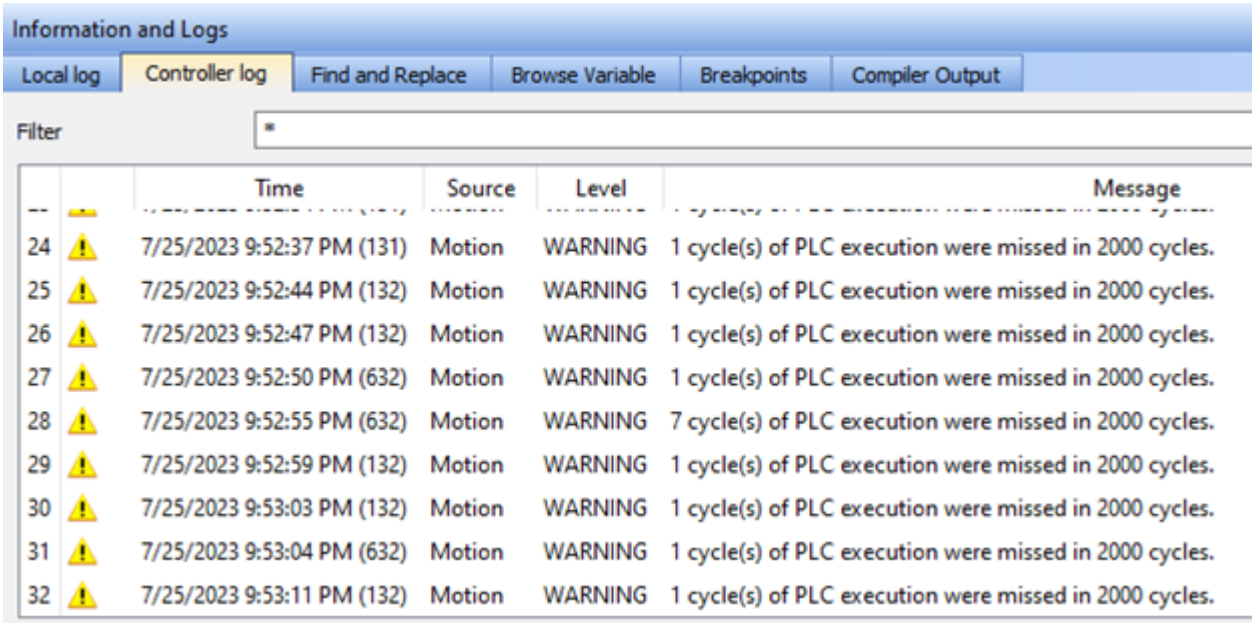


Figure 16-63: Controller Log

See Also

- "Example: PCMM2G Over Loaded CPU" (➔ p. 763)

12.6.4 Example: PCMM2G Over Loaded CPU

This example shows an application that is overloading a PCMM2G with the EtherCAT Cycle Time = 250 microseconds.

Specifically, this example looks at:

"1. MotionExec and PLCProgExec Times" (➔ p. 764)

"2. RealTimeMargin" (➔ p. 764)

"3. Controller Log" (➔ p. 765)

12.6.4.1 1. MotionExec and PLCProgExec Times

Use the techniques described in "Practical Application: Using Trace Time To Measure CPU Load" (➔ p. 756) to examine the **MotionExec** and **PLCProgExec** times.

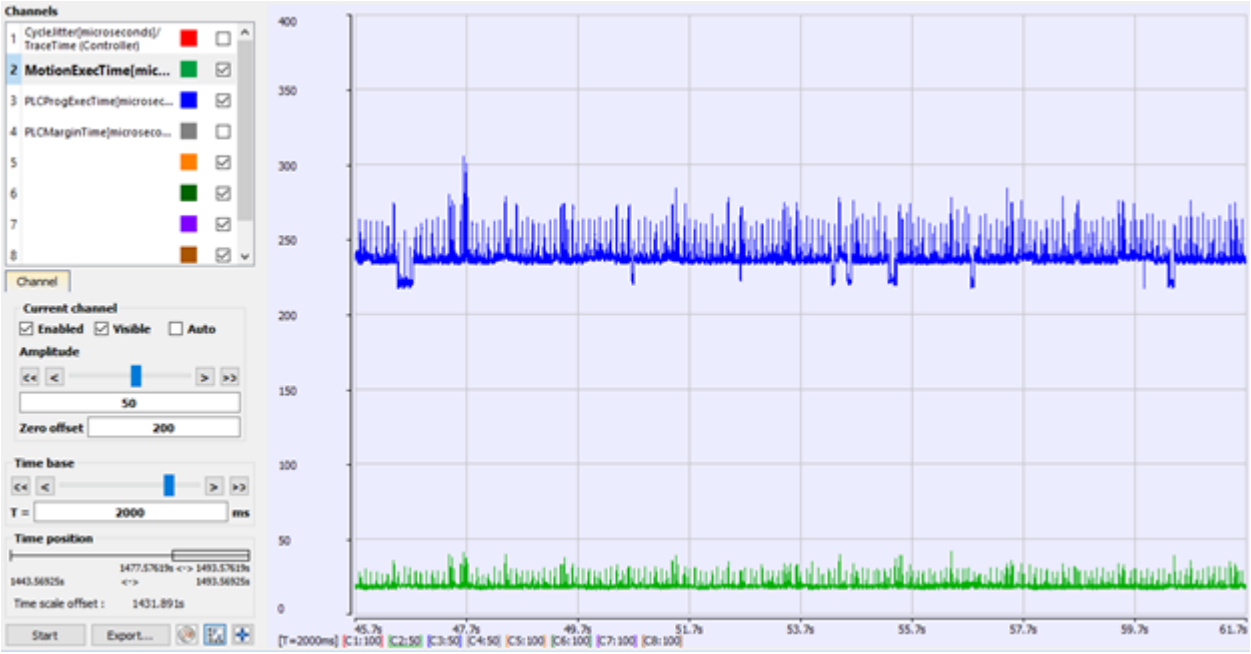


Figure 16-64: MotionExec and **PLCProgExec** Times

- The average **MotionExec** = 20 microseconds.
  - This is about 8% (20 / 250) of the cycle time.
- The average **PLCProgExec** = 235 microseconds.
  - This is about 94% (235 / 250) of the cycle time.

12.6.4.2 2. RealTimeMargin

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

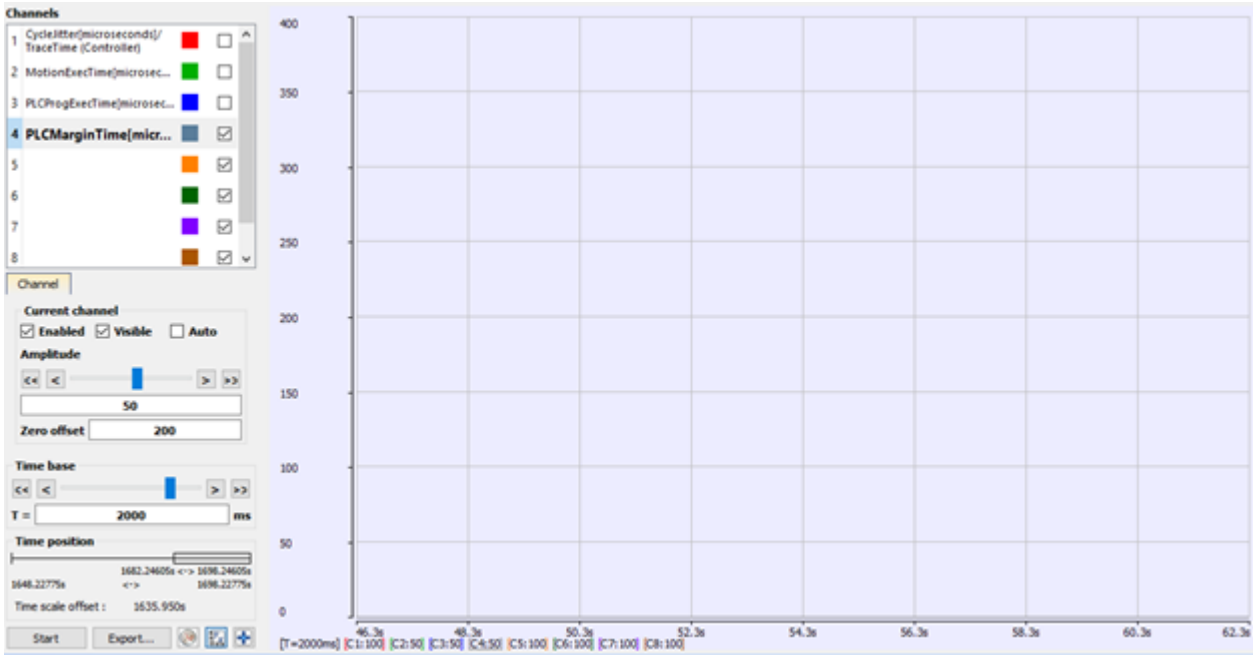
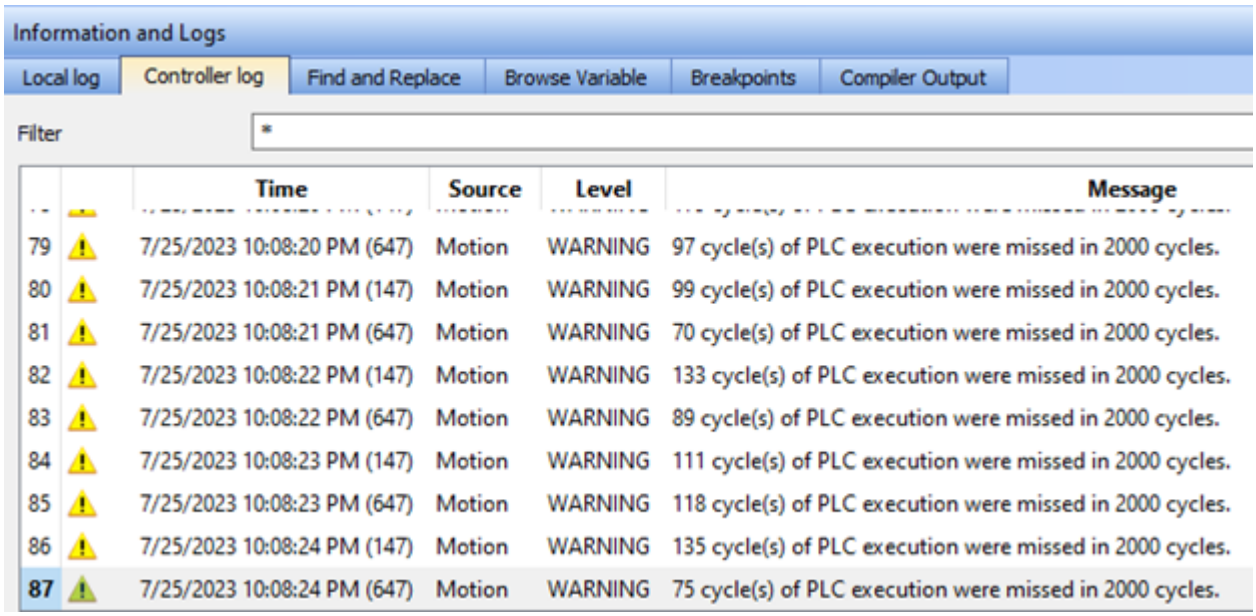


Figure 16-65: RealTimeMargin

- The **PLCMarginTime** is zero.
- There is not enough time to complete the PLC programs every cycle.

12.6.4.3 3. Controller Log

The Controller log confirms the missing PLC execution cycles:



		Time	Source	Level	Message
79	⚠	7/25/2023 10:08:20 PM (647)	Motion	WARNING	97 cycle(s) of PLC execution were missed in 2000 cycles.
80	⚠	7/25/2023 10:08:21 PM (147)	Motion	WARNING	99 cycle(s) of PLC execution were missed in 2000 cycles.
81	⚠	7/25/2023 10:08:21 PM (647)	Motion	WARNING	70 cycle(s) of PLC execution were missed in 2000 cycles.
82	⚠	7/25/2023 10:08:22 PM (147)	Motion	WARNING	133 cycle(s) of PLC execution were missed in 2000 cycles.
83	⚠	7/25/2023 10:08:22 PM (647)	Motion	WARNING	89 cycle(s) of PLC execution were missed in 2000 cycles.
84	⚠	7/25/2023 10:08:23 PM (147)	Motion	WARNING	111 cycle(s) of PLC execution were missed in 2000 cycles.
85	⚠	7/25/2023 10:08:23 PM (647)	Motion	WARNING	118 cycle(s) of PLC execution were missed in 2000 cycles.
86	⚠	7/25/2023 10:08:24 PM (147)	Motion	WARNING	135 cycle(s) of PLC execution were missed in 2000 cycles.
87	⚠	7/25/2023 10:08:24 PM (647)	Motion	WARNING	75 cycle(s) of PLC execution were missed in 2000 cycles.

Figure 16-66: Controller log

- The PCMM2G controller CPU core has plenty of time to execute the Motion Engine and Axes.
  - However, the CPU core running the PLC cannot complete within one Motion/EtherCAT cycle.
- The PLC programs are spanning multiple Motion/EtherCAT cycles.
  - This may cause trouble for application programs that rely on Motion and EtherCAT data updated every PLC cycle.

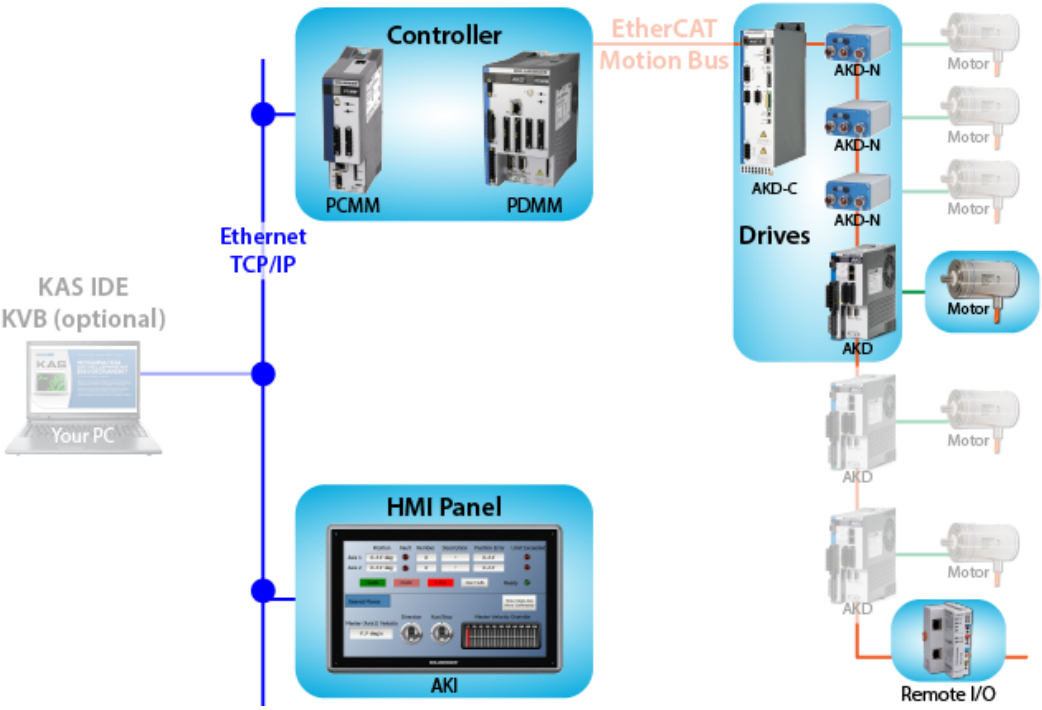
See Also

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Example: AKD PDMM and PCMM Heavily Loaded CPU
- Example: AKD PDMM and PCMM Over Loaded CPU
- "Example: PCMM2G Heavily Loaded CPU" (➔ p. 760)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

13 KAS Component Manuals



13.1 BBH FSoE Master

This BBH Products Safety equipment is required to program and operate the SCU-1-EC FSoE Safety master with these Kollmorgen products:

- AKD2G
- AKT2G-SDI-004-000
- AKT2G-SDO-004-000
- KAS-IDE software
- PCMM2G

Part Number	Description	Image
SCU-1-EC	FSoE Master - modular basic module	
SMX91	Programming adapter (cable)	
SafePLC2	Programming software, License including Hardlock	

Order these items from Kollmorgen.

13.2 Controllers

IMPORTANT

- Controller model capabilities are configured during the manufacturing process.
- Upgrading a PCMM2G to a higher axis count model after shipment requires coordination with the manufacturing facility.
- Customers and distributors should contact their local Kollmorgen support representative for assistance with controller model upgrades or configuration changes.

NOTE

- The maximum axis count specified for each controller model applies only to physical axes.
  - This includes both PLCopen and Pipe Network axes.
- This limit **does not** apply to virtual axis or digitizing axis.

NOTICE

- The PCMM2G firmware image is common across all available controller models.
- Model-specific functionality and axis count limits are determined by the controller configuration and licensing, not by separate firmware images.

Controller	Description	Model Number	Main Characteristics
PCMM2G Essentials	1.5GHz Quad-core	PCMM2G-C2-08-008-00-000	• 0 to 8 axes. • 2nd Generation, Multi-Axis Controller
PCMM2G	1.5GHz Quad-core	PCMM2G-C2-08-024-00-000	• 0 to 24 axes. • 2nd Generation, Multi-Axis Controller
PCMM2G	1.5GHz Quad-core	PCMM2G-C2-08-064-00-000	• 0 to 64 axes. • 2nd Generation, Multi-Axis Controller
PCMM2G (Non-preferred)	1.5GHz Quad-core	PCMM2G-C2-08-008-00-000	• 0 to 24 axes. • 2nd Generation, Multi-Axis Controller **NOTE** • This is a non-preferred legacy model. • Do not use this model with new applications.

13.2.1 Controller Differences

This table list the hardware differences between AKD PDMM, PCMM, and PCMM2G.

See "PCMM2G Hardware Specifications" (➔ p. 769).

13.2.1.1 Hardware Differences

Specification	AKD PDMM and PCMM	PCMM2G
Battery	N/A	• CR2032 Lithium Manganese Dioxide primary cell, 3V. • At least 190mAh.
Certifications	CE, UL	CE, RoHS, UKCA, UL
Critical Temperature Limit (CPU)	127 °C	83 °C
Dimensions	174 mm (H) x 50 mm (W) x 111.5 mm (D)	174 mm (H) x 50 mm (W) x 111.5 mm (D)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Specification	AKD PDMM and PCMM	PCMM2G
External Memory	• Removable SD Card slot. • SD Card not included.	• Removable USB flash drive slot. • USB flash drive not included.
High Temperature Limit (CPU)	120 °C	78 °C
Input Power	24 VDC @ 1.25 A	24 VDC @ 1.25 A
Internal Memory	64MB Flash memory for program storage.	8GB Flash memory
Local I/O	• 6 digital inputs • 2 digital outputs	• 6 digital inputs • 2 digital outputs
Motion Network	EtherCAT, max 4kHz update rate	EtherCAT, max 4kHz update rate
Networking Protocols	1 Ethernet Port 100BaseT connection supporting: • EtherNet/IP • HTTPS - PCMM2G only • Modbus TCP • PROFINET • TCP/IP • UDP	2 Ethernet Ports 100BaseT connection supporting: • EtherNet/IP • HTTPS - PCMM2G only • Modbus TCP • PROFINET • TCP/IP • UDP
Operating Temperature (ambient range)	0 °C to +55 °C	0 °C to +55 °C
Processor	• 800MHz (MC model) • 1.2GHz (M1 model) • Dual-core 1.2GHz (M2 model)	Quad-core 1.5GHz (ARM Cortex A-72 processor)
Sealing	IP20	IP20

13.2.2 PCMM2G Hardware Specifications

NOTE

- The specifications on this page apply to all PCMM2G models.
- See PCMM2G Programmable Controller.

The PCMM2G programmable motion controller delivers a small yet powerful and cost-effective hardware platform.

PCMM2G is ideally suited for modular or stand-alone machines that want the maximum in flexibility and performance.

Specification	Description
Battery	• CR2032 Lithium Manganese Dioxide primary cell, 3V. • At least 190mAh.
Certifications	CE, RoHS, UKCA, UL
Critical Temperature Limit (CPU)	83 °C
Dimensions	174 mm (H) x 50 mm (W) x 111.5 mm (D)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

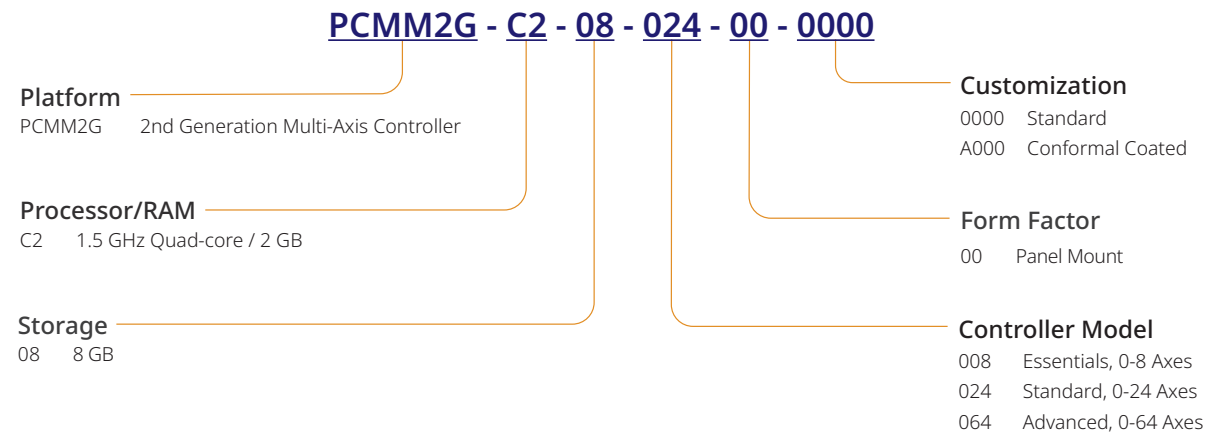
Specification	Description
External Memory	• Removable USB flash drive slot. • USB flash drive not included.
High Temperature Limit (CPU)	78 °C
Input Power	24 VDC @ 1.25 A
Internal Memory	8GB Flash memory
Local I/O	• 6 digital inputs • 2 digital outputs
Motion Network	EtherCAT, max 4kHz update rate
Networking Protocols	2 Ethernet Ports 100BaseT connection supporting: • EtherNet/IP • HTTPS - PCMM2G only • Modbus TCP • PROFINET • TCP/IP • UDP
Operating Temperature (ambient range)	0 °C to +55 °C
Processor	Quad-core 1.5GHz (ARM Cortex A-72 processor)
Sealing	IP20

13.2.3 Part Number Scheme

IMPORTANT

- The part number scheme is for product identification only.
- Do not use for the order process because not all combinations of features are possible.

PCMM2G Model Nomenclature



IMPORTANT

- Controller model capabilities are configured during the manufacturing process.
- Upgrading a PCMM2G to a higher axis count model after shipment requires coordination with the manufacturing facility.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Customers and distributors should contact their local Kollmorgen support representative for assistance with controller model upgrades or configuration changes.

13.2.4 NVRAM

KAS uses NVRAM to save "Retain Variables" (➔ p. 292).

Hardware Type	NVRAM Size Allocation
KAS Simulator	256 Kbytes
PCMM2G	256 Kbytes

IMPORTANT

Part of NVRAM allocation is reserved to store some internal data (144 bytes).
As a consequence, not all of the complete physical NVRAM is available for the retain variables.

If the size is big enough, KAS updates the non-volatile memory to store the retain variables values.

- This operation is performed in the background every 20 seconds.
- The frequency increases to every two seconds when the application is running or when you shutdown the application.

13.2.4.1 NVRAM Life Expectancy

- **Data Retention**: At maximum operating temperature, the data written to the memory is retained for 10 years.
- **Endurance**: NVRAM memory has an endurance limit of 10^{14} read / write accesses.
  - More than 6,000,000 years of being accessed every 2 seconds.

NOTE

- Using the retain variables is very cycle time consuming.
- As a consequence, Kollmorgen strongly recommends to carefully monitor the system load with the TraceTimes command.
- See "Oscilloscope Control Panel" (➔ p. 739).

13.2.4.1.1 Verify the NVRAM Space

To calculate NVRAM space, add the size of each retained variable according to the:

- "Data Types" (➔ p. 286)
- "Arrays" (➔ p. 287) - The numbers of elements in case the variable is declared as an array.

Remember! Add the 144 bytes!

In Figure 17-1, the total size is: **3684 bits** (which is less than 0.5 Kbytes).

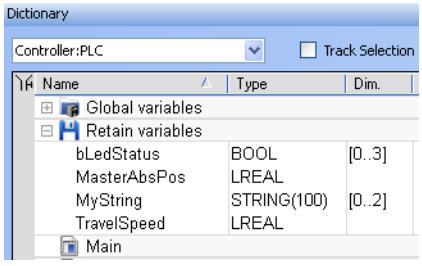


Figure 17-1: Example: Retain variables

Variable	Size / element	Number of Element	Total Size / variable
bLedStatus	1-bit	4	4

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Variable	Size / element	Number of Element	Total Size / variable
Internal data	1152-bits (144 bytes)	N/A	1152
MasterAbsPos	64-bits	1	64
MyString	800-bits (100 bytes)	3	2400
TravelSpeed	64-bits	1	64

13.3 Drives

These websites provide detailed information about drives:

- AKD Series Servo Drives
- AKD2G Servo Drive
- AKD-N Decentralized Servo Drive

13.3.1 Related Documents

See these manuals for additional information on drives:

Drive Manuals	Description
AKD EtherCAT Manual	Describes the installation, setup, range of functions, and software protocol for the EtherCAT AKD product series.
AKD Installation Manual	• Installation manual for AKD and AKD PDMM drives. • Describes the AKD series of digital drives and includes mechanical, electrical, and software installation information needed to safely install AKD.
AKD, AKD2G, S700 (in NA) Accessories Manual	Describes the accessories for Kollmorgen digital drive systems and servo drive motors.
AKD2G EtherCAT Manual	Describes the installation, setup, range of functions, and software protocol for the AKD2G product series.
AKD2G Installation Manual	• Describes the AKD2G series of digital drives. • Includes mechanical, electrical, software, and functional safety options.
MKD Installation Manual	• Installation manual for MKD drive system modules: • MKD-C power supply. • MKD-N servo drives. • It includes mechanical, electrical and software installation information needed to safely install the MKD system modules.
S300 Reference Documentation	Kollmorgen website with access to all S300 manuals.
S700 Reference Documentation	Kollmorgen website with access to all S700 manuals.

13.3.2 AKD-M Kx PDMM Controller / Drive Variant

The AKD-M Kx PDMM is a controller / drive variant introduced in 2022 with a substitute chip.

It is identified with **-Kx** the part number:

AKD-M-xxxxx-KCEC-xxxx

AKD-M-xxxxx-K1EC-xxxx

Contact your local Kollmorgen representative for more information about the AKD-M PDMM.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

13.3.2.1 FAQs for the Kx Controller / Drive

Which AKD firmware versions support the AKD **-Kx** controller / drive?

AKD firmware v01-21-02-000 and later support the **-Kx** controller / drive.

Which versions of the KAS-IDE are compatible with the AKD **-Kx** controller / drive?

KAS-IDE v3.07.0.88325 and later are compatible.

Kollmorgen recommends using v3.07.0.88325 and later because the drive features available in AKD firmware v01-21-02 are integrated with and supported by KAS-IDE v3.07.0.88325.

Are there any backwards compatibility issues with the AKD **-Kx** controller / drive?

EtherCAT Devices Backup and Restore requires AKD firmware v01-21-02-000 at a minimum.

The firmware for the **-Kx** controller / drive is not available in the backup files if the existing backup contains older firmware.

See "EtherCAT Devices Backup" (➜ p. 607) to update the backup files.

Where can I find the KAS firmware package with **-Kx** support?

The AKD firmware and KAS firmware packages are available in the downloads section of Kollmorgen website.

- AKD backup/restore firmware package
- AKD firmware

Which AKD PDMM controller / drive variants can be swapped with the **-Kx** model?

The **Mx** variants are functionally compatible and can be swapped using the "Backup and Restore tab" (➜ p. 606) feature.

What is the recommended usage for **-Kx** controller / drive?

Kollmorgen recommends using the **-Kx** PDMM controller / drives for new machines.

Kollmorgen does **not** recommend using **-Kx** PDMM controller / drives as replacements with existing machines because this may require a drive firmware upgrade and an update to existing EtherCAT devices backup files.

13.3.3 AKD-N "ES" Drive Variant

The AKD-N "ES" is a drive variant introduced in 2022 with a substitute chip.

It is identified with "-ES" the part number:

AKD-N-xxxxx-ESxx-xxxx

Contact your local Kollmorgen representative for more information about the drive.

13.3.3.1 FAQs for the "ES" Drive

Which AKD firmware versions support the AKD-N "ES" drive?

AKD firmware v01-22-00-000 and later support the "ES" drive.

Which versions of the KAS-IDE are compatible with the AKD-N "ES" drive?

KAS-IDE v4.00 and later are compatible.

Kollmorgen recommends using v4.00 and later because the drive features available in AKD firmware v01-22 are integrated with and supported by KAS-IDE v4.00.

Are there any backwards compatibility issues with the AKD-N "ES" drive?

EtherCAT Devices Backup and Restore requires AKD firmware v01-22-00-000 at a minimum.

The firmware for the "ES" drive is not available in the backup files if the existing backup contains older firmware.

See "EtherCAT Devices Backup" (➜ p. 607) to update the backup files.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Where can I find the KAS firmware package with "ES" support?

The AKD firmware and KAS firmware packages are available in the downloads section of Kollmorgen website.

- AKD backup/restore firmware package
- AKD firmware

Which AKD-N drive variants can be swapped with the "ES" model?

The "NB" variants are functionally compatible and can be swapped using the "Backup and Restore tab" (→ p. 606) feature.

What is the recommended usage for "ES" drives?

Kollmorgen recommends using "ES" drives for new machines.

We do not recommend using "ES" drives as replacements with existing machines because this may require a drive firmware upgrade and an update to existing EtherCAT devices backup files.

13.3.4 AKD-P "NC" Drive Variant

The AKD-P "NC" is a drive variant introduced in 2021 with a substitute chip.

It is identified with "-NC" in the part number:

AKD-Pxxxxx-NCxx-xxxx

Contact your local Kollmorgen representative for more information about the drive.

13.3.4.1 FAQs for the "NC" Drive

Which AKD firmware versions support the AKD "NC" drive?

AKD firmware v01-20-00-003 and later support the "NC" drive.

Which versions of the KAS-IDE are compatible with the AKD "NC" drive?

KAS-IDE v2.09 and later are compatible.

Kollmorgen recommends using v3.05 and later because the drive features available in AKD firmware v01-20 are integrated with and supported by KAS-IDE v3.05.

Are there any backwards compatibility issues with the AKD "NC" drive?

EtherCAT Devices Backup and Restore requires AKD firmware v01-20-00-003 at a minimum.

The firmware for the "NC" drive is not available in the backup files if the existing backup contains older firmware.

See "EtherCAT Devices Backup" (→ p. 607) to update the backup files.

Where can I find the KAS firmware package with "NC" support?

The AKD firmware and KAS firmware packages are available in the downloads section of the Kollmorgen website.

- AKD backup/restore firmware package
- AKD firmware

Which AKD drive variants can be swapped with the "NC" model?

The "NB" variants are functionally compatible and can be swapped using the "Backup and Restore tab" (→ p. 606) feature.

What is the recommended usage for "NC" drives?

Kollmorgen recommends using "NC" drives for new machines.

We do not recommend using "NC" drives as replacements with existing machines because this may require a drive firmware upgrade and an update to existing EtherCAT devices backup files.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

13.3.5 AKD-P "ND" Drive Variant

The AKD-P "ND" is a drive variant introduced in 2022 with a substitute chip.

It is identified with "-ND" in the part number:

AKD-Pxxxxx-NDxx-xxxx

Contact your local Kollmorgen representative for more information about the drive.

13.3.5.1 FAQs for the "ND" Drive

Which AKD firmware versions support the AKD "ND" drive?

AKD firmware v01-21-02-000 and later support the "ND" drive.

Which versions of the KAS-IDE are compatible with the AKD "ND" drive?

KAS-IDE v3.07.0.88325 and later are compatible.

Kollmorgen recommends using v3.07.0.88325 and later because the drive features available in AKD firmware v01-21-02 are integrated with and supported by KAS-IDE v3.07.0.88325.

Are there any backwards compatibility issues with the AKD "ND" drive?

EtherCAT Devices Backup and Restore requires AKD firmware v01-21-02-000 at a minimum.

The firmware for the "ND" drive is not available in the backup files if the existing backup contains older firmware.

See "EtherCAT Devices Backup" (➔ p. 607) to update the backup files.

Where can I find the KAS firmware package with "ND" support?

The AKD firmware and KAS firmware packages are available in the downloads section of the Kollmorgen website.

- AKD backup/restore firmware package
- AKD firmware

Which AKD drive variants can be swapped with the "ND" model?

The "NB" and "NC" variants are functionally compatible and can be swapped using the "Backup and Restore tab" (➔ p. 606) feature.

What is the recommended usage for "ND" drives?

Kollmorgen recommends using "ND" drives for new machines.

We do not recommend using "ND" drives as replacements with existing machines because this may require a drive firmware upgrade and an update to existing EtherCAT devices backup files.

13.3.6 MKD-C "ND" Central Power Supply Variant

The MKD-C "ND" is a central power supply variant introduced in 2022 with a substitute chip.

It is identified with "-NC" in the part number:

MKD-Cxxxxxx-NDEC-xxxx

Contact your local Kollmorgen representative for more information about the drive.

13.3.6.1 FAQs for the "ND" Central Power Supply

Which AKD firmware versions support the MKD-C "ND" central power supply?

AKD firmware v01-22-00-000 and later support the "ND" central power supply.

Which versions of the KAS-IDE are compatible with the MKD-C "ND" central power supply?

KAS-IDE v4.00 and later are compatible.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Kollmorgen recommends using v4.00 and later because the features available in AKD firmware v01-22 are integrated with and supported by KAS-IDE v4.00.

Are there any backwards compatibility issues with the MKD-C "ND" central power supply?

The MKD-C “ND” support was introduced in KAS-IDE v4.00 with AKD firmware v01-22.

Previous MKD-C central power supplies are not supported.

Where can I find the KAS firmware package with "ND" support?

The AKD firmware and KAS firmware packages are available in the downloads section of Kollmorgen website.

- AKD backup/restore firmware package
- AKD firmware

Which MKD-C "ND" central power supply variants can be swapped with the "ND" model?

Previous models are not supported and should not be swapped with the “ND” model.

What is the recommended usage for MKD-C "ND" central power supply?

Kollmorgen recommends using MKD-C "ND" central power supply for new machines.

13.4 HMI

NOTE

Download the applicable documentation from here: KAS HMI Panel Downloads.

HMI Part Number	Description	KVB
AKI-CDC-MOD-21T-000	Graphical Display 21.5" TFT LCD, Touchscreen	v2.0
AKI-CDC-MOD-15T-000	Graphical Display 15.4" TFT LCD, Touchscreen	v2.0
AKI-CDC-MOD-12T-000	Graphical Display 12.1" TFT LCD, Touchscreen	v2.0
AKI2G-CDA-MOD-07T-000	Graphical Display 5" TFT LCD, Touchscreen	v2.20
AKI2G-CDA-MOD-05T-000	Graphical Display 7" TFT LCD, Touchscreen	v2.20
AKI2G-CDB-MOD-12T-000	Graphical Display 12" TFT LCD, Touchscreen	v2.20
AKI2G-CDB-MOD-07T-000	Graphical Display 7" TFT LCD, Touchscreen	v2.20

NOTE

See the Kollmorgen Automation Suite™ web page for up-to-date information and materials.
Scan this QR code to access our web site with your mobile device.

13.4.1 HMI Accessories

Various accessories are available to compliment the HMI, including:

- Programming Cable, RS232 to HMI Terminal RS232
- Key cover for 5.7" Graphical Display
- Key cover for 10.4" Graphical Display

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Touch cover for 3.5" Touchscreen Graphical Display
- Touch cover for 5.7" Touchscreen Graphical Display
- Touch cover for 10.4" Touchscreen Graphical Display
- Touch cover for 15.1" Touchscreen Graphical Display
- 512 MB Compact Flash Industrial Grade
- 1 GB Compact Flash Industrial Grade

13.5 Remote Input/Output Terminals

KAS remote I/Os provide a complete spectrum of bus couplers, digital and analog inputs, digital and analog outputs, stepper, counter, and thermocouple modules.

Related Documents

This table is a list of each I/O component available.

I/O Terminal Part Number	I/O Terminal Description
AKT2G-AC-FAN-001	Fan cartridge for EtherCAT and Bus Terminals.
AKT2G-AN-240-000	• 2-channel input terminal PT100 (RTD) for resistance sensors. • 16-bit, 2-, 3-wire system.
AKT2G-AN-400-000	• 4-channel thermocouple input terminal. • Preset to type K, with wire breakage detection. • 16-bit
AKT2G-AN-430-000	• 4-channel analog input. • Parameter capable • -10/0 to 10V • -20/0/+4 to +20mA • 16-bit
AKT2G-AT-410-000	• 4-channel analog output terminal. • 0 to 10V • 1-wire system • 12-bit
AKT2G-AT-425-000	• 4-channel analog output terminal. • -10V to +10V • 4 x 2-wire system • 12-bit
AKT2G-BRC-000-000	Brake Chopper Terminal
AKT2G-DN-002-000	• Up/down counter 24 VDC • 100 kHz • 32-bit counter depth
AKT2G-DN-008-000	• 8-channel digital input terminal 24 VDC. • Filter 3.0ms • 1-wire system
AKT2G-DNH-008-000	• 8-channel digital input terminal 24 VDC. • Filter 10µs • 1-wire system
AKT2G]-DT-008-000	• 8-channel digital output terminal 24 VDC • 0.5 A • 1-wire system
AKT2G-ECT-000-000	EtherCAT Coupler for E-bus terminals

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

I/O Terminal Part Number	I/O Terminal Description
AKT2G-EM-000-000	• Bus end cover for E-bus terminals • Cover for power and E-bus contacts, gray
AKT2G-ENC-180-000	• 1-channel incremental encoder interface • 32-bit
AKT2G-ENC-190-000	• Incremental encoder interface with differential input • 16-bit / 32-bit
AKT2G-PSF-024-000	• Power supply terminal with fuse • 24 VDC
AKT2G-SDI-004-000	• 4-channel digital input terminal 24 VDC. • Safety
AKT2G-SDO-004-000	• 4-channel digital input terminal 24 VDC. • Safety • 0.5 A
AKT2G-SM-L15-000	• Stepper motor terminal • 24 VDC • 1.5 A • Vector control
AKT2G-SM-L50-000	• Stepper motor terminal • 50 VDC • 5A • Vector control

IMPORTANT

- 24-volt power is NOT passed through AKT-AN-200-000 and AKT-AN-400-000 thermocouple modules.
- These are the solutions to get 24 VDC power to devices that need it (e.g., an AKT_AT-410-000 Output module):
  - Place the module requiring 24 VDC before the thermocouple module.
  - Add a power feed module (AKT-PS-024-000 or AKT-PSF-024-000) after the thermocouple module.

I/O Terminal Part Number	Description	Replaced By
AKT-AN-200-000	2-channel thermocouple input module	AKT2G-AN-240-000
AKT-AN-400-000	4-channel thermocouple input module	AKT2G-AN-400-000
AKT-AN-410-000	• 4-channel analog input module • 0 to 10 VDC	AKT2G-AN-430-000
AKT-AN-420-000	• 4-channel analog input module • 0 to 20 mA	AKT2G-AN-430-000
AKT-AN-810-000	• 8-channel analog input module • 0 to 10 VDC	AKT2G-AN-430-000
AKT-AN-820-000	• 8-channel analog input module • 0 to 20 mA	AKT2G-AN-430-000
AKT-AT-220-000	• 2-channel analog output module • 0 to 20 mA	N/A
AKT-AT-410-000	• 4-channel analog output module • 0 to 10 VDC	N/A

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

I/O Terminal Part Number	Description	Replaced By
AKT-AT-420-000	• 4-channel analog output module • 0 to 20 mA	N/A
AKT-AT-810-000	• 8-channel analog output module • 0 to 10 VDC	AKT2G-AT-410-000
AKT-AT-820-000	• 8-channel analog output module • 0 to 20 mA	N/A
AKT-DN-004-000	• 4-channel digital input module • 24 VDC • 3.0ms	AKT2G-DN-008-000
AKT-DN-008-000	• 8-channel digital input module • 24 VDC • 3.0ms	AKT2G-DN-008-000
AKT-DNH-004-000	• 4-channel digital input module • 24 VDC • 0.2ms	AKT2G-DNH-008-000
AKT-DNH-008-000	• 8-channel digital input module • 24 VDC • 0.2ms	AKT2G-DNH-008-000
AKT-DT-004-000	• 4-channel digital output module • 24 VDC • 0.5 A	AKT2G-DT-008-000
AKT-DT-008-000	• 8-channel digital output module • 24 VDC • 0.5 A	AKT2G-DT-008-000
AKT-DT-2RT-000	• 2-channel Relay Output Module • 230VAC • 2.0A Rel.2NO Pot.-Free	N/A
AKT-ECT-000-000	EtherCAT Bus Coupler	AKT2G-ECT-000-000
AKT-EM-000-000	Standard-Bus End Terminal	AKT2G-EM-000-000
AKT-IM-000-000	Isolation / Separation Terminal	N/A
AKT-PS-024-000	Power Supply, 24 VDC	AKT2G-PSF-024-000
AKT-PSF-024-000	Fused Power Supply with diagnostics, 24 VDC	AKT2G-PSF-024-000
AKT-SM-L15-000	• Stepper motor terminal • 24 VDC • 1.5 A	AKT2G-SM-L15-000
AKT-SM-L50-000	• Stepper motor terminal • 50 VDC • 5A	AKT2G-SM-L50-000

13.6 Safety Solutions

13.6.1 KSM-compact Safety Module

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

KSM 5x
KSM 11
KSM 12
KSM 21R
KSM 31

- For 1 or 2 safe axes.
- Maximum of 2 expansion modules.
- Base unit with 16 safe I/O.
- Expandable to a maximum 60 safe I/O.
- 1 safe relay output, expandable.
- 2 pulse outputs, 2 signal outputs.
- Expandable to a maximum 6 pulse and 6 signal outputs.
- Maximum of 800 function blocks.
- Space-saving, compact design.

13.6.2 KSM-modular Safety PLC

KSM 100-1
KSM 100-2
KSM 100-4
KSM 121/-2
KSM 122/-2
KSM 122A
KSM 131
KSM 131R

- Maximum of 12 safe axes.
- Maximum of 8 expansion modules.
- Base unit with 56 safe I/O.
- Expandable to a maximum of 200 safe I/O.
- 1 safe relay output, expandable.
- 2 pulse outputs.
  - Maximum of 10 signal outputs.
- Expandable to a maximum of 14 pulse and 22 signal outputs.
- Maximum of 3000 function blocks.
- For applications with large number of interfaces.

13.6.3 Safety Functions (PLe & SIL3)

- Safe Brake Function: SBC
- Safe Direction Function: SDI
- Safe Position Function: SLP, SCA, SLI
- Safe Stop Function: STO, SS1, SS2, SOS
- Safe Velocity Function: SLS, SSM, SSR, SMS

Learn more about the safety functions on the Kollmorgen Developer Network.

13.6.4 Connectivity

- CANopen
- EtherCAT
- PROFINET
- PROFIsafe and EtherCAT FSoE slave

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

14 Troubleshooting

These topics may prove useful in troubleshooting:

- "FAQs" (→ p. 781)
- "Compiler Errors" (→ p. 795)
- "Compiler Warnings" (→ p. 800)
- "Connect Remotely" (→ p. 803)
- "CPU Load Reduction Techniques" (→ p. 804)
- "Error Management" (→ p. 857)
  - This topic provides a programmatic procedure for when a non-fatal error occurs in the machine and motion needs to be halted quickly.
- "EtherCAT Coupler Error Handling and Diagnosis" (→ p. 804)
- "EtherCAT Diagnostics and Errors" (→ p. 807)
- "Practical Application: Using Trace Time To Measure CPU Load" (→ p. 756)
  - This topic provides a procedure for calculating CPU usage.
- "Re-import Kollmorgen UDFBs" (→ p. 812)
- "Troubleshoot KAS-IDE and SafePLC2" (→ p. 816)
- "Troubleshoot FSoE Safety Parameters" (→ p. 813)

14.1 FAQs

- "Basic" (→ p. 781)
- "Computer / Controller" (→ p. 786)
- "EtherCAT" (→ p. 788)
- "KAS-IDE Crashes" (→ p. 790)
- "Performance" (→ p. 790)
- "Programming" (→ p. 792)
- "Setup" (→ p. 794)

14.1.1 Basic

Why does the Installer not Start when I insert the CD?

The Autorun feature may be deactivated.

1. Open an Explorer window to see the autorun.exe file.
2. Use the Run command in the contextual menu to manually start the installer.

Can I install more than one version of the KAS-IDE?

That depends on the version of KAS.

You may have one version of KAS-IDE v3.00 and earlier installed along with any number of later versions.

KAS Version	Only 1 Version	Multiple Installed Versions
earlier than v3.00		
v3.01 and later		

Can I run more than one version of the KAS-IDE simultaneously?

Yes, as long as one of the versions is v3.02 or later.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- With the release of v3.02, multiple instances of the KAS-IDE are permitted.
- This includes:
  - Running one older (earlier than v3.02) instance and any number of newer instances.
  - Multiple instances of the same version.
- The only limitation is available RAM.



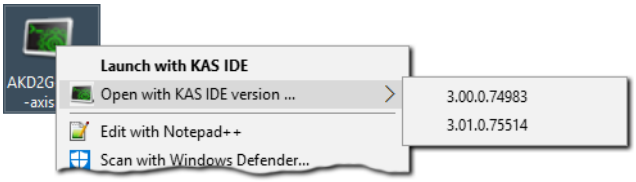
TIP

- The KAS-IDE supports simultaneously running multiple instances.
- Double-click a shortcut or project file, or selecting the application from the Start menu, opens another instance.
- Double-click a project file more than once opens a copy of the project in another instance of the KAS-IDE.

How do I launch my project with a particular version of the KAS-IDE?

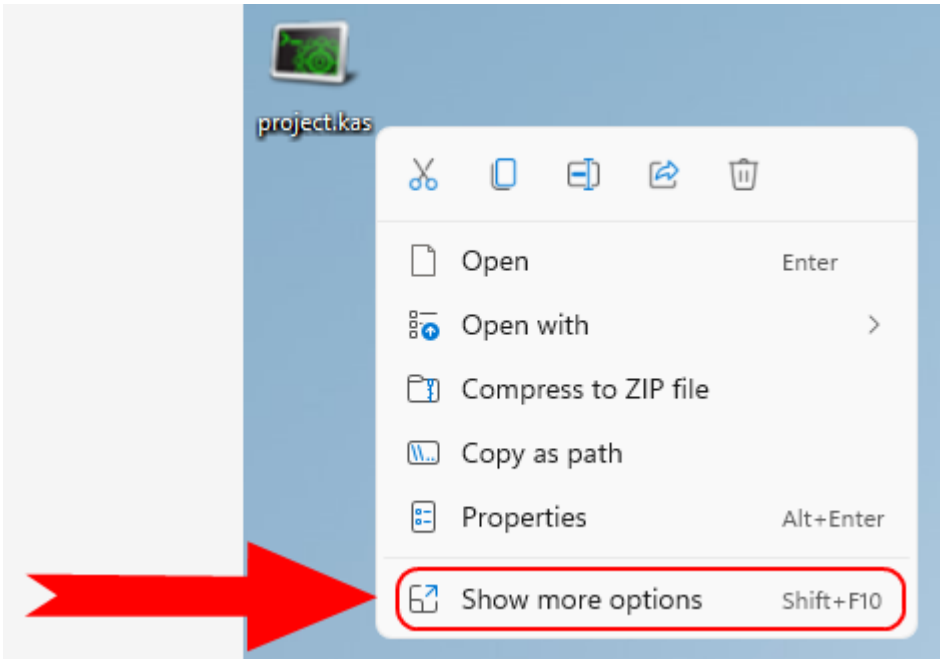
Window 10 and Windows 11

Right-click a KAS project file to choose which of your installed versions of the KAS-IDE to use for that instance.



Windows 11

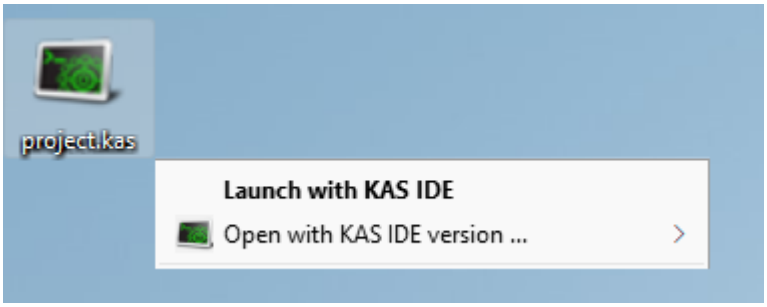
Right-click a KAS project file and click **Show more options**.



Or

Select the KAS project file and press <Shift+F10>.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



Why does the context menu for my project file not display the “Open with KAS IDE version...” option?

The option for “Open with KAS IDE version...” does not appear if KAS-IDE has never been installed on the computer for **All Users** (after KAS-IDE v4.04).

- In this case, some libraries may not be registered and the context menu option is not enabled.
- To enable the context menu option, either:
  - Complete an installation of KAS-IDE from v4.03 or earlier.
  - Select the **All Users** option on the Choose Users window during the KAS-IDE installation.

NOTE

The **All Users** option requires Administrative Privileges.

How do I choose the default version of the KAS-IDE to use?

The default version of the KAS-IDE is the one most recently installed.

This can be changed to a preferred version by changing an Environment Variable.

1. In Windows search field, type Environment.
2. Click the **Edit environment variables for your account** option.
   The Environment Variables dialog box opens.
3. From the list of **User** variables, select KAS_INSTALL_DIR and click **Edit**.
   The **Edit User Variable** dialog box opens. (Figure 18-1)

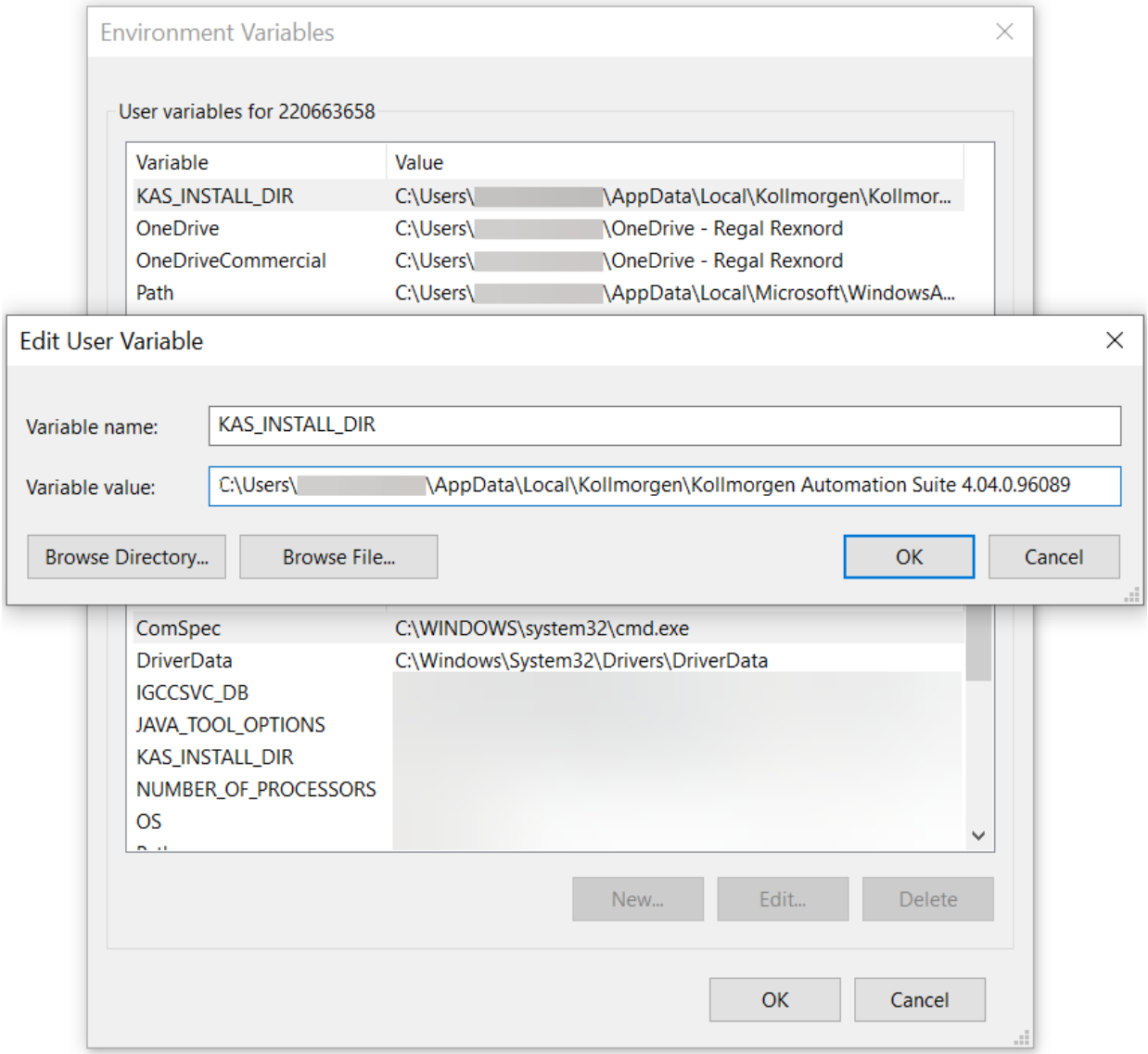


Figure 18-1: Edit User Variable dialog box

4. In the **Variable value:** text box, enter the path to the preferred version.
   Option: Click **Browse Directory** to search for the location and select the version.
5. Click **OK** to save the changes or selections and close the dialog box.

Can I copy and paste or drag and drop between instances of the KAS-IDE

Yes, the KAS-IDE supports copying and pasting between simultaneously running instances.

- This is supported in both the same version and different version numbers.
- This includes drag and drop for supported objects (e.g., FBD blocks).

The KAS-IDE attempts to recreate what you have copied, but there are some limitations.

Example: It may not automatically associate variables with a function.
If you copy and paste only a function block from an FFLD program, the assigned variables is not copied and pasted.
If you select the entire network, the KAS-IDE automatically creates the variables for you.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

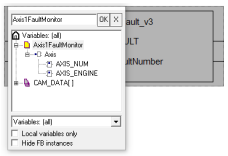


Figure 18-2: Copying a block

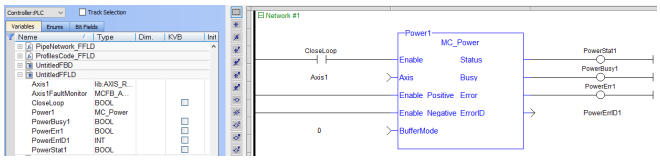


Figure 18-3: Copying an entire network recreates assigned variables in the Dictionary

IMPORTANT

- Closing the KAS-IDE automatically clears the clipboard.
- Anything that was copied is cleared from memory.
- This means you can not copy something, close the KAS-IDE, and then paste.

Why does the KAS-IDE not display all the items in the Project Explorer when I create a new project based on a template?

A side effect with some remaining files that were not deleted properly can interfere with your new project. To fix this issue:

1. Close your current project without saving.
2. Open Windows Explorer and go to **C:\Documents and Settings\(user)\Local Settings\Application Data\Kollmorgen\KAS\Project**, where “(user)” is the Windows' username you are currently logged in with.
3. Delete all the remaining files and folders.
4. You can now create your new project.

Where can I get the latest User Manuals?

Everything in our manuals may be found in the online help.

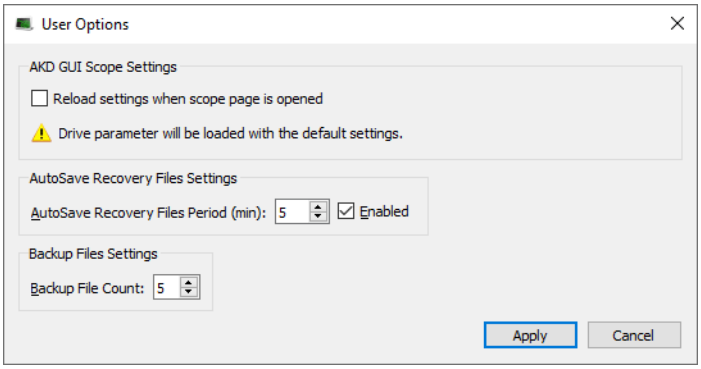
If you need a printable document or want stand-alone manuals, the documentation is available as PDFs on KDN.

See "Learning Kollmorgen Automation Suite" (➔ p. 58).

Can I automatically save my files?

The KAS IDE does have an AutoSave feature which automatically backs up your crash recovery files at a specified interval.

This is set in the **Tools** menu.



How does file recovery work?

With the AutoSave function, project files are saved for recovery.

More than one file may exist due to multiple versions of the KAS-IDE being installed or used.

- When you restart the KAS-IDE you are prompted to recover a found file.
- When there is more than one file, the recovery process starts with the latest date from the newest KAS-IDE version.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Assuming you choose to recover the file it is opened and you may resume work but the older recovery files still exist.

The next time you launch the KAS-IDE, you are prompted to recover the next oldest file. Choosing to not recover the file will delete it, and you are prompted to open the next oldest file until none are left.

14.1.2 Computer / Controller

Why are my Recent Projects different in KAS-IDE v4.04 and later than from previous KAS-IDE versions?

- Starting with v4.04, a new settings file is used to store recently accessed files, recently searched variables, and screen dimensions. (Figure 18-4)
  - This is due to an incompatibility with non-ASCII character support introduced to the application.
- This means that, upon the first launch of an instance of KAS-IDE using v4.04 or later, these stored settings will be reset.
- All versions of v4.03 or earlier are not affected by this change.

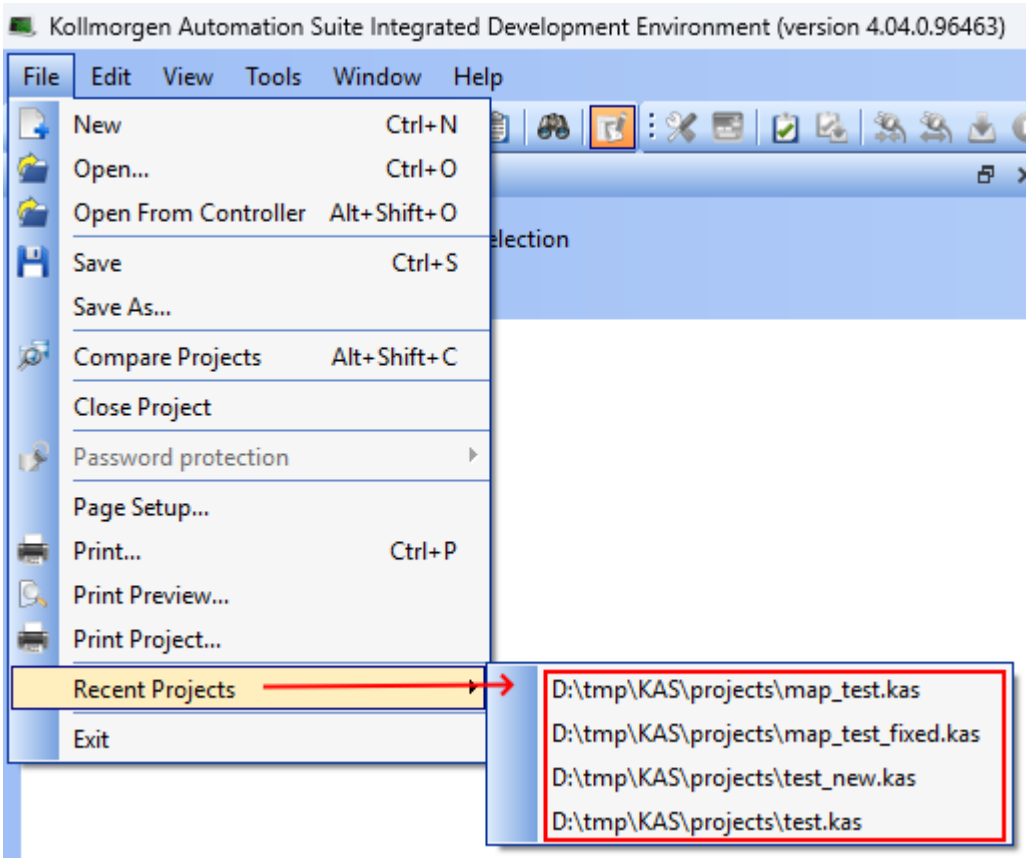


Figure 18-4: KAS Recent Projects list

How many axes can the KAS-IDE manage in 1ms?

- This number is mainly dependent upon the application and your PC's computing power.
- An average number would be 20 axes/ms.

I cannot view the controller's webpage or offline documentation in the KAS-IDE? What do I do?

IMPORTANT

- The KAS software uses Microsoft WebView2 Runtime software to display the documentation and web server pages from the KAS-IDE.

Troubleshooting WebView2

Follow these procedures to resolve most issues related to Microsoft WebView2.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

These procedures are for Windows 10.
Administrator rights are required for these procedures.

1. **Verify Installation**
   a. Press <Windows + I> to open Settings.
   b. Click Apps.
      The list of installed apps appears.
   c. Search for Microsoft Edge WebView2 Runtime.
   d. If it's listed, verify it's the latest version.
      If it's outdated or missing, reinstall it using the Installation procedure.
2. **Repair the WebView2 Installation**
   a. Press <Windows + I> to open Settings.
   b. Click Apps.
      The list of installed apps appears.
   c. Search for and select Microsoft Edge.
      If Microsoft Edge is not installed, download it from Download Microsoft Edge.
      If Microsoft Edge is installed, continue with Step 2d.
   d. Click Modify.
      The Repair Microsoft Edge dialog box opens.
   e. In the dialog box, click Repair.
   f. In the list of apps, search for Microsoft Edge WebView2 Runtime.
   g. If it's listed, verify it's the latest version.
      If it's outdated or missing, reinstall it using the Installation procedure.
3. **Verify Microsoft Edge is Updated**
   a. Open Microsoft Edge.
   b. Go to Settings > About Microsoft Edge.
   c. Let it check for updates and install them.
   d. Restart Microsoft Edge and your PC.
4. **Check for Windows Updates**
   a. Go to Settings > Windows Update.
   b. Install all pending updates, including optional and driver updates.
   c. Restart your PC.

How can I see the CPU load between the PLC and motion parts?

This procedure allows you to determine if your controller is overloaded due to the PLC program or motion system load.

You can use the Softscope and the **Trace Times** button to display these CPU loads:

- Cycle Jitter (Microseconds).
- Motion execution time (Microseconds).
- PLC execution time (Microseconds).
- PCMM2G: (Microseconds).

To view the load:

1. Open the Softscope.
2. Plug four probes to any kind of data.
   See "Map a Variable to the Oscilloscope" (➜ p. 750).
3. In the Control Panel, click the TraceTimes button.
   See "Oscilloscope Control Panel" (➜ p. 739).

I cannot log into the controller's webpage. What do I do?

If you are unable to log into the controller's webpage using a valid user ID and password, the controller's file system may be corrupt.

- To reset the file system, follow the instructions in the "File System tab" (➔ p. 666) to restore the controller to factory settings.
- If you have downgraded from v5.00 to an earlier version, your credentials will revert to the default values because of the password security implementations introduced in v5.00.

If the issue persists, use the push buttons on the controller to fully recover the system.

- For PCMM2G, use the "Production Image Recovery" (➔ p. 672).

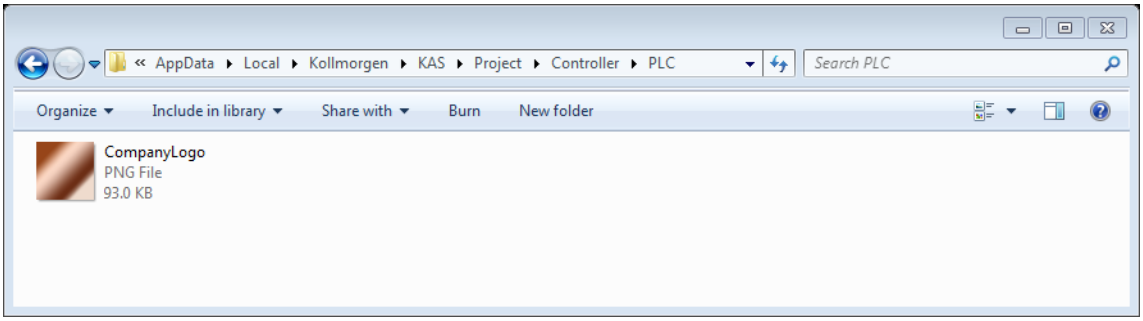
How can I check if there is enough NVRAM space to store my Retain Variables?

See "NVRAM" (➔ p. 771).

I have custom graphics in my project's control panel. Why don't they show up when the file is opened on other computers?

Graphics need to be made portable with a project, otherwise the KAS-IDE attempts to point to the original location of the graphic files.

To make graphics portable, copy them to `C:\Users\[user folder]\AppData\Local\Kollmorgen\Project\Controller\PLC`. After this the graphics are a part of the .KAS file.



Why is the timestamp on the controller logs incorrect?

On the PCMM2G, the time is incorrect if all these conditions are TRUE:

- The controller looses power.
- Network Time Protocol (NTP) is disabled.
- There is no battery backup.

14.1.3 EtherCAT

Why am I getting an E33 error?

An "E33" (➔ p. 581) occurs when the EtherCAT network fails to start when first loading a KAS project.

This typically occurs when the devices on the EtherCAT network do not exactly match the ENI file inside the KAS project.

- This can mean that devices were added, removed, or put in a different order.
- If this happens with a machine in the field that was working, it means that one or more devices are no longer being seen on the EtherCAT network because they are not powering on or an Ethernet cable came unplugged or broke.

Error Correction:

- Check the EtherCAT network wiring and devices state.
- Re-start the application.
- If you have the KAS-IDE software, you can scan the EtherCAT network to see if all the devices are found and in the correct order.

See "ENI and ESI Files" (➔ p. 424).

Is EtherCAT limited to Master/Slave applications?

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

No. As with every real-time Industrial Ethernet system, one device (the master) has to be in charge of the network management and organize the Medium Access Control. With EtherCAT, Slave-to-Slave communication is supported either with:

- Topology dependent within one communication cycle ("upstream" device talks to "downstream" device).
- Topology independent within two cycles.

Since EtherCAT is so much faster than competing systems, slave-to-slave communication, using two cycles is faster.

What is the maximum number of EtherCAT nodes per network?

The maximum number of node addresses is 65,535.

Typically, the number of the nodes on the network is limited by the maximum EtherCAT frame size, which is 1500 bytes.

See the Frame Size in the "EtherCAT Master Settings tab" (→ p. 539).

When can I expect my SDO command to be completed?

If you need to rely on SDO communication to set the parameter of an EtherCAT device, you can do this with the ECATWriteSdo FB

Being asynchronous and based on the EtherCAT mailbox, the SDO communication is not deterministic.

- The EtherCAT master uses a polling mode to ensure the SDO command is completed.
- In operational mode, this polling is performed every 50 cycles [1].
  - As a consequence, you can expect the acknowledgment of your SDO command usually before less than 100ms.
- A good practice is to set the update rate for SDO communication in your PLC application each 25 cycles.

See "EtherCAT Motion Bus Concepts" (→ p. 416).

When I compile I get a warning about the total number of PDO entries. How do I fix this?

The AKD2G has a limit on the number of PDO objects which can be mapped, depending on the sample rate.

Sample Rate	Cycle Time	Max. PDOs
4kHz	250µs	14 Rx, 14 Tx
2kHz	>500µs	30 RX, 30 Tx

This limitation can be hit during compilation with this error:

```
EtherCAT: ERROR: In AKD2G_1, the total number of entries for all selected
Tx PDOs is greater than the max allowed at a cycle time of 250 µs: 17 > 14
```

There are three possible solutions.

- Increase the Cycle Time - see "EtherCAT Master Settings tab" (→ p. 539)
- Deselect unused PDOs - see "PDO Selection/Mapping tab" (→ p. 554)
- Delete unused objects - see "Delete a CoE Object" (→ p. 565)

Why is the PLC execution rate not the same as the EtherCAT rate with the KAS Simulator?

When the application runs on the KAS Simulator, the PLC execution rate is approximately 10ms.

[1] To avoid overloading the controller, this rate is set according to the communication load, as well as the duration the AKD takes to process commands

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

KAS Simulator cannot execute the PLC programs faster because Windows is not able to handle timing less than 10ms.

14.1.4 KAS-IDE Crashes

Intel Display Adapters and WorkBench Windows

There is a known bug in the driver code for some Intel display adapters.

- This bug can cause a crash when an AKD Drive Configuration window is opened.
- This bug was introduced to the UHD Graphics 770 driver sometime after version 30.0.101.1218.
  - It was confirmed that the crash does not occur with this version.
  - Rolling back to this version could be problematic, so it is not recommended.
- Some information can be found here: EXCEPTION_ACCESS_VIOLATION_READ Crashes in igd9trinity32 - Intel Community.
- These GPUs are known to have the defective driver:
  - UHD 730
  - UHD 770
  - Intel Arc
  - Iris Xe Graphics

Possible Workarounds

1. Use the standalone WorkBench instead of the WorkBench views in the IDE.
2. Disable onboard graphics and use an external graphics card
3. Disable WPF hardware acceleration globally across the entire machine via the Windows Registry.
   a. Open the Windows Registry
   b. Navigate to: HKEY_CURRENT_USER\SOFTWARE\Microsoft\Avalon.Graphics
   c. Create a DWORD value named DisableHWAcceleration and set it to 1.

14.1.5 Performance

Why does the PCMM2G take a long time to boot from the recovery image?

It is generally recommended to use USB 3.0 (USB - Wikipedia) or higher standard on the PCMM2G to have a better performance.

The normal recovery image boot up is around 30 to 40 seconds.

- If the controller 7-segment display is  for a longer period of time (more than two minutes), this delay could be because the recovery image is being flashed to an incompatible USB flash drive.
- This delay may not prevent the user from following the recovery procedure.
- See "Boot from Recovery Image (B2 Button)" (➔ p. 602).

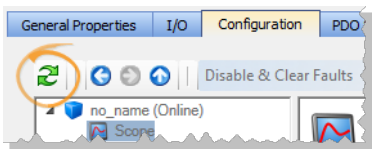
Contact Kollmorgen Technical Support if the issue is still not resolved using a USB 3.0 or higher.

I am having trouble with the WorkBench navigation tree. What can I do?

Due to excessive ASCII SDO commands or very slow Ethernet communication, the WorkBench tree may not fully populate or navigation may be unresponsive.

To recover from this situation:

1. Click the **Refresh Tree** button in KAS-IDEWorkBench view.



2. Disconnect and Reconnect the KAS-IDE to restart KAS-IDE-WorkBench communication.

What can I do if the KAS-IDE's embedded WorkBench (drive Configuration tab) is slow or unresponsive?

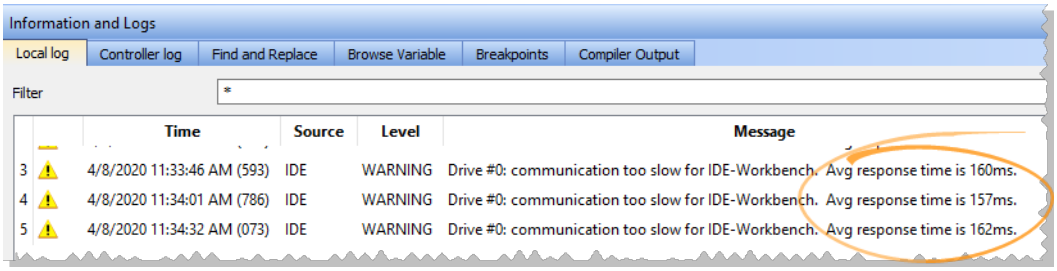
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The status bar may display a warning if communication between the KAS-IDE, controller, and drive is too slow.



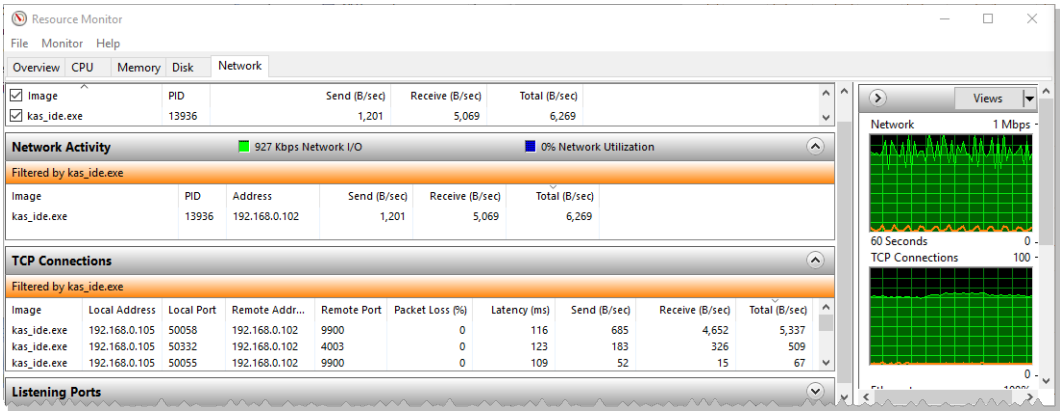
To begin troubleshooting, configure the Local log for **WARNING** level (or lower) and check for **communication too slow** messages.

The IDE-WorkBench navigation and views may become very slow or completely unresponsive if the average response time for drive communication is too long.



Communication delays between the KAS-IDE and Controller can be caused by degraded Ethernet bandwidth or high latencies.

You can use the Windows Resource Monitor (resmon.exe) and select **kas_ide.exe** to monitor and diagnose high latency and/or poor bandwidth.



Communication delays between the Controller and Drive can occur while an application is running on the controller.

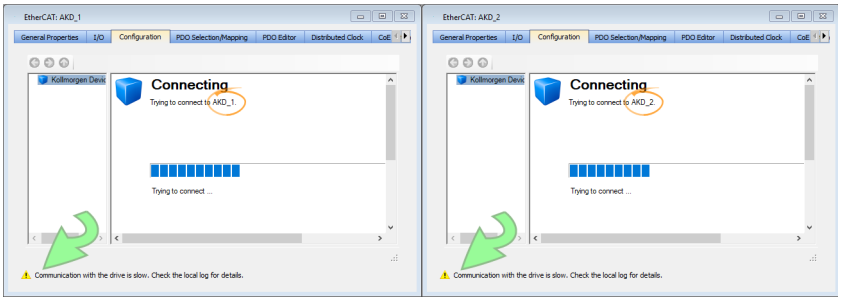
Degraded drive communication can be caused by excessive ASCII SDO commands from the application code.

- Check your application code for calls to DriveParamRead/Write(...), DriveParamStrRead(...), or ECATRead/WriteSdo(...), and minimize the frequency of these calls.
- Use "Online Configuration Mode" (➔ p. 471) with KAS-IDEWorkBench to avoid ASCII SDO channel competition between your application and KAS-IDEWorkBench

I am having trouble connecting to drives with multiple WorkBench views /windows open. What can I do?

There are times when trying to connect to multiple drives simultaneously with WorkBench is problematic, particularly on slower networks.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



14.1.5.1 Resolution

- Close all of the open windows in the IDE workspace, leaving only one WorkBench window open.
- Connect the KAS-IDE and run the project or use Online Configuration mode.

These actions reduce the network and KAS-IDE loads. This allows the embedded WorkBench to establish communication and access the views.

To access multiple drives at the same time:

1. Open one WorkBench window
2. Wait for communication and initialization to complete successfully
3. Open each additional WorkBench window one at a time.
   Allow enough time for each to successfully complete initialization.

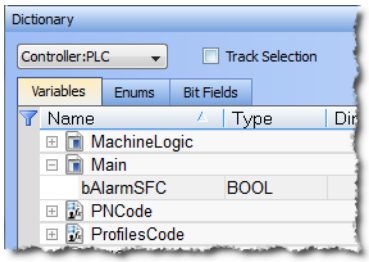
14.1.6 Programming

How can I control the time execution for an SFC step?

When you want to check the maximum time execution for an SFC step, you have to program this action based on the SFC alarm capability.

To show this status:

1. In the Dictionary, declare a Boolean PLC variable linked to the related SFC program.



2. Add the instruction in the Actions tab related to the SFC step, with first parameter set to **A** (for Alarm).

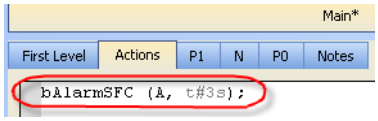


Figure 18-5: SFC Step - Timeout Alarm

How can I fix the Library Access issue?

If you open a project containing a link on a library which is no longer available, a warning is displayed. To fix this issue, refer to "What happens when a library no longer exists?" (→ p. 246)

What is the Fast Input?

The Fast Input allows an application to get information about the occurrence of an external event at a higher resolution than the cycle time.

See "Fast Inputs with Pipe Network Motion" (→ p. 859).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

How do I implement feedback?

There are two kinds of feedback:

14.1.6.1 Primary Feedback

- With a S300 drive you can use a resolver for primary feedback.

Secondary Feedback

- If a secondary feedback is required with your S300 drive, you can use a BiSS feedback device.
  - If you use the same setup with an S300 drive, the S300's EXTPOS parameter has to be set to **-11**.
    IMPORTANT: do not omit the negative sign!
  - To access secondary feedback, use a SAMPLER Pipe Network block.
- Use the MLSmpConPNAxis, MLSmpConPLCAxis, or MLSmpConECAT functions to configure the block.

How do I implement Torque Feed-forward?

Current drives that support torque feed-forward arethe S300 and AKD drives.

To use torque feed-forward, you have to rely on a CONVERTER Pipe Network block.

Use the MLCNVConnectEx function to configure the block.

The arguments must be:

- The Pipe Network block ID being configured.
- The ID of the axis to which the torque feed-forward is applied.
- The constant EC_ADDITIVE_TORQUE_VALUE.
- An ignored integer value (usually set to zero).

See:

- Torque Feed-forward
- Guidelines for Choosing feed-forward Control in Industrial Applications
- Tuning with Feed-forwards
- Measurement-based Feed-forward Tuning

How is Torque Feed-forward Scaled?

If I measure a number (e.g., 500) as an input at the Convertor block which is connected with the PDO object (Additive Torque Value 0x60B2), how many Amps are fed in the current loop at the AKD?

```
Current loop feed-forward value =  Rated current x IL.KBUSFF x input at converter block / 1000
```

Example

With an AKD where:

- Rated current = 3A
- IL.KBUSFF = 1.0
- Additive Torque (PDO object) = 500 units

Then:

```
IL.FF = 3 x 1.0 x 500 / 1000
IL.FF = 1.5A
```

What are the limitations with cams?

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

There is no limitation with the cams, the number of cams, the number of cam points, etc.

The limitation is only given by the processing power of your PC.

If a variable is associated with an I/O point value, would it get automatically updated?

Yes, I/O points represent the state of real world values.

How does the Pipe Network engine interact with a PLC program?

See "Use Motion Function Block for Pipe Network" (➔ p. 337) for this information.

When I pressed Ctrl-V to paste Pipe Network blocks, nothing happened.

The system needs to know where to place the copied blocks.

Press Ctrl-V then use the mouse to point to where you want the copied blocks to be pasted.

How can I capture the correct pipe position?

There are network and mechanical delays in systems that make capturing a correct pipe position difficult.

To remove the delays, use a sampler block to capture the actual position of an axis.

Figure 18-6 shows how this works:

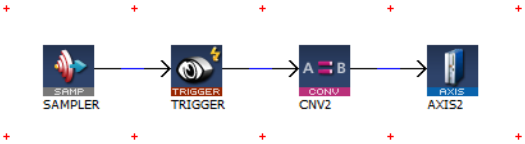


Figure 18-6: Example: Trigger Block

- This sampler block is configured to capture the actual position of axis 1.
  - The conversion and axis blocks are only needed to terminate the pipe.
  - MLTrigWriteDelay may still be needed to compensate for sensor delay.
- If you need to calculate the master pipe value corresponding to the captured axis position, add the difference between the axis command position and the master pipe value.

A hardware limit flag is being thrown by a Pipe Network axis at the start of my program.

If a hardware limit flag is being triggered at the program startup by a Pipe Network axis, MLAxisReAlign may need to be called immediately before motion occurs.

In this case, the distance to travel should be at least as long as the maximum distance the hardware limit flag can remain active.

14.1.7 Setup

How are fieldbuses connected to the KAS Runtime?

The fieldbus serially links all the drives to the industrial PC.

NOTE

When the KAS-IDE is used to deploy an automation system on a master drive (also known as programmable drive), the fieldbus serially links all the drives to the master drive.

How does Kollmorgen Automation Suite communicate with a Host?

KAS can communicate with outside world through CANopen, DeviceNet, or Ethernet.

See "Communication and Fieldbus" (➔ p. 79).

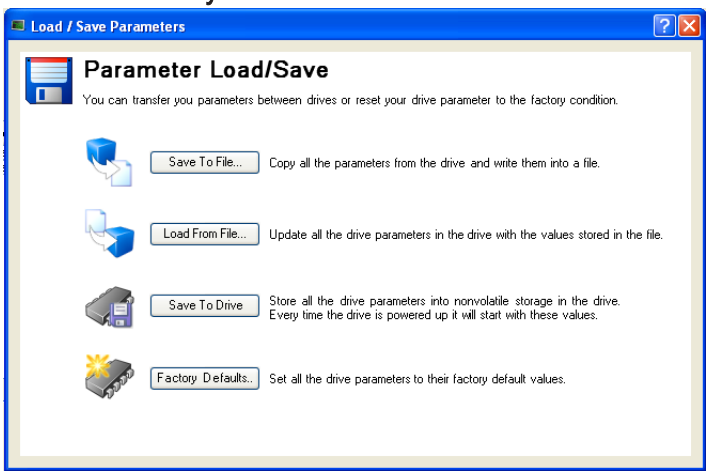
Why does Online Configuration Mode not work after I reload the drive's factory default parameters?

14.1.7.1 Description

This issue occurs with these actions:

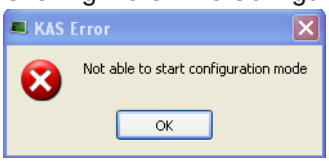
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. Connect to the controller and download your application.
2. In the Project Explorer, open the EtherCAT properties.
3. Click the"Online Configuration Mode" (➔ p. 471).
4. In the Project Explorer, right-click the AKD_1 and select **Load/Save Parameter...**.
5. Select the **Factory Defaults...** command to reset the drive to its default parameters.



NOTE

Clicking the Online Configuration Mode leads to this error:



Reason

If you set the drive to its default parameters, then all the AKD parameters are restored and the unique ID (FBUS.PARAM03) used to identify the drive is lost.

Solution

You have to perform a new scan operation after setting the parameters to its default values.

TIP

You can also clear the **Write a unique ID** option in the XML configuration tab.
See "ENI File tab" (➔ p. 541).

How can I fix security issues?

If you encounter any security issues during execution of Kollmorgen Automation Suite, contact your IT department to set your proxy properly.

14.1.7.2 Firewall

You may have to define your firewall settings to allow accessing the IP addresses used by KAS (e.g., IP address of the target system or localhost IP address for the KAS Runtime Simulator: 127.0.0.1).

14.1.7.3 Port Numbers

Port numbers have to be set properly in your firewall settings to avoid any trouble during communication.

Example: When downloading the application to the target or plugging a probe to the Softscope.

Kollmorgen strongly recommends opening port numbers over 1024, as well as the range 502 to 520.

14.2 Compiler Errors

These errors may be seen when compiling a project:

- "Compile Optimized Code FAILED" (→ p. 796)
- "Compiling Optimized Code Fails due to User Permissions" (→ p. 796)
- "EtherCAT: ERROR: In AKD_1, PDO object index 0x6040, subIndex 0 is redundant." (→ p. 798)
  - "Failed to export ENI file, check the logs for more details." (→ p. 798)
  - "EtherCAT: Warning : In AKD_1, PDO object index 0x3470, subIndex 4 is redundant." (→ p. 798)
- "EtherCAT: ERROR: Failed to retrieve Vendor..." (→ p. 797)
  - "EtherCAT: ERROR: Failed to retrieve EtherCAT device... " (→ p. 797)
- "Unexpected complex data in expression." (→ p. 798)
  - "Boolean expression expected after IF." (→ p. 798)

14.2.1 Compile Optimized Code FAILED

These compiler outputs indicate a failure to compile the optimized `C code`:

```
Controller:PLC:> ------------- Compile Optimized Code -------------
Controller:PLC:> Compiling "C" code.
Controller:PLC:> ERROR: Failed to compile PLC "C" code.
Controller:PLC:> Compilation log:
...
Controller:PLC:> End of Compilation log
Controller:PLC:> Compile Time Elapsed:1.37 seconds
Controller:PLC:> ------------- Compile Optimized Code FAILED -------------
```

Error Correction

Locate and send these items to Kollmorgen technical support:

- The compilation log (copied from the compiler output).
- All the files in this directory:
  `%LOCALAPPDATA%\Kollmorgen\KAS\IDE_Instances\<IDE_Instance_ID>\Project\Controller\PLC\CC\`

To help us resolve the error with you, call or e-mail the appropriate support for your region.

Kollmorgen technical support contact information can be found here:

- Global Support Contacts
- Kollmorgen Support and Service

See Also

- "Compiling Optimized Code Fails due to User Permissions" (→ p. 796)
- "EtherCAT: ERROR: In AKD_1, PDO object index 0x6040, subIndex 0 is redundant." (→ p. 798)
  - "Failed to export ENI file, check the logs for more details." (→ p. 798)
  - "EtherCAT: Warning : In AKD_1, PDO object index 0x3470, subIndex 4 is redundant." (→ p. 798)
- "EtherCAT: ERROR: Failed to retrieve Vendor..." (→ p. 797)
  - "EtherCAT: ERROR: Failed to retrieve EtherCAT device... " (→ p. 797)
- "Unexpected complex data in expression." (→ p. 798)
  - "Boolean expression expected after IF." (→ p. 798)

14.2.2 Compiling Optimized Code Fails due to User Permissions

A possible cause for optimized PLC code to fail compilation is if the user does not have permission to write to an output file or folder.

Example 1: The file cclog.txt is not writeable.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
Controller:PLC:> ------------ Compile Optimized Code ------------
Controller:PLC:> ERROR: cannot remove old output file
"C:\Users\my.name\AppData\Local\Kollmorgen\KAS\IDE_
Instances\40228\Project\Controller\PLC\CC\cclog.txt"
Controller:PLC: > To resolve the issue, set the permissions on the file so
that it is writeable.
Controller:PLC:> Compile Time Elapsed:0.02 seconds
Controller:PLC:> ------------ Compile Optimized Code FAILED ------------
```

Example 2: The file main.c is not writeable.

```
Controller:------------ Compile PLC --------------
...
Controller:PLC:main
Controller:PLC:c:\users\my.name\appdata\local\kollmorgen\kas\IDE_
Instances\40228\project\controller\plc\cc\main.c:
Cant write output file
Controller:PLC:appli
Controller:PLC:Error(s) detected
Controller:------------ PLC failed ---------------
```

Error Correction

Change the file permissions to allow writing to the file.

See Also

- "Compile Optimized Code FAILED" (→ p. 796)
- "EtherCAT: ERROR: In AKD_1, PDO object index 0x6040, subIndex 0 is redundant." (→ p. 798)
  - "Failed to export ENI file, check the logs for more details." (→ p. 798)
  - "EtherCAT: Warning : In AKD_1, PDO object index 0x3470, subIndex 4 is redundant." (→ p. 798)
- "EtherCAT: ERROR: Failed to retrieve Vendor..." (→ p. 797)
  - "EtherCAT: ERROR: Failed to retrieve EtherCAT device... " (→ p. 797)
- "Unexpected complex data in expression." (→ p. 798)
  - "Boolean expression expected after IF." (→ p. 798)

14.2.3 EtherCAT: ERROR: Failed to retrieve Vendor...

or

14.2.3.1 EtherCAT: ERROR: Failed to retrieve EtherCAT device...

These errors indicate either:

- The ESI file for the EtherCAT device is missing.
- Information is missing from the ESI file for a device and/or module.

Examples

```
EtherCAT: ------------ Generating EtherCAT Network Information (ENI) file --
------------
EtherCAT: ERROR: Failed to retrieve Vendor for vendorID=0x21 in ESI cache
Project compile failed
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
EtherCAT: ------------- Generating EtherCAT Network Information (ENI) file --
-------------
EtherCAT: ERROR: Failed to retrieve EtherCAT device (vendorID=0x2,
productCode=0x44c2c52, revision=0x120000) in ESI cache
Project compile failed
```

Error Correction

1. Identify which EtherCAT devices or modules are listed as **Unknown** in the Project View.
2. Import an appropriate ESI file.

See Also

- "Unknown - Missing ESI File" (➔ p. 167).
- "Compile Optimized Code FAILED" (➔ p. 796)
- "Compiling Optimized Code Fails due to User Permissions" (➔ p. 796)
- "EtherCAT: ERROR: In AKD_1, PDO object index 0x6040, subIndex 0 is redundant." (➔ p. 798)
  - "Failed to export ENI file, check the logs for more details." (➔ p. 798)
  - "EtherCAT: Warning : In AKD_1, PDO object index 0x3470, subIndex 4 is redundant." (➔ p. 798)
- "Unexpected complex data in expression." (➔ p. 798)
  - "Boolean expression expected after IF." (➔ p. 798)

14.2.4 EtherCAT: ERROR: In AKD_1, PDO object index 0x6040, subIndex 0 is redundant.

or

14.2.4.1 Failed to export ENI file, check the logs for more details.

or

14.2.4.2 EtherCAT: Warning : In AKD_1, PDO object index 0x3470, subIndex 4 is redundant.

These errors indicate redundant PDO entries.

Error Correction

See "Redundant PDO Entries" (➔ p. 567).

See Also

- "Compile Optimized Code FAILED" (➔ p. 796)
- "Compiling Optimized Code Fails due to User Permissions" (➔ p. 796)
- "EtherCAT: ERROR: Failed to retrieve Vendor..." (➔ p. 797)
  - "EtherCAT: ERROR: Failed to retrieve EtherCAT device... " (➔ p. 797)
- "Unexpected complex data in expression." (➔ p. 798)
  - "Boolean expression expected after IF." (➔ p. 798)

14.2.5 Unexpected complex data in expression.

or

14.2.5.1 Boolean expression expected after IF.

These errors indicate that the IF THEN ELSE ELSIF END_IF condition used in the application code does not evaluate to a BOOL expression.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
Controller:------------- Compile PLC ---------------
…
Controller:PLC:MCFB_Jog
Controller:PLC:MCFB_Jog: (16): Unexpected complex data in expression
Controller:PLC:MCFB_Jog: (16): Boolean expression expected after IF
Controller:PLC:MCFB_Jog: (19): END_IF: New statement expected
Controller:PLC:Error(s) detected
Controller:------------- PLC failed ---------------
Controller:------------- Device compile failed ---------------
...
Project compile failed
```

This compile error can happen when a non-compliant syntax is used for verifying the function block outputs.

- In KAS v4.0 and earlier releases, it was possible to use the function block return statement in the same way as function return values.
  - This could result in undefined behavior.
- In KAS v4.0 and later releases, the compiler generates a an **Unexpected complex data in expression** error for non-compliant syntax.

```
IEC 61131-3 Non-Compliant Syntax:
If inst_functionBlock(myVar1) Then
   // ...
End_If;
```

```
IEC 61131-3 Compliant Syntax:
inst_functionBlock(myVar1);
If inst_functionBlock.Q Then
   // ...
End_If;
```

Error Correction

1. Identify the code that caused compile error.
2. If the error is due to non-compliant syntax, replace this code using compliant syntax for the corresponding function block.

Example: Non-compliant Syntax

```
   IF Inst_r_trig(Inst_TON.Q) THEN
     Inst_DriveParamRead( TRUE, any_to_int(Inst_MC_ReadParam.Value),
'DRV.FAULT1' );
   END_IF;
```

Example: Compliant Syntax

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
   Inst_r_trig(Inst_TON.Q);
   IF Inst_r_trig.Q THEN
     Inst_DriveParamRead( TRUE, any_to_int(Inst_MC_ReadParam.Value),
'DRV.FAULT1' );
   END_IF;
```

NOTE

If you prefer to not update a KAS project to use compliant syntax, select the **Allow non-compliant syntax to evaluate function block outputs** check box in the "Project Settings dialog box" (→ p. 461).

⚠WARNING

UDFBs and libraries can have improper syntax.

These Kollmorgen UDFBs have been corrected and can be re-imported into the application code.

See Re-import Kollmorgen UDFBs.

- FB_AKDFltRpt
- FB_AxisPlsPosModulo
- FB_AxisPlsPosNoModulo
- FB_S700FltRpt
- MCFB_AKDFault
- MCFB_DriveFault
- MCFB_Jog
- MCFB_StepAbsSwitchFastInput
- MCFB_StepLimitSwitchFastInput
- MLFB_DriveFault
- MLFB_PlsPosFw
- MLFB_PlsPosFwBw
- MLFB_PlsTimeFw

See Also

- "Compile Optimized Code FAILED" (→ p. 796)
- "Compiling Optimized Code Fails due to User Permissions" (→ p. 796)
- "EtherCAT: ERROR: In AKD_1, PDO object index 0x6040, subIndex 0 is redundant." (→ p. 798)
  - "Failed to export ENI file, check the logs for more details." (→ p. 798)
  - "EtherCAT: Warning : In AKD_1, PDO object index 0x3470, subIndex 4 is redundant." (→ p. 798)
- "EtherCAT: ERROR: Failed to retrieve Vendor..." (→ p. 797)
  - "EtherCAT: ERROR: Failed to retrieve EtherCAT device... " (→ p. 797)

14.3 Compiler Warnings

These messages explain the cause of the warning messages from the compiler.

- "10001 - Macro Redefinition" (→ p. 801)
- "10002 - Not IEC Compliant" (→ p. 801)
- "10003 - Same Name to a Global Variable and a Local Variable" (→ p. 801)
- "10014 - Empty Line Ignored in Signal Resource (No Time Defined)" (→ p. 801)
- "10015 - On Line Change may not be Possible: Too Many POUs or Variables" (→ p. 801)
- "10019 - Definition of the Block was Changed" (→ p. 801)
- "10020 - Backward Jump may Block the Target Cycle" (→ p. 801)
- "10021 - FB Instance Missing (Unsafe)" (→ p. 801)
- "10022 - Output Used Before it is Set. Execution may be Unsafe" (→ p. 802)
- "10023 - Loop Instructions may Block the Target Cycle" (→ p. 802)
- "10024 - Implicit Loops may be Unsafe" (→ p. 802)
- "10025 - Input Parameters are not Considered as IN/OUT" (→ p. 802)
- "10026 - Backward Jumps may Block the Cycle" (→ p. 802)
- "10030 - Invalid PRINTF Format" (→ p. 802)
- "10033 - LREAL# Prefix is Required to Avoid Loss of Accuracy" (→ p. 803)
- "10034 - Loop Instructions may Block the Target Cycle" (→ p. 803)
- "10037 - The Same Modbus Address is Used for Several Variables" (→ p. 803)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

14.3.1 10001 - Macro Redefinition

This warning occurs when the same identifier is used for several definitions.

Example:

- #define PI 3.14
- #define PI 3.1416

This can also happen if the same identifier is used for a global definition and definition local to a program.

- If an identifier is reused, the compiler uses the last definition.
- If the identifier is defined both in global and local definitions, the local one is used.

14.3.2 10002 - Not IEC Compliant

This warning occurs when the use a semantic or block not compliant with the IEC 61131-3 standard.

This can occur in case of a specific block (e.g., **blink**) or some C-like operators (e.g., **Addition + or EQ =**).

14.3.3 10003 - Same Name to a Global Variable and a Local Variable

This warning occurs when the same name is used for a global variable and a variable local to a program.

- If the variable name is used in the body of the program, it represents the local variable.
- It is strongly **not** recommended to reuse names because this creates unclear programming and may lead to confusion and errors.

14.3.4 10014 - Empty Line Ignored in Signal Resource (No Time Defined)

This warning occurs when an empty line is inserted in the definition of a signal resource.

The compiler considers a line as empty if the line has no time value defined.

14.3.5 10015 - On Line Change may not be Possible: Too Many POUs or Variables

If Online Change is enabled, this warning occurs when more than 10 POUs in an application.

- Ignore this warning.
  - This warning is for a future feature, when the number of additional POUs during Online Change is configurable.

14.3.6 10019 - Definition of the Block was Changed

This warning occurs when the set of arguments of a block used in a FBD diagram has changed and diagram was not updated with the new block interface.

Re-select the block and check its input and output parameters in the program referred to by the message.

14.3.7 10020 - Backward Jump may Block the Target Cycle

This warning highlights the use in an FBD diagram of:

- A backward **Jumps JMP JMPC JMPNC JMPCN** instruction.
- A loop instruction (**REPEAT UNTIL END_REPEAT** or **WHILE DO END_WHILE**) put on a network break.

When using these programming semantics, **be very careful** the loop is finite and it brings no risk to hang the PLC execution cycle.

14.3.8 10021 - FB Instance Missing (Unsafe)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This warning occurs when a function block is used in an FBD diagram without any instance specified.

- In some conditions, the compiler may authorize this and an instance is automatically created.
  - This is not safe.
  - This can lead to unpredictable errors if you operate an On Line Change.
- Always specify a valid declared instance for a function block call.

14.3.9 10022 - Output Used Before it is Set. Execution may be Unsafe

This warning occurs when a sub-program uses the value of a local variable before setting its value.

This can lead to unpredictable errors if you operate an On Line Change.

14.3.10 10023 - Loop Instructions may Block the Target Cycle

This warning highlights the use in an FBD diagram of:

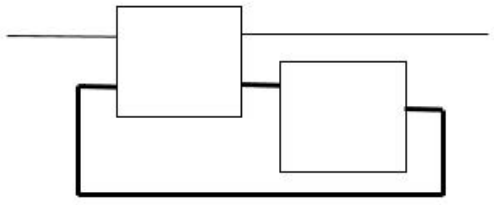
- A backward **Jumps JMP JMPC JMPNC JMPCN** instruction.
- A loop instruction (**REPEAT UNTIL END_REPEAT** or **WHILE DO END_WHILE**) put on a network break.

When using these programming semantics, **be very careful** the loop is finite and it brings no risk to hang the PLC execution cycle.

14.3.11 10024 - Implicit Loops may be Unsafe

This warning indicates an implicit loop in a FBD diagram.

- An implicit loop is a set of links in between functions without any variable specified in the entire flow. Example:



- This can lead to unpredictable errors if you operate an On Line Change.
- To avoid any problem, insert a variable on this link.

14.3.12 10025 - Input Parameters are not Considered as IN/OUT

This warning indicates that a sub-program assigns the value of one of its INPUT parameters.

- This can be correct in some use cases.
- Avoid any confusion by copying the input parameter to a temporary variable and use it in the sub-program.

14.3.13 10026 - Backward Jumps may Block the Cycle

This warning highlights the use in an FFLD of a backward **Jumps JMP JMPC JMPNC JMPCN** instruction (i.e., a jump to a rung defined before).

When using these programming semantics, **be very careful** the loop is finite and it brings no risk to hang the PLC execution cycle.

14.3.14 10030 - Invalid PRINTF Format

This warning indicates a program uses a call to the **printf** function with a potentially unsupported “%” format.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Only “%ld”, “%lu”, “%lx” formats are supported by all systems.

14.3.15 10033 - LREAL# Prefix is Required to Avoid Loss of Accuracy

By default, REAL and LREAL constants are interpreted as REAL (single precision).

- If this constant is used in an LREAL expression or statement, it is converted to LREAL (double precision) and this will lose its accuracy.
  - To avoid this, always specify the LREAL# prefix for LREAL expressions.
- Floating-point numbers are store in memory in binary radix.
  - Even a simple decimal value(e.g., 1.3) is not finite when coded in binary.

14.3.16 10034 - Loop Instructions may Block the Target Cycle

This warning highlights the use in an ST Language program of a loop instruction:

- **FOR TO BY END_FOR**
  - Even a FOR loop with bounds may become infinite if the index variable is forced within the body of the loop.
- **REPEAT UNTIL END_REPEAT**
- **WHILE DO END_WHILE**
- When using these programming semantics, **be very careful** the loop is finite and it brings no risk to hang the PLC execution cycle.

14.3.17 10037 - The Same Modbus Address is Used for Several Variables

This warning indicates a MODBUS configuration (master or slave) contains two or more variables mapped to the same MODBUS address.

- This can make sense in some use cases (e.g., map two variables on the same input) and can really highlight a configuration error.
  - Example: Two variables mapped on the same output bit or register.

14.4 Connect Remotely

You can connect to a controller from an external network using VPN or other tunneling protocol.

You must open these ports which are used by KAS to connect to a controller:

Port	Component / Protocol Using the Port
PCMM2G: 80, 443	KAS-IDE: • Controller Web Server • Download PLC application • Runtime front end communication. • This is necessary for the connection to the runtime. • If port 80 is used by any other application, the Simulator uses an alternate port available in the host computer for HTTP communication.
502	HMI and other Modbus devices to communicate with Modbus TCP.
2222	EtherNet/IP communication.
4002	KAS-IDE: Oscilloscope variable monitoring.
4003	KAS-IDE: • Communication with the PLC Virtual Machine. • This is necessary for the connection to the runtime.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Port	Component / Protocol Using the Port
PCMM2G: 4840 (user-configurable)	OPC UA Server Communications: • The OPC UA Server uses port 4840 by default. • Users can configure the controller to use a different port number if required. • See OPC UA - Insert an Endpoint for configuration details.
9900	KAS-IDE: • Communication with the runtime engine; WorkBench views. • This is necessary for the connection to the runtime.
34962	PROFINET communication.
34964	PROFINET communication.
44848	EtherNet/IP communication.

14.5 CPU Load Reduction Techniques

There are several things you can change to reduce the CPU load:

- Reduce the cyclic update rate by increasing the Cycle Time in the EtherCAT view. This will have the biggest impact to improve the RealTimeMargin.
- Reduce the number of axes. This will decrease the MotionExecTime.
- Distribute the PLC program execution across multiple PLC cycles (see "Define the PLC Cycle" on page 253). This will decrease the PLCProgExecTime.

14.6 EtherCAT Coupler Error Handling and Diagnosis

This section provides information about the diagnostic LEDs for the EtherCAT Coupler (AKT-ECT-000).

NOTE

This section is an excerpt of the KAS IO Module Installation Manual (AKT-ECT-000) EN.

14.6.1 EtherCAT Diagnostic LEDs

After switching on, the EtherCAT Bus Coupler immediately checks the connected configuration.

- Error-free start-up is indicated when the red I/O ERR LED goes out.
  - If the I/O ERR LED blinks, an error in the area of the terminals is indicated.
  - The error code can be determined from the frequency and number of blinks.
- The EtherCAT Bus Coupler has respectively a green and yellow LED at the RJ45 plug sockets.
  - This indicates the state of the fieldbus.
  - The RUN and ERROR LEDs (upper middle) indicate the state of the EtherCAT State Machine.
- On the upper right side of the Bus Couplers are two more green LEDs that indicate the supply voltage.
  - The left LED indicates the presence of the 24 VDC supply for the Bus Coupler.
  - The right LED indicates the presence of the supply to the power contacts.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

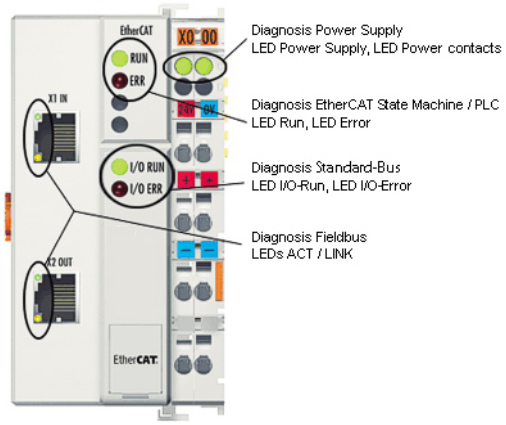


Figure 18-7: EtherCAT Coupler Diagnostic LEDs

14.6.2 EtherCAT LEDs - Fieldbus Connection Diagnosis

LEDs	Color	Display	Status	Description
LINK (X1 IN)	Yellow	Off		No connection with the previous EtherCAT client.
		On	Linked	Previous EtherCAT-client connected.
ACT (X1 IN)	Green	Blinking	Active	Communication with the previous EtherCAT client.
		Off		No connection with the previous EtherCAT client.
		On		No communication with the previous EtherCAT client.
LINK (X2 OUT)	Yellow	Off	Linked	Next EtherCAT client connected.
		On	Active	Next EtherCAT client connected.
ACT(X2 OUT)	Green	Blinking	Active	Communication with the next EtherCAT client.
		Off		No connection with the next EtherCAT client.
		On		No communication with next previous EtherCAT client.

14.6.3 EtherCAT LED - Off Power Supply Diagnosis

LEDs	Color	Display	Description
Left LED	Green	Off	Bus coupler has no power.
Right LED	Green	Off	No 24 VDC power connected to the power contacts.

14.6.4 EtherCAT LED - Power Supply Diagnosis

LED	Color	Display	Description
Power Supply	Green	Off	No operating voltage connected.
		On	24 VDC operating voltage connected.
Power Contacts	Green	Off	No 24 VDC power connected to the power contacts.
		On	24 VDC power connected to the power contacts.

14.6.5 EtherCAT LEDs - Standard Data Diagnosis

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

LEDs		Display	Status	Description
I/O-Run	Green	Off	–	EtherCAT Bus inactive.
I/O-Run	Green	On	–	EtherCAT Bus active.

14.6.5.1 I/O Errors

LED Red; I/O Error	Error Code Argument	Description	Remedy
Persistent, continuous blinking		EMC problems.	• Check power supply for overvoltage or under voltage peaks. • Implement EMC measures. • If an EtherCAT Bus error is present, it can be localized by a restart of the coupler (i.e., switch it off and on again).
1 Pulse	0	EEPROM checksum error.	Set manufacturer's setting with the configuration software.
	1	Code buffer overflow.	• Insert fewer Bus Terminals. • The programmed configuration has too many entries in the table Software update required for the Bus Coupler.
	2	Unknown data type.	Software update required for the Bus Coupler .
2 Pulses	0	Programmed configuration has an incorrect table entry.	Check programmed configuration for correctness.
	n (n > 0)	Table comparison (Bus Terminal n).	Incorrect table entry.
3 Pulses	0	EtherCAT Bus command error.	• No Bus Terminal inserted. • One of the Bus Terminals is defective. 1. Halve the number of Bus Terminals attached. 2. Check whether the error is still present with the remaining Bus Terminals. 3. Repeat until the defective Bus Terminal is located.
4 Pulses	0	EtherCAT Bus data error, break behind the Bus Coupler.	• Check whether the n+1 Bus Terminal is correctly connected. • Replace if necessary.
	n	Break behind Bus Terminal n.	Check whether the Bus End Terminal is connected.
5 Pulses	n	EtherCAT Bus error in register communication with Bus Terminal n.	Exchange the nth Bus Terminal.
14 Pulses	n	nth Bus Terminal has the wrong format.	• Start the Bus Coupler again. • If the error occurs again, exchange the Bus Terminal.
15 Pulses	n	Number of Bus Terminals is no longer correct.	• Start the Bus Coupler again. • If the error occurs again, restore the manufacturers setting using the configuration software.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

LED Red; I/O Error	Error Code Argument	Description	Remedy
16 Pulses	n	Length of the EtherCAT Bus data is no longer correct.	• Start the Bus Coupler again. • If the error occurs again, restore the manufacturers setting using the configuration software.

14.6.6 EtherCAT LEDs - State Machine / PLC Diagnosis

LEDs	Color	Display	Status	Description
Run	Green	Off	Init	State of the EtherCAT State Machine: INIT = Initialization.
		Blinking	Pre-operational.	State of the EtherCAT State Machine: PREOP = Pre-Operational.
		Single Flash	Safe operational.	State of the EtherCAT State Machine: SAFEOP = Safe-Operational.
		On	Operational	State of the EtherCAT State Machine: OP = Operational.
		Flashes	Bootstrap	State of the EtherCAT State Machine: BOOT = Bootstrap (Update of the coupler firmware).
Error	Red	Off	–	No errors.
		Blinking	Err-Operational No Communication.	PLC error / Lost frames.

14.7 EtherCAT Diagnostics and Errors

14.7.1 EtherCAT Diagnostics

The EtherCAT nodes provide several types of diagnostic information through various registers.

The KAS controller uses these diagnostic registers to detect error conditions in the EtherCAT network.

This can include:

Diagnostic Register	Description
Physical Link status	Indicates if a hardware connection between two nodes is established.
Link Lost Counter	• Indicates how many connection lost events have occurred between two nodes. • Some EtherCAT devices do not detect the lost link event.
Rx Error Counter	This counter is incremented when the node detects any signal error detected by the hardware.
Rx Invalid Frame Counter	Invalid frame error is detected when the Cyclic Redundancy Check (CRC) does not match the received frame CRC.
Forwarded Rx Error Counter	Frame error detected by the previous node and forwarded to this node.
EtherCAT Processing Unit Error Counter	• Counts errors in the EtherCAT frames such as frame length or a non-EtherCAT frame. • Not all EtherCAT devices support this register.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- **Application Started:** When the application is started, the KAS controller resets these items of all the nodes:
  - Link Lost Counter
  - Rx Error Counter
  - Rx Invalid Frame Counter
  - Forwarded Rx Error Counter
  - EtherCAT Processing Unit Error Counter registers.
- **Application Running:** While the application is running, the KAS controller monitors every cyclic frame and detects any missing or invalid frames.
  - If any frame is missed or invalid, it generates an "A38" (→ p. 584) alarm.
- **Application Stopped:**When the application is stopped, the controller reads the EtherCAT node diagnostic registers and process them.
  - If any error condition is detected, an EtherCAT diagnostic log message is generated.
  - The log messages can be used to identify the location of the communication failure.

The node register of **Link Lost Counter**, **Rx Error Counter**, **Rx Invalid Frame Counter**, **Forwarded Rx Error Counter**, and **EtherCAT Processing Unit Error Counter** have non-zero values when a cable disconnection or an EtherCAT frame corruption occurs.

These examples indicate a communication failure occurred between Node 0 and Node 1.

```
EtherCAT   | WARNING | EtherCAT diagnostics: AKD_1, port B (out) RX Error
Counter is 255
```

```
EtherCAT   | WARNING | EtherCAT diagnostics: AKD_2, port A (in) RX Error
Counter is 255
```

14.7.2 EtherCAT Error Messages

This section covers these error messages linked to the EtherCAT motion bus displayed in the "Information and Logs window" (→ p. 123):

```
Working Counter failure limit exceeded due to failed node response to cyclic
commands. Stopping EtherCAT network cyclic communication.
```

```
Link Error detected! Please, check controller connection.
```

```
Slave %s is not responding to acyclic frame. Please, check power supply or
connection.
```

14.7.2.1 Error Message Causes

These messages can appear because of these causes:

14.7.2.1.1 Wrong / Missing Device

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Case Description	Results
The XML network configuration file contains the list of all EtherCAT devices present in the network. At the EtherCAT initialization phase, the master checks that: • Every physical device in the network corresponds to the configured devices. • The master detects if the configuration does not match the physical devices. • The Standard I/O Couplers and I/O slices are correct by adding the proper commands in the network configuration file. • This allows the detection of wrong or missing Standard I/O Coupler.	• An Error log is generated with the relevant information. • The EtherCAT startup is aborted, as well as the startup of the machine.

14.7.2.1.2 Link Loss / Device Fault

Case Description	Results
• This kind of error can appear anytime in the EtherCAT communication, typically when a cable is disconnected, cut, or whenever an EtherCAT device is damaged. • The master has a mechanism that detects such situations.	• An Error log is generated with the relevant information. • All still-reachable axes must be stopped and powered off. • The EtherCAT communication is aborted. • If the network is cut, the drives on the side of the network disconnected from the master are moved into an error state (**F29**). • They are automatically stopped and powered off. **NOTE** It can be necessary to put the axes in a safe position before powering it off. This action is application dependent.

14.7.2.1.3 Frame Loss

Case Description	Results
• For security, all frames sent must be received in a given timeout period (at least before the next cycle is started). • The master detects this case by managing the appropriate timeout watchdogs.	• An Error log is generated with the relevant information. • The EtherCAT communication is aborted.

14.7.2.1.4 Working Counter

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Case Description	Results
• When EtherCAT master (KAS controller) sends out an EtherCAT data gram, it sets the Working Counter (WC) value to 0. • The nodes receiving the frame increment the WC at the hardware level according to a precise rule. • When the controller receives the frame, it compares the WC with the expected value to detect error conditions. Possible reasons for the working counter error: • The slave EtherCAT device does not exist or cannot be physically reached due to connector/cable failure. • Slave hardware failure. • Slave is still busy with a previous command and is not ready.	• The controller generates an A31 alarm and logs a warning message when the WC values does not match the expected value. • When the controller detects three such working counter failures within 1000 EtherCAT frames, it generates an "E30" (➔ p. 580) error and shuts down the EtherCAT communication.

14.7.2.2 Other Messages Linked to EtherCAT

See "Communication and Fieldbus" (➔ p. 79).

This message is displayed if an error or inconsistency is discovered during the parsing of the XML file when the application is started:

```
Unable to open EtherCAT config file <file-name>
<file-name>:<line>:<column>: <parsing error>
```

14.7.3 Example: EtherCAT Communication Diagnosis Steps

These are the steps to diagnose EtherCAT communication errors.

Examples are for how an application program can help a technician detect the issue and repair it.

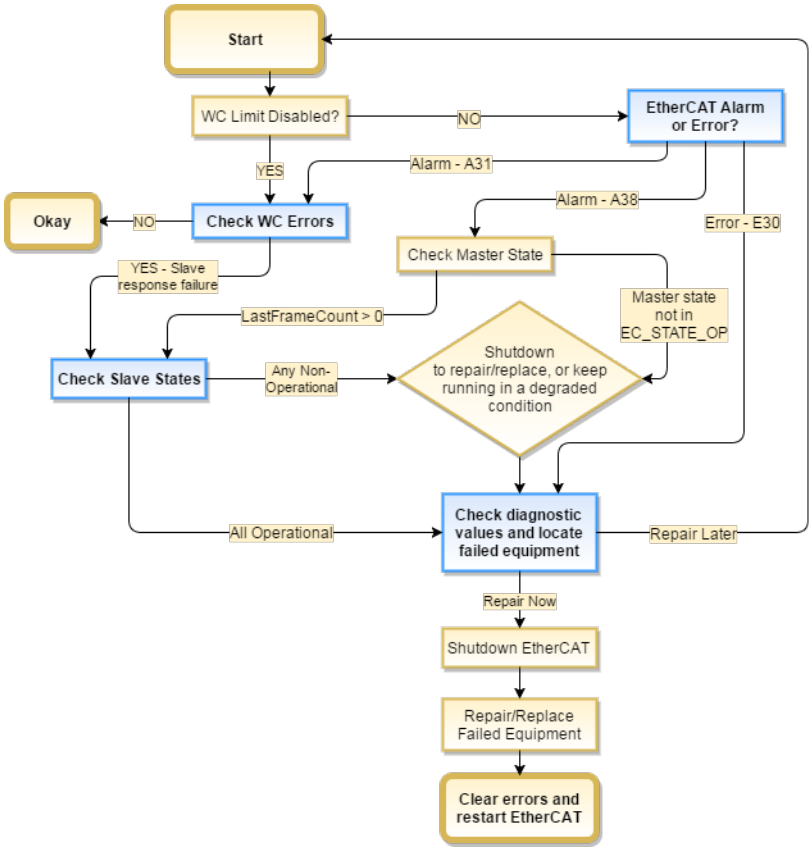
14.7.3.1 Diagnosing EtherCAT Communication Errors

This flow chart explains the steps involved in diagnosing the EtherCAT errors.

TIP

Click the blue fields in the flow chart to see sample code to use in a program to help with diagnosis.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



14.7.3.1.1 Code Examples for Diagnosing EtherCAT Communication Errors

Check for Working Counter Errors

This sample code checks for any Working Counter failure using the function ECATWCStatus.

If there is one or more working counter errors, the return value is greater than 0 (zero).

```
WC_ErrorCount := ECATWCStatus( 0 );
IF WC_ErrorCount > 0 THEN
   WC_error :=TRUE; // There is a communication problem.
   ELSE
   WC_error := FALSE; // No communication problem
END_IF;
```

Check for existing EtherCAT Alarms and Errors

The GetCtrlErrors function is used to get the Errors and Alarms in the controller that are related to EtherCAT.

```
ControllerErrorStatus:= GetCtrlErrors(ActiveError, ActiveAlarm) ;
A31Active := ActiveAlarm[31]; // True if A31 is active.
   A38Active := ActiveAlarm[38]; // True if A38 alarm is active
   E30Active := ActiveError[30] // True if E30 is active.
   // Process the alarms and Errors here.
```

Check the Device States

The ECATDeviceStatus function block is used to read the device state of the application.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
Inst_ECATDeviceStatus(True, slaveAddress);
IF Inst_ECATDeviceStatus.Done THEN
   IF Inst_ECATDeviceStatus.LinkStatus = EC_LINK_NO_COMMUNICATION THEN
      device_not_reachable := TRUE; // Device is not reachable. Link to the
device is broken.
   END_IF
   IF Inst_ECATDeviceStatus.State <> EC_STATE_OP THEN
      device_not_in_OPMODE := TRUE; // Slave is not in Operational mode.
   END_IF
END_IF;
```

Check the Connections for Errors

This sample code uses the ECATCommErrors function block to identify the connections that have errors.

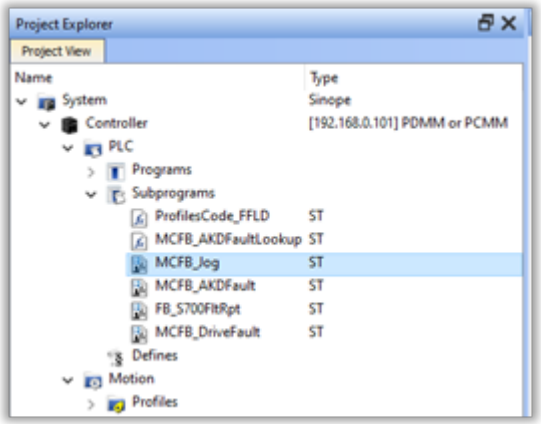
```
Inst_ECATCommErrors( TRUE, connections);
IF Inst_ECATCommErrors.Done THEN
   IF  Inst_ECATCommErrors.ConnectionCount > 0 THEN
   // Process the connection errors here.
   END_IF;
END_IF
```

TIP

See the article Building KAS Applications with Built-in EtherCAT Diagnostics on KDN for a code sample and project file that uses EtherCAT diagnostics.

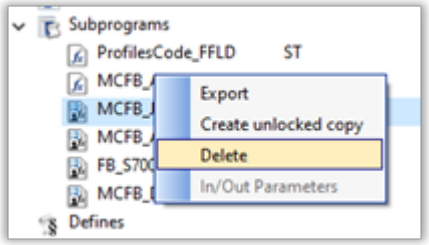
14.8 Re-import Kollmorgen UDFBs

1. In the Project Explorer, expand **Subprograms**.

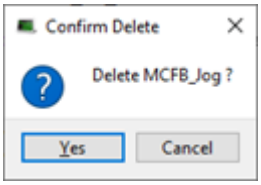


2. Right-click the Kollmorgen UDFB with the improper syntax and click **Delete**.

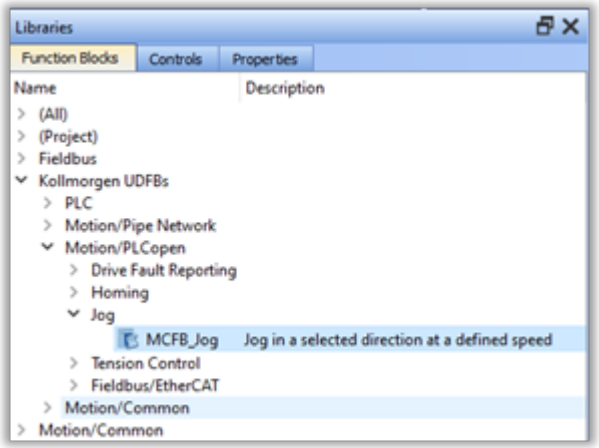
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



A confirmation message appears.



3. Click **Yes** to confirm deletion.
4. Click the **Libraries** tab.
5. Locate the UDFB that was just deleted.



6. Drag and Drop the selected UDFB to any program in the PLC code.
   NOTE
   This step copies the latest UDFB subprogram from the Kollmorgen library to the PLC code.
   Press the **Escape** key or click the **X** in the popup to avoid creating an instance of the function block in the program and delete the new code that was added here.
7. Compile the program and verify that the compilation error is resolved.

14.9 Troubleshoot FSoE Safety Parameters

Safety parameters can be set by using the FSoEParamsInit function block.

This operation requires proper configuration of the FSoE master, FSoE safety devices, and the EtherCAT master (PCMM2G).

Are there any errors?

Check the ErrorID output of FSoEParamsInit.

There are some common failure cases identified from the ErrorID values and the controller log messages.

Failure Case	Description
ECERR_DEVICE_NOTREADY	The Safety parameters can only be sent after the EtherCAT network state is in op-mode. Verify EtherCAT is ready: • Is the motion engine is running. • Use MLMotionStatus to verify the status is MLSTATUS_RUNNING before calling FSoEParamsInit. • Check the EtherCAT state directly. • Use ECATMasterStatus to verify the state is EC_STATE_OP.
ECERR_DEVICE_FSOE_MASTER_INVALID	This error is generated if the argument specified for FSoEParamsInit's FSoEMasterAddress is not an FSoE master. • Check to ensure that the EtherCAT address specified for the FSoE master is correct. • Check the supported FSoE masters list.
ECERR_DEVICE_TIMEOUT	This can occur if the FSoE master is not responding or is not configured properly. • Check the FSoE master diagnostic LEDs, Error Codes or Messages, log messages, and any other diagnostic tools available from the FSoE master vendor. • Example: With the BBH SCU-1-EC FSoE master, use the SafePLC2 software.

FSoEParamsInit(...) is successful but the AKD2G Safety State does not reach 6 - Normal operation.

If FSoeParamsInit(...) is successful but the AKD2G Safety State does not reach 6 - Normal operation, then the CRC expected by the AKD2G SMM does not match the CRC sent by the FSoE master.

Procedure

1. Use WorkBench or the KAS-IDE's integrated WorkBench (Configuration tab for the AKD2G) to diagnose.
2. Click the **Safety** item in the status bar.
   - If FSoE is operating normally:
     - The Safety State is 6 - Normal Operation.
     - The FSoE State is 54 - Process Data. (Figure 18-8)

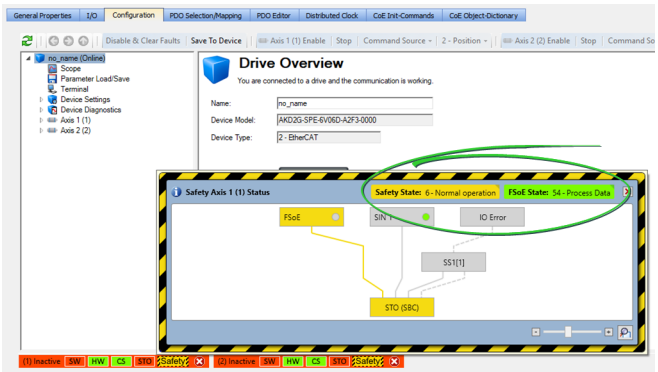


Figure 18-8: FSoE State: 54

   - If the CRC does not match the Safety State is 3 -Configuration required. (Figure 18-9)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

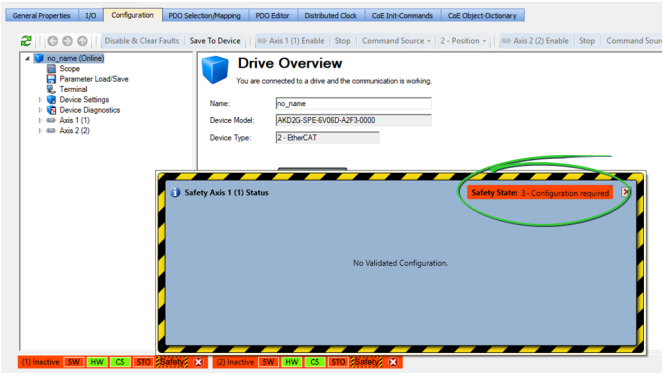


Figure 18-9: Safety State: 3

AKD2G module(s) are incorrectly set to use Workbench Parameters

The AKD2G has two options for configuring safety parameters.

These are determined by selecting the modules associated with the AKD2G in the KAS-IDE Project Tree.

- If the modules are set for **SRA Parameters Configured via Workbench**, the FSoE master cannot set the AKD2G's safety parameters.
- The **SRA Parameters** modules are set by default when an AKD2G safety device is discovered during EtherCAT scan.
  - If the **SRA Parameters Configured with Workbench** modules are selected, these modules must be changed to **SRA Parameters** to accept the parameter from the Safety Maser to use FSoEParamsInit.
  - Figure 18-10 shows the appropriate modules for using FSoEParamsInit with AKD2G.

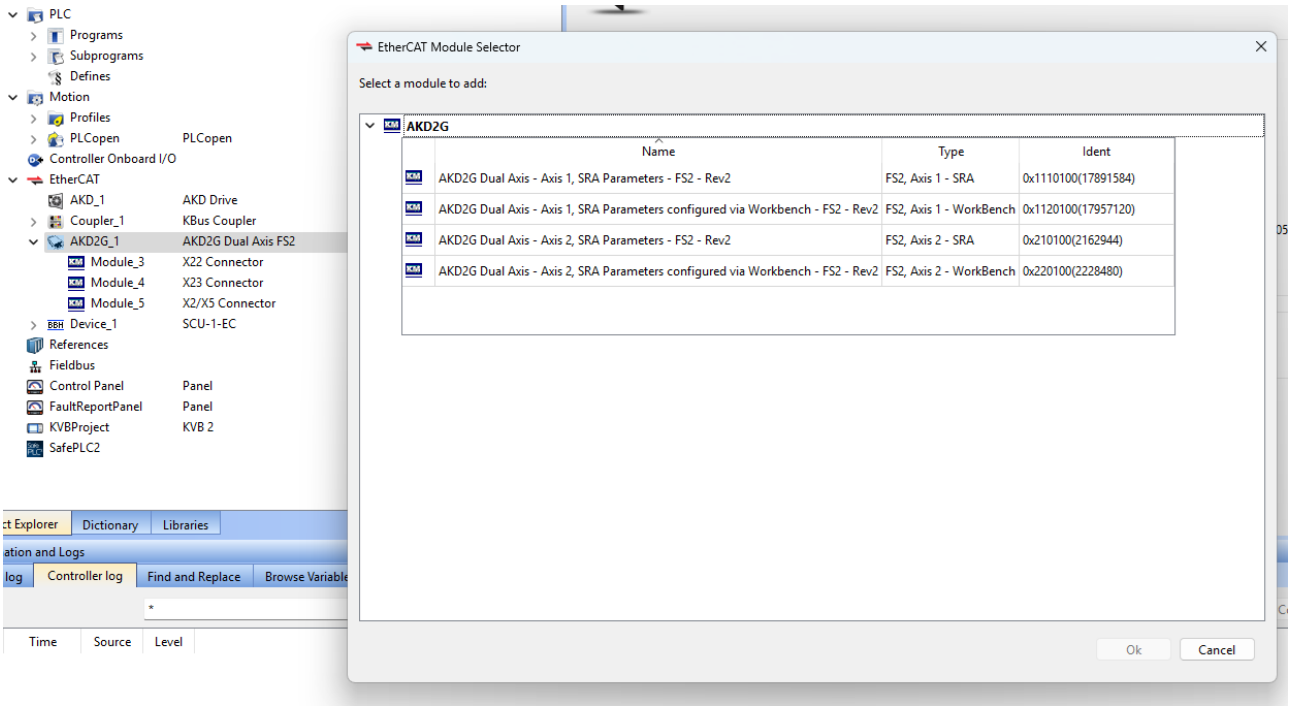


Figure 18-10: EtherCAT Module Select dialog box with AKD2G FSoE

The AKD2G profile in the SafePLC2 project is set to **SRA Parameters**.

This allows the "BBH FSoE Master" (➔ p. 767) to transmit safety parameters configured in the SafePLC2 project to the AKD2G via FSoE. (Figure 18-11)

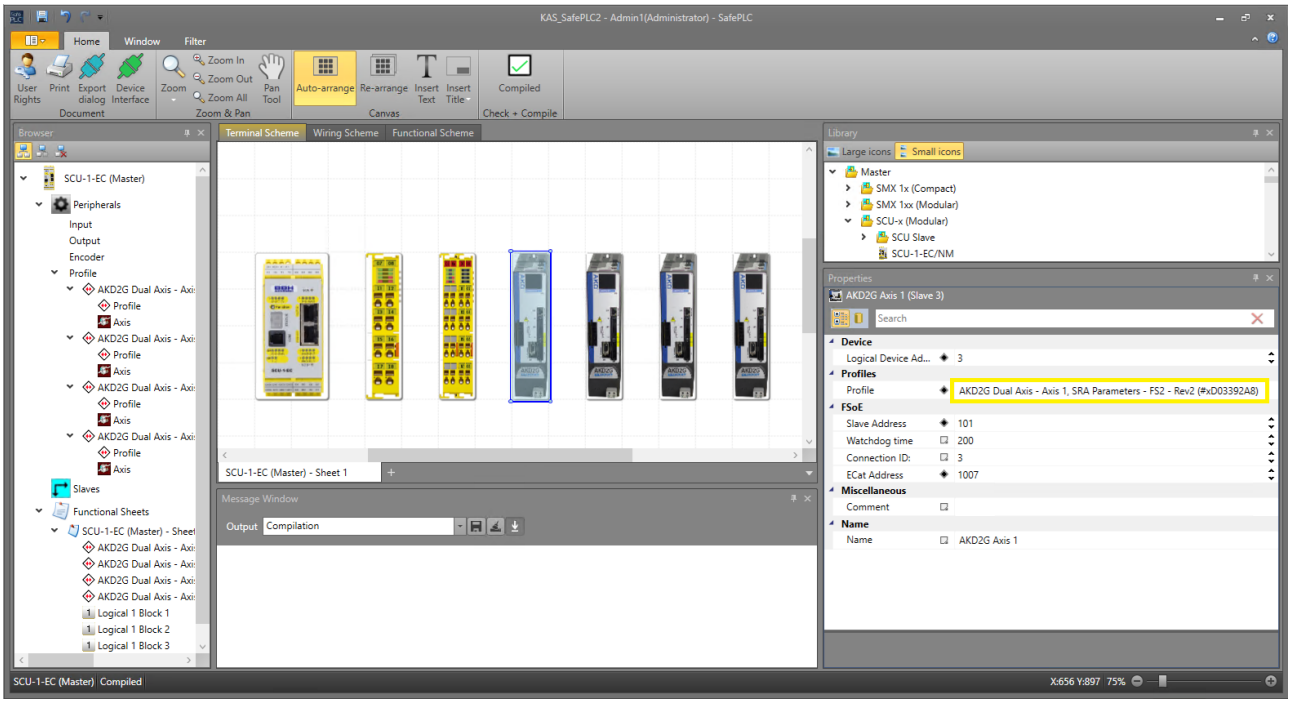


Figure 18-11: Example: KAS SafePLC2 window

FSoEParamsInit returns with Done = TRUE and Error = FALSE, but no parameters are initialized.

This can occur in two situations:

- If the FSoE safety parameters have been previously configured correctly.
- If the FSoE master has not been configured properly.

FSoEParamsInit only initializes safety parameters if the FSoE master determines there are parameters to set in the safety device(s).

Check the FSoE master configuration using the vendor's software.

Example: With the BBH SCU-1-EC FSoE master, use the SafePLC2 software.

See Also

- "Set Up FSoE Master and an AKD2G with SafeMotion Monitor" (➔ p. 891)
- "AKD2G Safety Parametrization Using FSoE with SCU-1-EC and PxMM / PCMM2G" (➔ p. 893)
- FSoEParamsInit

14.10 Troubleshoot KAS-IDE and SafePLC2

These are situations you may encounter when you are setting up a KAS project involving SafePLC2.

The ESI file for a Safety device is reported as missing.

EtherCAT devices missing ESI files are identified in the KAS-IDE's EtherCAT **Devices** tab. (Figure 18-12)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

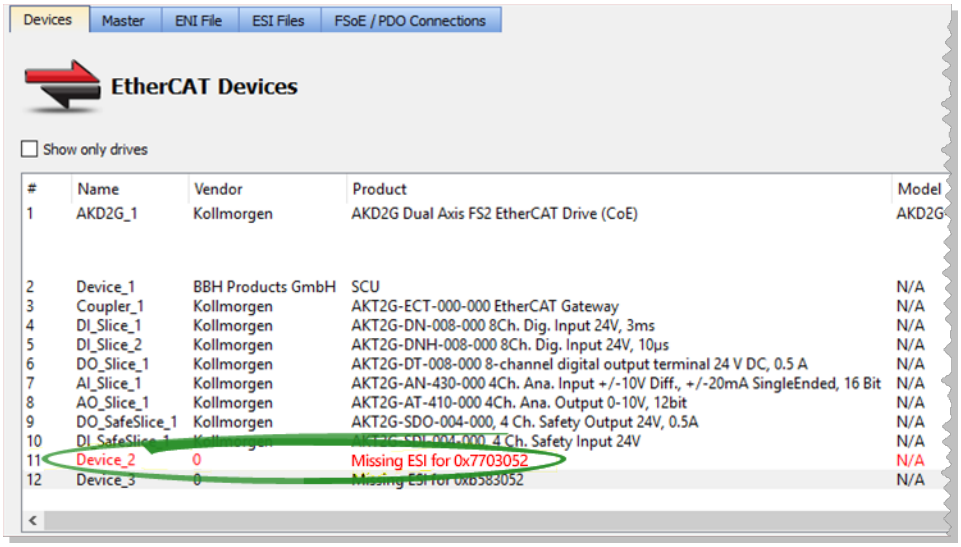


Figure 18-12: EtherCAT **Devices** tab

If a Safety device is added to a SafePLC2 project and the KAS-IDE's ESI library does not have a corresponding ESI file for the device, the Local log messages indicates a problem with device creation.

Example:

```
WARNING | Slave creation failed while trying to retrieve slave default
WARNING | Device_2: Slave creation failed while trying to retrieve
slave default Init Commands
WARNING | Device_2: Slave creation failed while trying to get
Distributed Clock(DC) details
WARNING | Slave creation failed while trying to retrieve slave default
PDOs
```

Avoid Error

- Add all Safety devices from the KAS-IDE, including their ESI files, and then launch SafePLC2 from the KAS-IDE project tree.
- SafePLC2 automatically reads the Safety device topology from the KAS-IDE.

Error Correction

- Use the KAS-IDE's EtherCAT-> **ESI Files** tab to add the ESI files for the missing devices.

Topology mismatch with the imported SafePLC2 project

If a SafePLC2 project file is imported into a KAS project, the KAS-IDE checks the incoming Safety project's EtherCAT topology.

Importing the project fails If there is a topology mismatch and the Local logs identifies the mismatched nodes. (Figure 18-13)

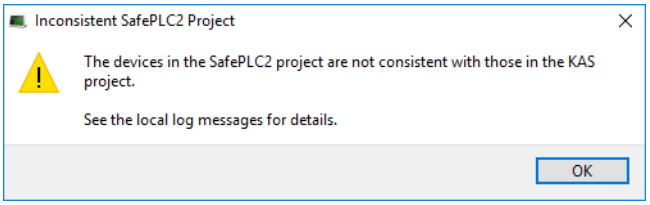


Figure 18-13: Inconsistent SafePLC2 Project error message

Example 1

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
ERROR | SafePLC2 topology parser: device #1 - SafeModule #1 -
ConnectsTo element - KAS ID 0 does not match that of the device with
SafePLC2 ID 1 - 4
ERROR | SafePLC2 topology parser: device #1 - SafeModule #2 -
ConnectsTo element - KAS ID 0 does not match that of the device with
SafePLC2 ID 1 - 4
ERROR | SafePLC2 topology parser: device #1 - SafeModule #3 -
ConnectsTo element - KAS ID 8 does not match that of the device with
SafePLC2 ID 3 - 9
ERROR | SafePLC2 topology parser: device #1 - SafeModule #4 -
ConnectsTo element - KAS ID 9 does not match that of the device with
SafePLC2 ID 4 - 8
ERROR | SafePLC2 topology parser: device #2 - the EtherCAT ID of the
device with KAS ID 4 does not match that of "DI_Slice_2"
ERROR | SafePLC2 topology parser: device #3 - SafePLC2 ID (3) does not
match that of "DI_SafeSlice_1" (4)
ERROR | SafePLC2 topology parser: device #3 - the EtherCAT ID of the
device with KAS ID 9 does not match that of "DI_SafeSlice_1"
ERROR | SafePLC2 topology parser: device #4 - SafePLC2 ID (4) does not
match that of "DO_SafeSlice_1" (3)
ERROR | SafePLC2 topology parser: device #4 - the EtherCAT ID of the
device with KAS ID 8 does not match that of "DO_SafeSlice_1"
```

Example 2

```
ERROR   | SafePLC2 topology parser: device #1 - no device with
identifier 11
ERROR   | SafePLC2 topology parser: device #2 - no device with
identifier 10
ERROR   | SafePLC2 topology parser: device #3 - no device with
identifier 18
ERROR   | SafePLC2 topology parser: device #4 - no device with
identifier 19
```

NOTE

Any pre-existing SafePLC2 project inside the KAS project is retained if importing the SafePLC2 project fails.

Error Correction

- Import the SafePLC2 project to the matching .kas project that was originally exported to SafePLC2.

Duplicate FSoE Address

During the KAS project compilation, the KAS-IDE verifies whether the FSoE addresses of all safety devices present in the KAS project are unique.

If duplicate FSoE addresses are detected, the KAS compiler generates a warning.

Example

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
EtherCAT: Warning: FSoE address 101 is duplicated for these devices:
AKD2G_1, DI_SafeSlice_1, DO_SafeSlice_1
```

Error Correction

1. Change the FSoE address of the devices and set them to be unique to the network.

   Refer to the device installation manual for setting the FSoE address.
2. Rescan the EtherCAT network and repeat the steps described in "SafePLC2 Projects" (➔ p. 219).

FSoE / PDO Connections do not match the EtherCAT topology.

The compiler generates warnings if the FSoE / PDO Connections defined by the Safety device topology from the SafePLC2 project do not match the EtherCAT Safety device topology defined by the KAS-IDE.

Example:

```
< compiler messages go here! >
```

Avoid Error

- Scan and discover all physical EtherCAT Safety devices from the KAS-IDE before launching SafePLC2 from the KAS-IDE Project tree.
- SafePLC2 automatically reads the Safety device topology from the KAS-IDE.

Error Correction

- Use the KAS-IDE's EtherCAT-> **Scan Devices** to discover and associate the physical Safety devices that are expected by the SafePLC2 project.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15 Advanced Topics

15.1 Coordinated Motion

This section provides an overview of Coordinated Motion, including general concepts to understand and use Coordinated Motion.

- "Group State Diagrams" (➔ p. 820)
- "Coordinate Systems" (➔ p. 820)
- Glossary and Acronyms proxy

See "Create a Linear or Circular Coordinated Motion Application" (➔ p. 824).

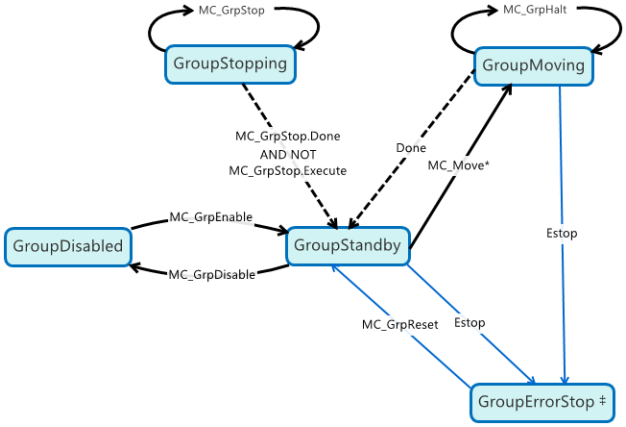
Coordinated motion in KAS-IDE is discussed in several locations and manners.

- "Design Coordinated Motion" (➔ p. 821) - Use this section to get started quickly with coordinated motion by walking you through the steps of setting up a project that uses coordinated motion.
- Motion / Common - Coordinated Motion - Use this reference section for function block parameters.

NOTE

The coordinated motion library supports coordinated motion for a maximum of 128 axes.
Hardware limitations may impose a lower limit for most applications.

15.1.1 Group State Diagrams



A limitation exists where a single axis ErrorStop condition does not change the AxesGroup state to GroupErrorStop.

15.1.2 Coordinate Systems

These are the different coordinate system (CS) types:

- Machine (MCS)
- Axes (ACS)
- Product/Program (PCS)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

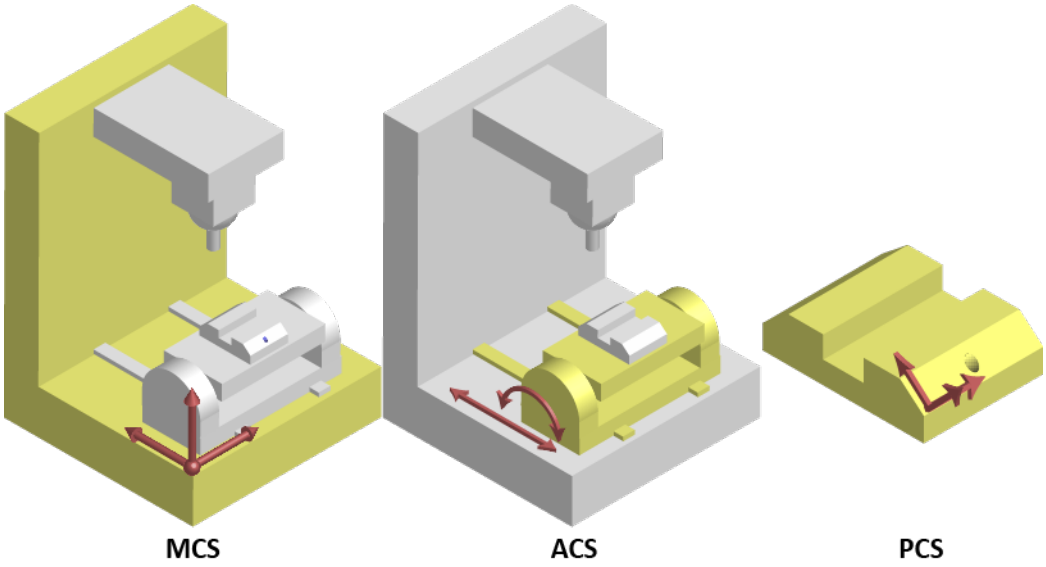
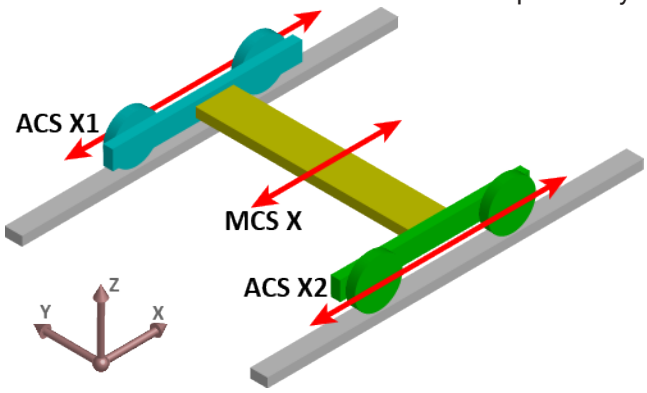


Figure 19-1: Examples of CS types on a machine and part.

- Many coordinated moves may be done in a choice of coordinate systems.
- The differences between the types are offsets and possibly transformational algorithms to convert between the different systems, which ultimately control the actual axes on a piece of machinery.

Example

- The X-axis of a Machine CS is meant to command a pair of Axes CS axes (X1 and X2) which together form a gantry.
- The relative movement of the MCS X axis would be added to both ACS axes.
- The two ACS axes can be commanded independently for minor alignment adjustments.



TIP

The Product Coordinate System is often rotated and/or offset from the Machine Coordinate System.

15.1.3 Design Coordinated Motion

These topics show how to create a coordinated motion application, including adding coordinated motion to existing applications.

- "Blending Between Moves" (→ p. 822)
- "Blending with Transitions" (→ p. 823)
- "Create a Linear or Circular Coordinated Motion Application" (→ p. 824)
- "Differences between MC_GrpHalt and MC_GrpStop" (→ p. 833)
- "Handling Axis Errors" (→ p. 838)
- "Transition Between Moves" (→ p. 840)

See Also

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Coordinated Motion" (➔ p. 820)
- Coordinated Motion Function Blocks

15.1.3.1 Blending Between Moves

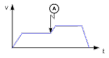
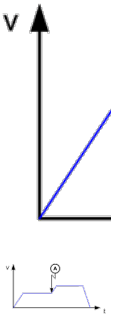
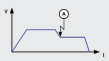
Some Coordinated Motion function blocks have a **BufferMode** input parameter.

Buffer modes include:

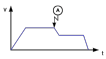
- BM_ABORTING = 0 = Aborting
- BM_BUFFERED = 1 = Buffered
- BM_BLENDING_PREVIOUS = 2 = Blending Previous
- BM_BLENDING_NEXT = 3 = Blending Next
- BM_BLENDING_LOW = 4 = Blending Low
- BM_BLENDING_HIGH = 5 = Blending High

When the current and next motion function blocks are blended (2 through 5), the axes group does not stop between motions.

- The velocity is blended according to the specified blending mode.
- The function block's **TransitionMode** parameter can be set to provide a smooth circular arc between moves.
- See "Transition Between Moves" (➔ p. 840).

Buffer Mode	Profile	Description
BM_ ABORTING		In the image, the second function block starts execution at Point A. • The current move stops execution immediately. • The next move becomes active and begins to accelerate or decelerate from the current velocity to its programmed velocity.
BM_ BLENDING_ HIGH		In the image, the second function block starts execution at Point A. The velocity is blended to the highest velocity of the two moves. • If the current move's velocity is higher than the next move's velocity, **Blending Previous** is used. • If the next move's velocity is higher than the current move's velocity, **Blending Next** is used.
BM_ BLENDING_ LOW		In the image, the second function block starts execution at Point A. The velocity is blended to the lowest velocity of the two moves. • If the current move's velocity is lower than the next move's velocity, **Blending Previous** is used. • If the next move's velocity is lower than the current move's velocity, **Blending Next** is used.
BM_ BLENDING_ NEXT		In the image, the second function block starts execution at Point A. The current move begins to accelerate or decelerate so that it reaches the next move's programmed velocity at the time the current move reaches its endpoint.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

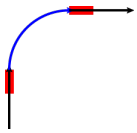
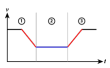
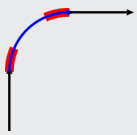
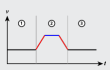
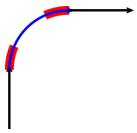
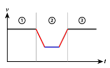
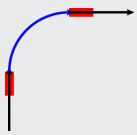
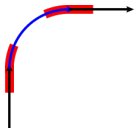
Buffer Mode	Profile	Description
BM_ BLENDING_ PREVIOUS		In the image, the second function block starts execution at Point A. • The current move stays at its programmed velocity to its endpoint. • When the current move reaches its endpoint, the next move becomes active and begins to accelerate or decelerate to its programmed velocity.
BM_ BUFFERED		In the image, the second function block starts execution at Point A. • The current move completes as was originally specified, coming to a stop. • When the current move reaches its endpoint, the next move becomes active and begins to accelerate to its programmed velocity.

15.1.3.2 Blending with Transitions

When blending and a transition are specified, the buffer mode is used to blend velocities of path segments when the path changes from the incoming segment to the transition segment and from the transition segment to the outgoing segment.

Notes

- S-Curve Corner transitions only support blending buffer modes.
  - All blending moves are treated the same when using S-Curve Corner transitions since the transition includes blending.
  - See "S-Curve Corner Line-to-Line Transitions" (➔ p. 849).
- The most common blending mode choices for applications are listed in the table.
  - This shows the velocity profile and where acceleration occurs on the path.
  - In the examples:
    - Blue = Transition arc.
    - Red = Where the acceleration occurs.
    - 1 = Incoming move.
    - 2 = Transition.
    - 3 = Outgoing move.

Buffer Mode	Example	Profile
BM_BLENDING_LOW When the transition velocity is smaller than the incoming and outgoing move velocities.		
BM_BLENDING_LOW When the transition velocity is larger than the incoming and outgoing move velocities.		
BM_BLENDING_HIGH When the transition velocity is smaller than the incoming and outgoing move velocities.		
BM_BLENDING_HIGH When the transition velocity is larger than the incoming and outgoing move velocities.		
BM_BUFFERED		

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.1.3.3 Create a Linear or Circular Coordinated Motion Application

A Coordinated Motion application can be created in one of these ways:

- Use a Coordinated Motion template to create a new application.
  - Two Coordinated Motion templates are currently available.
    - The first template controls two PLCopen axes in coordinated motion.
    - The second template controls two PLCopen axes in coordinated motion plus a third independent Pipe Network axis.
- Modify an existing application to included coordinated motion functions.
  - When modifying an existing application, axes need to be grouped to define the axes that will be active when performing coordinated motion on that group.
  - See "What are Axes Groups?" (➔ p. 827).

NOTE

Coordinated motion can only be performed with PLCopen axes.
Pipe Network axes do not support this feature.
Pipe Network axes can be moved independently from coordinated motion groups.
Any synchronization between coordinated motion and Pipe Network axes must be performed by the PLC application.

15.1.3.3.1.1 Groups

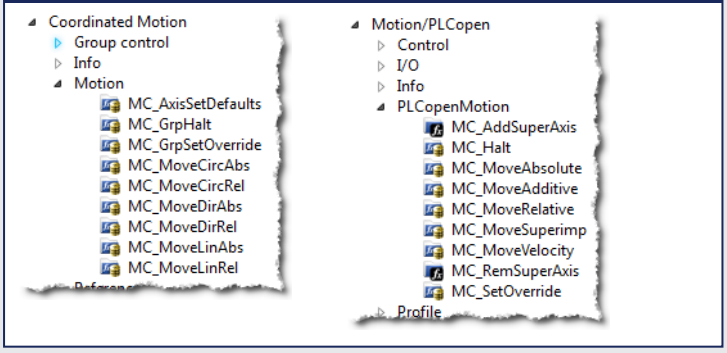
- Related axes are grouped in an axes group.
- Coordinated motion is then performed on an axes group.
- See "What are Axes Groups?" (➔ p. 827).

TIP

Remember these vital concepts when making interpolated motion.

- Interpolated motion requires creating a motion group that results in a second group coordinate system.
- Group coordinate system positions are only affected by group motion.
- Non-group coordinate system positions are only affected by non-group motion.

Group motion commands	Non-Group motion commands



NOTE

Axes that become part of a motion group are first homed using non -group function blocks to establish a home or starting position for the group motion.

15.1.3.3.2 Function Block Calls

These function blocks should be called before executing coordinated motion.

1. Call MLMotionInit (BasePeriod) to initialize the motion engine.
   Base period is specified in microseconds.

```
MLMotionInit(1000.0);   // 1000 µSec -> 1 mSec
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

2. Call MC_CreateAxesGrp (Enable, GroupName, UpdateRate, MaxNumberOfAxes, AxesGroupRef) to create a Coordinated Motion Axes Group.

   NOTE

   MC_CreateAxesGrp must be called between MLMotionInit() and MLMotionStart().

   ```
   Inst_MC_CreateAxesGrp(TRUE, 'GROUP1', 6, 2, Group1_ref);
   ```

   In this example, the axes group name is **GROUP1**, the update rate is 1 mSec (specified by '6') and the maximum number of axes that can be added to the group is 2.
   The group reference variable **Group1_ref** is used in future coordinated motion function block calls to reference this newly created group.

3. Call MC_InitAxesGrp (Enable, AxesGroup, VelLimit, AccLimit, DecLimit, JerkLimit) to initialize the path limits for velocity, acceleration, deceleration, and jerk.

   ```
   Inst_MC_InitAxesGrp(TRUE, Group1_ref, 100.0, 300.0, 300.0, 1000.0);
   ```

   In this example, the kinematic limits for axes group **Group1_ref** is set.
   The velocity limit is set to 100.0 user units/second.
   Acceleration and deceleration limits are set to 300.0 User units / sec^2.
   Jerk is set to 1000.0 User units / sec^3.

4. Call MC_CreatePLCAxis (AxisName, BusInterface, BusAddress, AxisNumber, AxisType, UserUnits, FeedbackUnits, Rollover, UpdateRate) to create a Coordinated Motion Axis.
   This function must be called for each Coordinated Motion Axis wanted in the application.

   NOTE

   MC_CreatePLCAxis needs to be called between MLMotionInit() and MLMotionStart().

   ```
   Inst_MC_CreateAxis(TRUE, 'CoordAxis1', 'EtherCATDriver', 1001, CoordAxis1_
   AxisNum, 0, 360, 1048576, 0, 6);
   Inst_MC_CreateAxis(TRUE, 'CoordAxis2', 'EtherCATDriver', 1002, CoordAxis2_
   AxisNum, 0, 360, 1048576, 0, 6);
   ```

   In this example:

   - Two axes are created and are named **CoordAxis1** and **CoordAxis2**.
   - The bus interface for both is **EtherCATDriver**.
   - The address of the drive on the bus is 1001 and 1002.
   - The axis numbers are set with variables CoordAxis1_AxisNum and CoordAxis2_AxisNum which is set to an integer value between 1 and 256.
     - Each axis number is unique.
   - The axis type for both, 0, indicates a servo axis.
   - The user units are 360, which is the user unit portion of the user unit/feedback ratio.
   - The feedback units are 1048576.
     - This is the feedback portion of the user unit/feedback ratio.
   - The rollover position for both, 0, indicates no rollover.
   - The update rate for both, 6, indicates a 1mSec update rate.

5. Call MLMotionStart () to start the Motion and the motion bus driver.
   This also initializes the EtherCAT network to operational mode.

   ```
   MLMotionStart();
   ```

6. Call MC_AddAxisToGrp (Execute, AxesGroup, Axis, IdentInGroup) for each axis to add to the group.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
Inst_MC_AddAxisToGrp(TRUE, Group1_ref, CoordAxis1_ref, 0);
Inst_MC_AddAxisToGrp(TRUE, Group1_ref, CoordAxis2_ref, 1);
```

In this example, we are adding two axes, CoordAxis1 and CoordAxis2, to the group referenced by **Group1_ref**.
The axes are stored in the IdentInGroup positions 0 and 1.
When the group was created, it was specified that no more than 2 axes will be part of this group.
Therefore, valid IdentInGroup locations are 0 and 1.

7. Call MC_Power (Enable, Axis, EnablePositive, EnableNegative, BufferMode) for each Coordinated Motion Axis to enable the drive and close the servo loop.

```
Inst_MC_Power1(TRUE, CoordAxis1_ref, TRUE, TRUE, 0);
Inst_MC_Power2(TRUE, CoordAxis2_ref, TRUE, TRUE, 0);
```

In this example, drives CoordAxis1_ref and CoordAxis2_ref are enabled and the position loop is closed.

8. Call MC_GrpEnable (Execute, AxesGroup) to change the state of the Coordinated Motion Axis Group from GroupDisabled to GroupStandby and allow motion to be performed on the group.

```
Inst_MC_GrpEnable(TRUE, Group1_ref);
```

In this example, **Group1_ref** state is changed from GroupDisabled to GroupStandby.
The group must be in GroupStandby to perform the motion.

9. In these examples, we want to set the current location of the axes in the group to 0, 0.
This is done by calling MC_GrpSetPos (Execute, AxesGroup, Position[], Relative, CoordSystem, BufferMode).

```
PosAbs[1]:= 0;
PosAbs[2]:= 0;
Inst_MC_GrpSetPos(TRUE, Group1_ref, PosAbs, 0, MC_COORDINATE_SYSTEM_ACS,
0);
```

In this example, the axis positions of **Group1_ref** is set to 0, 0.
PosAbs specifies the position for each axis in the group.
Relative input, 0, uses PosAbs to set the absolute position.
The coordinate system is set to ACS.

NOTE

No motion is performed when this function block is executed.

10. **Optional:** To Add more axes to the group, change the code this way:
    - In Step 2: Update the MaxNumberOfAxes input argument so that the group can handle the desired number of axes.
    - In Step 4: Create the additional axes to add to the group.
    - In Step 6: Add the additional axes to the group.
    - In Step 7: Power on the additional axes.
    - In Step 9: Increase the size of the PosAbs array so it matches the number you used in step 2. Set the position of the additional axes to zero.

After the above function calls have been made, we can start coordinated motion moves.

See Also

- "Perform a Linear Move" (→ p. 831)
- "Perform a Circular Move" (→ p. 827)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.1.3.3.3 What are Axes Groups?

Related axes are grouped in an Axes Group to support interpolation. Axes Groups are accessed using the type AXES_GROUP_REF.

This image shows the relationships between the different CSs and groups.

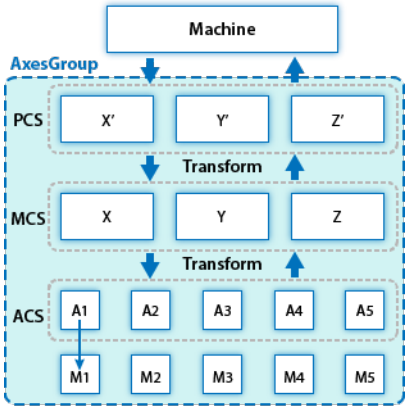


Figure 19-2: Overview of AxesGroup

The Axes Group, shown in blue, provides the interface to the user of the group of axes.

To access the relevant coordinate system, the relevant function blocks have an input Coordinate System which supports the three levels ACS, MCS, and PCS.

Parameters in the AxesGroupRef can include remaining time and remaining distance before target position (or velocity or equal) is reached.

15.1.3.3.4 Perform a Circular Move

Circular moves can be programmed using absolute or relative positions using the following function blocks:

- MC_MoveCircAbs which commands interpolated circular movement on an axes group to the specified absolute positions.
- MC_MoveCircRel which commands interpolated circular movement on an axes group to the specified relative positions.

Prior to performing any coordinated moves, some setup is needed.

- See "Create a Linear or Circular Coordinated Motion Application" (→ p. 824).
- Once these steps have been performed, a circular move can be performed.

In these examples, two circular moves are performed:

- "Absolute Circular Move" (→ p. 827)
- "Relative Circular Move" (→ p. 828)

15.1.3.3.5.1 Absolute Circular Move

In this move, an absolute circular move goes from (0, 0) to (90, 90).

- CircMode specifies that the aux point (0, 180) is crossed during the paths start to end.
- Call MC_MoveCircAbs.

```
PosCircAuxAbs[0] :=   0;   // A point on the circle that is crossed on
the
PosCircAuxAbs[1] := 180;   // path from start to end point.
PosCircEndAbs[0] :=  90;   // Absolute end point.
PosCircEndAbs[1] :=  90;
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
Inst_MC_MoveCircAbs(TRUE, Group1_ref, CIRC_MODE_BORDER, PosCircAuxAbs,
PosCircEndAbs, CIRC_PATH_CLKWISE, MaxVel, MaxAcc, MaxDec, 0, CS_ACS, BM_
BUFFERED, TM_NONE, TransParam);
```

In the example, a circular move is performed on axis group **Group1_ref**.

- CircMode is defined as CIRC_MODE_BORDER.
  - This mode indicates that the AuxPoint array input indicates a point on the circle which is crossed on the path from the starting point to the end point.
  - See "Circular Moves Diagrams" (➔ p. 829) about CircMode movement options.
- The AuxPoint array, **PosCircAuxAbs**, defines an absolute point on the circle which is crossed on the path from the starting point to the end point.
  - The contents of this array are determined by the CircMode variable CIRC_MODE_BORDER.
- The EndPoint array, **PosCircEndAbs**, contains the absolute end point for each axis in the group.
  - The absolute end point of the axis stored in position 0 (IdentInGroup) of the group is 90.0.
  - The absolute end point of the axis stored in position 1 of the group is 90.0.
- PathChoice is only relevant when CircMode is set to CIRC_MODE_CENTER.
  - In this case, this parameter is not used.
- The maximum velocity is specified by variable **MaxVel** and is specified in User units / sec.
- The maximum acceleration and deceleration are specified by variables **MaxAcc** and **MaxDec** and are specified in User units / sec^2.
- The maximum jerk is not supported and can be set to a value of 0.
- The coordinate system is ACS.
- The BufferMode is set to BM_BUFFERED, indicating the move is buffered.
  - See "Buffer Modes" (➔ p. 371).
- The TransitionMode is set to TM_NONE, indicating no transition mode is used.
  - See "Transition Between Moves" (➔ p. 840).
- The TransParam array is required.
  - The contents can be set to 0 because the transition mode is not being used.
  - The TransParam array is a 2-element array containing the corner distance and velocity for the transition.

15.1.3.3.6.2 Relative Circular Move

This is a relative circular move whose end point is (90, 90) from the end of the first move.

- CircMode specifies that the aux point (0, 90) is the relative center of the circle.
- The BufferMode input is set to **Buffered**; this move waits for the first move to complete before it begins executing.
- Call MC_MoveCircRel.

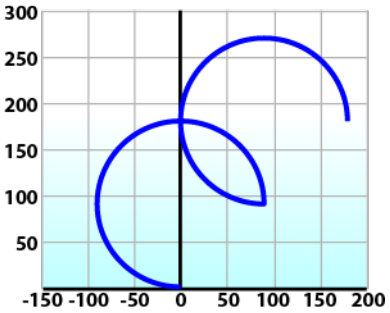
```
PosCircAuxRel[0] :=  0;   // Relative center of the circle.
PosCircAuxRel[1] := 90;
PosCircEndRel[0] := 90;   // Relative end point.
PosCircEndRel[1] := 90;   // Start pt 90,90 + rel 90,90 -> 180,180
absolute end pt

Inst_MC_MoveCircRel(TRUE, Group1_ref, CIRC_MODE_CENTER, PosCircAuxRel,
PosCircEndRel, CIRC_PATH_CLKWISE, MaxVel, MaxAcc, MaxDec, 0, CS_ACS, BM_
BUFFERED, TM_NONE, TransParam);
```

In the example, all the variables have the same meaning as the circular absolute example except:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

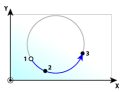
- CircMode is defined as CIRC_MODE_CENTER.
  - This mode indicates that the AuxPoint array input indicates the center point of the circle.
  - See "Circular Moves Diagrams" (➔ p. 829) about CircMode movement options.
- The AuxPoint array, **PosCircAuxRel**, defines the relative center point of the circle.
  - The contents of this array are determined by the CircMode variable, CIRC_MODE_CENTER.
- The EndPoint array, **PosCircEndRel**, contains the relative end point for each axis in the group.
  - The relative end point of the axis stored in position 0 (IdentInGroup) of the group is 90.0.
  - The relative end point of the axis stored in position 1 of the group will be 90.0.
- PathChoice is relevant when CircMode is set to CIRC_MODE_CENTER.
  - In this case, PathChoice is CIRC_PATH_CLKWISE which specifies the direction of the path.



15.1.3.3.7.3 Circular Moves Diagrams

15.2 CircMode = BORDER

- The user defines the end point and a border point (= input AuxPoint) on the sector of the circle which the machine will traverse.
- For Relative mode, both points are defined relative to the starting point.
- Advantage
  - The border point can usually be reached by the machine, i.e., it can be taught.
- Disadvantage
  - Restricted to angles < 360° in one single command.



1. Starting point.
2. Border point.
3. End point.

15.3 CircMode = CENTER

- The user defines the end point and center point (= input AuxPoint) of the circle.
- The input PathChoice defines clockwise or counter-clockwise motion.
- For Relative mode, both points are defined relative to the starting point.
- Advantage
  - Full 360° moves are possible.
- Disadvantages
  - Cannot perform zero-distance moves.
  - Over-determination of the circle equation.



1. Starting point.
2. Center point.
3. End point.
4. Clockwise move.
5. Counter-clockwise move.

A 360° move is performed if the end point is the same as the start point.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



1. Starting and Ending point.
2. Center point.
3. Clockwise move.
4. Counter-clockwise move.

15.3.0.0.1.1 Precision Requirements for Circular Move Input Parameters

The input parameters to MC_MoveCircAbs and MC_MoveCircRel are validated when constructing a circle.

- The distance of the center point from the start and end points is checked.
- Ideally, the two distances are the same but calculation errors or input data precision can cause the two distances to be slightly different.
- If the validation fails, PLCopen error 50 (Cannot construct a circle with specified parameters) is returned from the function block.

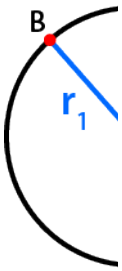
15.4 Parameter Validation

Input parameters are validated using these methods:

See "PLCopen Function Block ErrorIDs" (➔ p. 343).

Here is a circle we want to create.

The distances from the center to the start and end point are measured as r_1 and r_2 respectively.



A. Center point.
B. Start point.
C. End point.

The methods for validating the circle parameters are:

- "Default Validation Method" (➔ p. 830)
- "Validation with Custom Tolerance" (➔ p. 831)

NOTE

If MC_GRP_PARAM_CIRCLE_TOLERANCE is set to zero (0), the default validation method will be used.

15.5 Default Validation Method

- The difference between the two measurements must not be larger than one part in 100,000.
- If all positions are specified to six significant digits, this requirement is met.
- Assuming that r_2 is greater than r_1, we can write r_2 in terms of r_1 and a small deviation value named **ε**:

$$r_2 = r_1(1+\varepsilon)$$

- If **ε** exceeds a value of 10-5, the validation fails.
- A PLCopen error 50 is returned from the function block.

Examples of Default Validation

If the desired circle has a radius of:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- 50 user units, the center must be specified with a precision of 0.0005 user units.
- 2,000 user units, the center must be specified with a precision of 0.02 user units.
- 500,000 user units, the center must be specified with a precision of 5 user units.

15.6 Validation with Custom Tolerance

A custom tolerance can be specified via the group parameter MC_GRP_PARAM_CIRCLE_TOLERANCE.

- See "Axes Group Parameters" (➔ p. 347).
- This parameter represents the amount of error allowed, measured in user units.
- The parameter can be set by using MC_GrpWriteParam and can be read via MC_GrpReadParam.
- Assuming that r_2 is greater than r_1, we can write r_2 in terms of r_1 and a deviation value named **Δ**:

```
r2= r1 + Δ
```

- The validation fails if **Δ** exceeds the value of the group parameter MC_GRP_PARAM_CIRCLE_TOLERANCE.
- A PLCopen error 50 is returned from the function block.

Examples of Validation using a Custom Tolerance

- If r_1 has a value of 50 user units and the circle tolerance parameter is .1 user units, then r_2 must be in the range of 49.9 to 50.1 user units.
- If r_1 has a value of 2000 user units and the circle tolerance parameter is 1 user units, then r_2 must be in the range of 1999 to 2001 user units.

TIP

Use LREAL variables and LREAL versions of math functions when calculating the desired circle parameters inside a KAS application.
The LREAL versions of functions usually have an 'l' at the end of their name.
Example: The LREAL version of cos is cosl.
This helps avoid errors.

15.6.0.0.1.1 Perform a Complete Circular Move

Use this procedure to perform a full circle.

1. Call either the MC_MoveCircAbs or MC_MoveCircRel function block.
2. Set **CircMode** to Center (**MC_CIRC_MODE_CENTER**).
3. Set the **EndPoint** to be the same as the start point.
4. Repeat as necessary for multiple rotations.

NOTE

MC_CIRC_MODE_BORDER cannot be used because it is limited to angles <360°.

15.6.0.0.2 Perform a Linear Move

Linear moves can be programmed using absolute or relative positions using these function blocks:

- MC_MoveLinAbs commands interpolated linear movement on an axes group to the specified absolute positions.
- MC_MoveLinRel commands interpolated linear movement on an axes group to the specified relative positions.

Prior to performing any coordinated moves, some setup is needed.

- See "Create a Linear or Circular Coordinated Motion Application" (➔ p. 824).
- Once these steps have been performed, a linear move can be performed.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

In these examples, two linear moves are performed.

- "Absolute Linear Move" (➔ p. 832)
- "Relative Linear Move" (➔ p. 832)
- "Linear Move With More Than Two Axes" (➔ p. 833)

15.6.0.0.3.1 Absolute Linear Move

In this move, an absolute linear move goes from (0, 0) to (100, 200).

- Call MC_MoveLinAbs.
  - PositionArray is an array of absolute end positions containing one position for each axis in the group.
  - The input velocity, acceleration, deceleration, and jerk establish the maximum values for the move.

```
PosArrayAbs[0] := 100;
PosArrayAbs[1] := 200;
TransParam[0]  :=   0;

Inst_MC_MoveLinRel(TRUE, Group1_ref, PosArrayAbs, MaxVel, MaxAcc,
MaxDec, 0, CS_ACS, BM_BUFFERED, TM_NONE, TransParam);
```

In the example, a linear move is performed on axis group **Group1_ref**.

- PosArrayAbs contains the absolute end points of the axes in the group.
  - The axis stored in position 0 (IdentInGroup) of the group is moved to 100.0.
  - The axis stored in postiion 1 of the group is moved to 200.0.
- The maximum velocity is specified by variable **MaxVel** and is specified in User units / sec.
- The maximum acceleration and deceleration are specified by variables **MaxAcc** and **MaxDec** and are specified in User units / sec^2.
- The coordinate system is ACS.
- The BufferMode is set to BM_BUFFERED, indicating the move is buffered.
  - See "Buffer Modes" (➔ p. 371).
- The TransitionMode is set to TM_NONE, indicating no transition mode is used.
  - See "Transition Between Moves" (➔ p. 840).
- The TransParam array is required.
  - The contents can be set to 0 because the transition mode is not being used.
  - There has to be one array entry for each axis in the group.

15.6.0.0.4.2 Relative Linear Move

In this move, a relative linear move goes a distance of (-75, 50) from the end of the first move.

- Call MC_MoveLinRel.
- The Distance input is an array of distances, one distance for each axis in the group.
- The input velocity, acceleration, deceleration, and jerk establish the maximum values for the move.

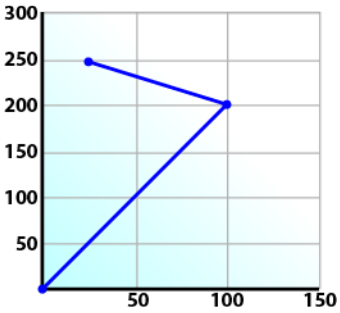
```
DistArrayRel[0] := -75.0;   // Start pt 100 - rel 75 ->  25 absolute end
pt
DistArrayRel[1] :=  50.0;   // Start pt 200 + rel 50 -> 250 absolute end
pt
TransParam[0]   :=   0;

Inst_MC_MoveLinRel(TRUE, Group1_ref, DistArrayRel, MaxVel, MaxAcc,
MaxDec, 0, CS_ACS, BM_BUFFERED, TM_NONE, TransParam);
```

In the example, all the variables have the same meaning as the absolute linear example except DistArrayRel.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- DistArrayRel contains the relative distance to move for each axis in the group.
- The axis stored in position 0 (IdentInGroup) of the group is moved a distance of -75.0.
- The axis stored in position 1 of the group is moved a distance of 50.0.



15.6.0.0.5.3 Linear Move With More Than Two Axes

NOTE

The dimensionality of the move is determined by the number of axes mapped to the group.
This implies that a group which could hold a maximum of three or more axes does two dimensional moves if it only has two valid axes mapped to it.

To perform higher dimensional moves, additional axes must be added to the group.

The procedure for this are in "Create a Linear or Circular Coordinated Motion Application" (→ p. 824).

After the additional axes are added, perform these steps.

1. From the Dictionary, update the array size of the variable being passed (PosArrayAbs and DistArrayRel in the "Absolute Linear Move" (→ p. 832) and "Relative Linear Move" (→ p. 832) examples) to the Position input so its length matches the maximum number of axes allowed in the group.
2. Set the desired values for the additional axes in the now larger position arrays.

15.6.0.1 Differences between MC_GrpHalt and MC_GrpStop

MC_GrpHalt and MC_GrpStop both perform a controlled motion stop of all axes in an axes group.

These are the differences between the function blocks.

MC_GrpHalt	MC_GrpStop
MC_GrpHalt can be aborted by these other commands: • MC_GrpHalt • MC_GrpStop • MC_Move command (e.g., MC_MoveLinAbs, MC_MoveCircRel, etc.)	MC_GrpStop cannot be aborted by any other command.
While MC_GrpHalt is active, the group state is **GroupMoving**.	• While MC_GrpStop is active, the group state is **GroupStopping**. • See "Group State Diagrams" (→ p. 820).
When MC_GrpHalt is complete, the group state goes to **GroupStandBy**.	• When MC_GrpStop is complete, the DONE output is TRUE. • The EXECUTE input must be set to FALSE before the group state can go to **GroupStandBy**.

See Also

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "What MC_GrpHalt Does" (➔ p. 834)
- "What MC_GrpStop Does" (➔ p. 836)

15.6.0.1.1 What MC_GrpHalt Does

MC_GrpHalt performs a controlled motion stop of all axes in a coordinated motion group.

See "Differences between MC_GrpHalt and MC_GrpStop" (➔ p. 833).

Example

This example demonstrates a linear coordinated move with a starting point of (0,0) and an endpoint of (200, 0).

- The first and second MC_GrpHalt commands are called while the linear coordinated move is still moving to the endpoint.
- In the oscilloscope, the second call to MC_GrpHalt aborts the first MC_GrpHalt as seen by the new deceleration rate.
- The path velocity reaches zero (approximately (60, 0)) before the linear coordinated move reaches its endpoint value.

NOTE

For demonstration purposes, it is assumed that the axes and group have been properly setup and configured. Example steps to setup coordinated motion are listed in "Create a Linear or Circular Coordinated Motion Application" (➔ p. 824).

```
1: // Perform Linear ABSOLUTE move start (0,0) end (200,0)
PosAbs[0]:= 0;
PosAbs[1]:= 200;
PosAbs[2]:= 0;

pathVelocity := Inst_MC_GrpReadCmdVel2.PathVelocity;

Inst_MC_MoveLinAbs(TRUE, Group1_ref, PosAbs, Velocity, Acceleration,
Deceleration, Jerk, CS_ACS, BM_ABORTING, TM_NONE, TransParam);
Inst_TON3(true, t#600ms);   //Allow for the move to reach path velocity
before calling MC_GrpStop
Inst_TON3.Q  THEN
     MC_MoveCounter := MC_MoveCounter + 1;
     Inst_TON3(false, t#100ms);
END_IF;

2: //Perform a halt on the group
halt_deceleration := 50.0;

pathVelocity := Inst_MC_GrpReadCmdVel2.PathVelocity;

Inst_MC_GrpHalt(TRUE, Group1_ref, halt_deceleration, default_jerk);

Inst_TON3(true, t#200ms);   //Allow for first halt deceleration rate to be
captured on the scope
IF Inst_TON3.Q THEN
     MC_MoveCounter := MC_MoveCounter + 1;
     Inst_TON3(false, t#100ms);
END_IF;
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
3: // Perform a second halt increasing the deceleration value.
// The second call to MC_GrpHalt will abort the first MC_GrpHalt
halt_deceleration := 200.0;

pathVelocity := Inst_MC_GrpReadCmdVel2.PathVelocity;

Inst_MC_GrpHalt1(TRUE, Group1_ref, halt_deceleration, default_jerk);

Inst_TON3(true, t#200ms);
IF ((Inst_TON3.Q) and (Inst_MC_GrpHalt1.Done Or Inst_MC_GrpHalt1.Error)) THEN
     Inst_TON3(false, t#100ms);
     Inst_MC_MoveLinAbs(FALSE, Group1_ref, PosAbs, default_velocity, default_
acceleration, default_deceleration, default_jerk,
        CS_ACS, BM_ABORTING, TM_NONE, TransParam);
     Inst_MC_GrpHalt1(FALSE, Group1_ref, default_deceleration, default_jerk);
END_IF;
```

When MC_GrpHalt is called in the example:

- The current move is aborted.
  - A controlled motion stop is applied to axes group **Group1_ref**.
  - The deceleration value, **halt _deceleration** is set to 50.0 User units / sec^2 on the first call and 200.0 User units / sec^2 on the second call.
  - It is applied to the path velocity until it reaches 0 (zero) when the MC_GrpHalt command is issued during a coordinated motion move.
- The group state is **GroupMoving** while the coordinated move is decelerating.
- The group state goes to **GroupStandBy** once the path velocity reaches 0 (zero).
- Any coordinated moves in the buffer are flushed.
- New coordinated moves can be queued up upon completion of the MC_GrpHalt command.

NOTE

This behavior is different than the MC_GrpStop command.
See "Differences between MC_GrpHalt and MC_GrpStop" (➔ p. 833).

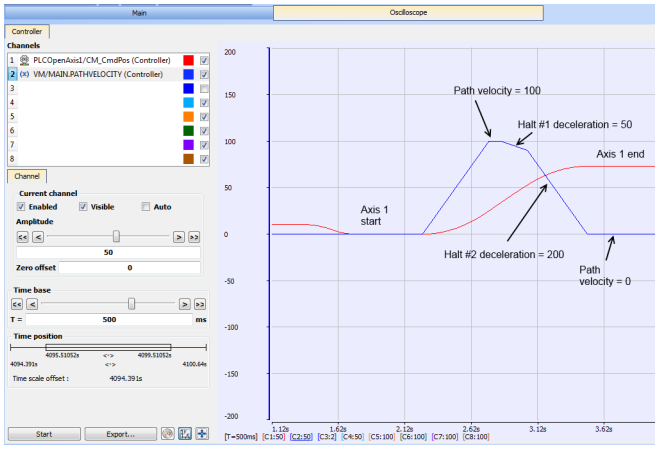


Figure 19-3: Oscilloscope Representation of linear coordinated move with a MC_GrpHalt command called twice.

15.6.1 Exceptions

- The deceleration rate from the MC_GrpHalt function block is only applied to the absolute and relative coordinated motion moves.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

    - Direct coordinated motion moves use the default deceleration value as defined by the AxisRef.
- MC_GrpHalt cannot occur if a group is not enabled.
- An MC_GrpHalt command may be aborted by another MC_GrpHalt command, a MC_GrpStop command or a MC_Move* command.
- When there are two coordinated motion moves (**Active** and **Next**) in the queue, and the path velocity does not reach 0 (zero) before the end of the **Active** move, the path velocity continues to reach zero during the **Next** move.
- The deceleration rate is increased if there is only one **Active** coordinated move and the path velocity cannot reach zero before the endpoint.
    - This prevents overshooting the endpoint.
- The halt action pauses during S-Curve Corner transitions.
    - Any halt in progress continues after the S-Curve Corner transition is complete.
    - If a halt is started during an S-Curve Corner transition, the halt action starts after the S-Curve Corner transition completes.
    - While a halt action is in progress, there may be a discontinuity in acceleration at the endpoints of S-Curve Corner transition.
- An MC_GrpHalt command does not prevent performing a single axis motion.

15.6.1.0.1 What MC_GrpStop Does

MC_GrpStop performs a controlled motion stop of all axes in a coordinated motion group.

See "Differences between MC_GrpHalt and MC_GrpStop" (➔ p. 833).

Example

This example demonstrates a linear coordinated move with a starting point of (0,0) and an endpoint of (200, 0).

- MC_GrpStop is called while the linear coordinated move is still moving to the endpoint.
- In the oscilloscope, the path velocity reaches zero (approximately [100,0]) before the linear coordinated move reaches its endpoint value.

Linear Move Parameters	MC_GroupStop Parameters
Velocity = 100	N/A
Acceleration = 200	N/A
Deceleration = 200	Declaration - 75
Jerk = 0	Jerk = 0

NOTE

For demonstration purposes, it is assumed that the axes and group have been properly setup and configured. Example steps to setup coordinated motion are listed in "Create a Linear or Circular Coordinated Motion Application" (➔ p. 824).

```
Inst_MC_GrpReadCmdPos(TRUE, Group1_ref, CS_ACS, CmdPositionArray);
Inst_MC_GrpReadCmdVel2(TRUE, Group1_ref, CS_ACS, VelocityArray);
CASE MC_MoveCounter OF0: // Enable the group
Inst_MC_GrpEnable(TRUE, Group1_ref);
Inst_TON3( true, t#1500ms );   // Allow for turning on the scopeIF ((Inst_
TON3.Q) and (Inst_MC_GrpEnable.Done OR Inst_MC_GrpEnable.Error)) THEN

     Inst_TON3(false, t#1s);
     IF (Inst_MC_GrpEnable.Error) THENPrintMessage( LEVEL_INFO (*DINT*),
'MC_GrpEnable failed - ErrorID: ' + any_to_string(Inst_MC_
GrpEnable.ErrorID));
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
     END_IF;

     Inst_MC_GrpEnable(FALSE, Group1_ref);
     MC_MoveCounter := MC_MoveCounter + 1;
END_IF;

1: // Perform Linear ABSOLUTE move start (0,0) end (200,0)
PosAbs[0]:= 0;
PosAbs[1]:= 200;
PosAbs[2]:= 0;

pathVelocity := Inst_MC_GrpReadCmdVel2.PathVelocity;

Inst_MC_MoveLinAbs(TRUE, Group1_ref, PosAbs,Velocity, Acceleration,
Deceleration, Jerk,
     CS_ACS, BM_ABORTING, TM_NONE, TransParam);

Inst_TON3(true, t#600ms);      //Allow for the move to reach path velocity before
calling MC_GrpStopIF Inst_TON3.Q  THEN
     MC_MoveCounter := MC_MoveCounter + 1;
     Inst_TON3(false, t#100ms);
END_IF;

2: //Perform a stop on the group
stop_deceleration := 75.0;

pathVelocity := Inst_MC_GrpReadCmdVel2.PathVelocity;

Inst_MC_GrpStop(TRUE, Group1_ref, stop_deceleration, default_jerk );

Inst_TON3( true, t#200ms );
IF ((Inst_TON3.Q) AND (Inst_MC_GrpStop.Done Or Inst_MC_GrpStop.Error)) THENIF
Inst_MC_GrpStop.Error THENPrintMessage( LEVEL_INFO (*DINT*), 'Step '+any_to_
string(MC_MoveCounter)+',MC_GrpStop ERROR. ErrorID('+any_to_string(Inst_MC_
GrpStop.ErrorID)+'), Description:'+MC_ErrorDescription(any_to_int(Inst_MC_
GrpStop.ErrorID)));
     END_IF;

     Inst_TON3(false, t#100ms);

     Inst_MC_MoveLinAbs(FALSE, Group1_ref, PosAbs, default_velocity, default_
acceleration, default_deceleration, default_jerk,
CS_ACS, BM_ABORTING, TM_NONE, TransParam);

     Inst_MC_GrpStop(FALSE, Group1_ref, stop_deceleration, default_jerk);
END_IF;
```

When MC_GrpStop is called in the example:

- The current move is halted.
- A controlled motion stop is applied to axes group **Group1_ref**.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The deceleration value, **stop_deceleration**, is set to 75.0 User units / sec^2.
  - It is applied to the path velocity until it reaches 0 (zero).
- The group state is **GroupStopping** when the MC_GrpStop function block becomes active.
- While the axes group is in the **GroupStopping** state, no other function blocks can perform any motion on the same axes group.
  - Once the path velocity reaches 0, the **Done** output is TRUE.
- The **Execute** input must be set to FALSE before the group state can go to **GroupStandBy**.
- Any coordinated moves in the buffer are flushed.
- New coordinated moves can be queued up upon completion of the MC_GrpStop command.

NOTE

This behavior is different than the MC_GrpHalt command.
See "Differences between MC_GrpHalt and MC_GrpStop" (➔ p. 833).

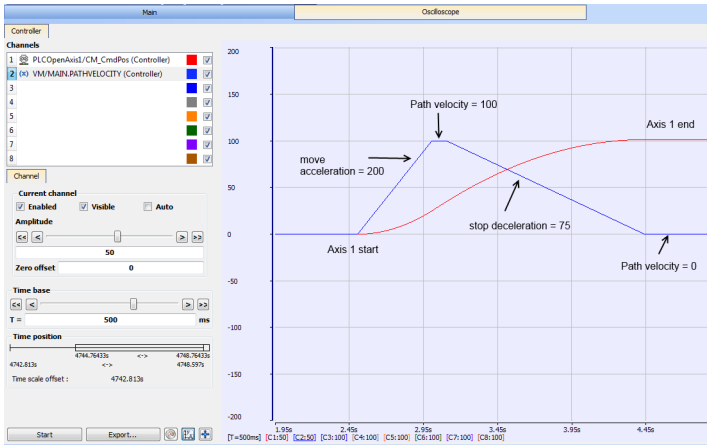


Figure 19-4: Oscilloscope Representation of linear coordinated move with a MC_GrpStop.

15.6.2 Exceptions

- The deceleration rate from the MC_GrpStop function block is only applied to the absolute and relative moves.
  - Direct moves use the default deceleration value as defined by the AxisRef.
- MC_GrpStop cannot occur if a group is not enabled.
- An MC_GrpStop command **cannot** be aborted by any other commands.
- When there are two coordinated motion moves (**Active** and **Next**) in the queue, and the path velocity does not reach 0 (zero) before the end of the **Active** move, the path velocity continues to reach zero during the **Next** move.
- The deceleration rate is increased if there is only one **Active** coordinated move and the path velocity cannot reach zero before the endpoint.
  - This prevents overshooting the endpoint.
- The stop action pauses during S-Curve Corner transitions.
  - Any stop in progress continues after the S-Curve Corner transition is complete.
  - If a stop is started during an S-Curve Corner transition, the stop action starts after the S-Curve Corner transition completes.
  - While a stop action is in progress, there may be a discontinuity in acceleration at the endpoints of the S-Curve Corner transition.
- An MC_GrpStop command does not prevent performing a single axis motion.

15.6.2.1 Handling Axis Errors

Coordinated motion error handling is configurable on a per axis group basis.

- When a PLCopenaxis error occurs, the "Default Error Handling Behavior" (➔ p. 839) is for all axes in the group stop.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- This means motion interpolation stops, active and next queues are cleared, and the group state **GroupErrorStop** is enabled.
- The position loop on the drives in the group are opened and the drives are disabled.

The "Optional Error Handling Behavior" (➔ p. 839) when a PLCopen axis error occurs is for only the PLCopen axis that caused the error to have its position loop opened and that drive is disabled.

- The application is expected to control the motion for the remaining axes in the group.
- The optional feature may be used when stopping all axes is worse than having one axis stop.

15.6.2.1.1 Optional Behavior

- To configure the optional behavior, use the function block MC_GrpWriteBoolPar and set the parameter **IGNORE_AXIS_ESTOP (1000)** to TRUE.
- To check if the optional behavior has been set, call MC_GrpReadBoolPar and read the state of parameter **IGNORE_AXIS_ESTOP (1000)**.
    - If the result is TRUE, the optional behavior is enabled.

The types of PLCopen Axis Errors that affect coordinated motion include:

- Drive errors.
- Drive communication errors.
- User causes E-stop using MC_EStop on an axis in a group.

15.6.2.1.2 Default and Optional Error Handling Behavior

15.6.2.1.3.1 Default Error Handling Behavior

In the default motion error handling case, a user can detect errors using any of these group function blocks:

- MC_GrpReadStatus - Output GroupErrorStop is true if a group error has occurred.
- MC_GrpReadError - Error output is TRUE.
    - ErrorID output is set to 12 if the error is due to E-stop.
- The originating Coordinated Motion command (MC_MoveLinAbs, MC_MoveCircAbs, etc.) will return Error ID 12

To determine which PLCopen axis has generated the error, call MC_ReadAxisErr for each axis in a group.

- The output ErrorID is set to 12.
- MC_ReadStatus for each axis in the group can be called where the output ErrorStop is TRUE if in a E-stop condition.
- Once an error has been detected, the error must first be resolved.
- MC_GrpReset can be called to reset all PLCopen axis errors.
    - This also resets the group status from GroupErrorStop to GroupStandby.
    - The Done output of MC_GrpReset is TRUE if all axis errors have been reset.
    - This function block may take up to 3 seconds to reset some error conditions.

15.6.2.1.4.2 Optional Error Handling Behavior

TIP

This is configurable on a per group basis.

In the optional motion error handling case, the group status GroupErrorStop is not set when an axis is in error.

The application is responsible for monitoring and handling error conditions. The remaining axes in the group will continue moving.

Errors can be detected using this group function block:

- MC_GrpReadBoolPar - Read the result of parameter ID AXIS_ESTOP_ACTIVE (1001).
    - If the result is TRUE, an axis error exists.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The group function blocks used for the default case of detecting errors (MC_GrpReadStatus and MC_GrpReadError) do not return an error.

To determine which PLCopen axis has generated the error, call MC_ReadAxisErr for each axis in a group.

- The output ErrorID is set to 12.
- MC_ReadStatus for each axis in the group can be called where the output ErrorStop is TRUE if in a E-stop condition.
- Once an error has been detected, the error must first be resolved.
- MC_ResetError can then be called on each axis in error.
  - This function only sends a request to the drive to clear any error.
  - The error is not yet reset when this function returns.
  - MC_ReadStatus still has to be called to verify that the drive error has been resolved.

15.6.2.1.5 Recovery of the System State After an Axis Error

Recovery from axis errors is more complex with the addition of Coordinated Axes Groups (PLCopen Part 4) to KAS Runtime. This is because of additions in the operation of PLCopen Motion and Coordinated motion. This includes:

- Addition of the Coordinated motion Boolean variable that allows the Runtime to IGNORE_AXIS_ESTOP (ID number 1000).
- Addition of the Coordinated motion Boolean variable that allows the Application to read the state of the Runtime Axis errors: AXIS_ESTOP_ACTIVE (ID number 1001).
- Addition of MC_GrpReadBoolPar.
- Addition of MC_GrpWriteBoolPar.

When the application needs to take control of the default behavior where all axes in a group stop when any of them detect an Estop condition, and the Estop condition does not in itself stop all axes (Loss of EtherCAT communication), the application can set the Group Boolean parameter IGNORE_AXIS_ESTOP (ID := 1000) using MC_GrpWriteBoolPar().

- When set to TRUE, the runtime tries to keep axes in the same group as a faulting axis still able to be commanded.
- The intent is to not give up control of the commanded motion, but allow the application to substitute error handling motion that allows a group of axes to stop in a controlled manner.
- Example: Command the remaining axes to a Home position or a relative move away from the position of the faulted axis.

Once the IGNORE_AXIS ESTOP parameter is set TRUE, the Group of axes does not enter the ERRORSTOP state when an axis encounters an ESTOP condition but remain in STANDBY or MOVING state.

- Instead, the AXIS_ESTOP_ACTIVE (ID := 1001) parameter can be monitored to trigger a response that aborts the current operation and takes control of the axes that can still be controlled.
- To determine which axis has faulted, the MC_ReadStatus() FB is used to monitor each axis and select the appropriate single axis or Coordinated motion commands to execute to take control of the motion and recover from the error.

To restart Coordinated Motion when the error recovery is completed and the faulting axis restored to normal operation, an MC_GrpSetPos() command must be issued to tell the Coordinated Motion Engine where the axes in the faulted Axes Group, following restoration of the single axis fault(s).

This command must be issued in order to reset the faulted axis.

Alternatively, if the MC_GrpWriteBoolPar() function is issued to set the IGNORE_AXIS_ESTOP parameter to FALSE, the Group enters the ERRORSTOP state if an Axis error has not been reset, and the default behavior and default usage of MC_GrpResetError() function can be used to clear faults instead of the MC_GrpSetPos() function.

15.6.2.2 Transition Between Moves

A transition mode must be specified when a new move is appended to a move that is already in progress.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Different transition parameters may be required, depending on the transition mode.
- This characterizes the contour of the transition segment.

The supported transition modes are:

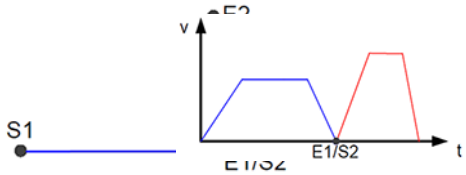
- "No Transition (TM_NONE)" (➔ p. 841)
- "Corner Distance (TM_CORNER_DISTANCE)" (➔ p. 841)
- "S-Curve Corner (TM_SCURVE_CORNER) " (➔ p. 842)

15.6.2.2.1.1 Transition Mode Parameters

Parameters	Number of Transition Parameter Array Elements	Transition Parameter Index	Transition Parameter Name	Transition Parameter Description	Units
TM_NONE	0	N/A	N/A	No transition.	N/A
TM_CORNER_DISTANCE	2	0	Corner Distance	Distance to the corner of the deviation and the return point from the original contour.	User units
TM_CORNER_DISTANCE	2	1	Velocity	The velocity value of the transition segment.	User units / sec
TM_SCURVE_CORNER	1	0	Corner Distance	Distance to the corner of the deviation and the return point from the original contour.	User units

15.6.2.2.2 No Transition (TM_NONE)

Insert no transition contour segment.

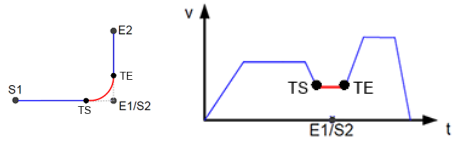


Motion Path **Motion Velocity Profile**

- The motion blocks are not modified and no transition curve is inserted using this mode.
  - This is the only possible transition mode for the "Buffer Modes" (➔ p. 371).
- No transition parameters are used for this transition mode.

15.6.2.2.3 Corner Distance (TM_CORNER_DISTANCE)

Transition with given corner distance.



Motion Path **Motion Velocity Profile**

The corner distance transition mode is specified using **TM_CORNER_DISTANCE** as the transition mode for a motion block.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

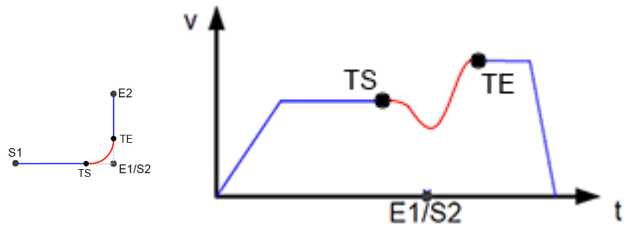
15.6.2.2.4.1 Transition Parameter Index

Transition Parameter Index	Name	Description
0	Corner Distance	Distance to the corner of the deviation and the return point from the original contour.
1	Velocity	The velocity value of the transition segment.

- Corner Distance transitions are handled differently, depending upon whether the connecting moves are lines or arcs, and all of the possible combinations, (line-line, arc-arc, line-arc, arc-line).
  - Arc-to-Arc transitions shorten the arc with the larger radius by the corner distance.
    - See "Arc-to-Arc Transitions" (➔ p. 843).
  - Line-to-Arc and Arc-to-Line transitions shorten the linear move.
    - See "Line-to-Arc and Arc-to-Line Transitions" (➔ p. 846).
  - Line-to-Line transitions shorten the next move by the corner distance.
    - See "Line-to-Line Transitions" (➔ p. 848).

15.6.2.2.5 S-Curve Corner (TM_SCURVE_CORNER)

Transition with given corner distance, using Jerk for each axis profile.



Motion Path	Motion Velocity Profile

The S-Curve Corner transition mode is specified using **TM_SCURVE_CORNER** as the transition mode for a motion block.

15.6.2.2.6.1 Transition Parameter Index

Transition Parameter Index	Name	Description
0	Corner Distance	Distance to the corner of the deviation and the return point from the original contour.

- All S-Curve Corner transitions start and end the transition with zero acceleration.
  - If an S-Curve Corner transition is used with buffer mode BM_ABORTING and during acceleration or deceleration phases of the first move, then the axes experience an acceleration discontinuity at beginning of the transition.
- Only Line-to-Line transitions are supported with S-Curve Corner.
  - Line-arc, arc-arc, arc-line are **not** supported.
- Line-to-Line transitions shorten the next move by the corner distance.
  - See "S-Curve Corner Line-to-Line Transitions" (➔ p. 849).
- Changes to the velocity override during the transition do not take effect until after the transition has completed.
- S-Curve Corner transitions only depend on the corner distance, the incoming and outgoing move velocities, and the incoming and outgoing move directions.
  - The transition may exceed the maximum specified values for velocity, acceleration, deceleration, and jerk of the incoming and outgoing moves.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Jerk and peak acceleration increase with a larger angle between the incoming and outgoing move directions.
- Jerk and peak acceleration decrease with a larger corner distance.

- To approximate the path peak acceleration and jerk during the transition, we can use these equations:
  - V_{avg} = The average of the slew speeds of the incoming and outgoing moves.
  - θ = The angle between incoming and outgoing moves.
  - CD = Corner distance.
  - $typical\ accel = \frac{8}{7} \cdot \frac{{v_{avg}}^2}{CD} \cdot \sqrt{\sin\left(\frac{\theta}{2}\right)}$
  - $typical\ jerk = 16 \cdot \frac{{v_{avg}}^3}{CD^2} \cdot \sin^2\left(\frac{\theta}{2}\right)$
  - If `actual accel = k_a typical accel` and `actual jerk = k_j typical jerk`, then:
    - k_a has an average value of 1 with a standard deviation of 0.5. Values of k_a have been observed as large as 3.
    - k_j has an average value of 1 with a standard deviation of 1.1. Values of k_j have been observed as large as 12.

NOTE

The formulas are approximate guidelines.
Your application may experience different behavior.

See Also

- MC_MoveCircAbs
- MC_MoveCircRel
- MC_MoveLinAbs
- MC_MoveLinRel

15.6.2.2.7 Arc-to-Arc Transitions

In arc-to-arc transition mode, the transitions are processed as.

1. The arc with the larger radius is shortened by an arc length equal to the corner distance.
2. Using this point, a transition move is calculated such that it is tangent to both arcs.
3. The arc with the smaller radius is shortened to the point where it is tangent with the transition move.

The transition move that connects the two original arcs derives its velocity from the velocity transition parameter and the acceleration and deceleration values are derived from the next move.

These are the special cases for Arc-to-Arc transitions:

- "Intersection Transition" (→ p. 843)
- "Line Segment Transition" (→ p. 844)
- "Same Circle: Opposite Direction Transition" (→ p. 844)
  - "n-Degree: Arcs in Opposite Direction" (→ p. 844)
- "Same Circle: Same Direction Transition" (→ p. 845)
  - "n-Degree: Arcs in Same Direction" (→ p. 845)
- "Tangent Transition: Opposite Direction" (→ p. 845)
- "Zero-distance Transition" (→ p. 846)

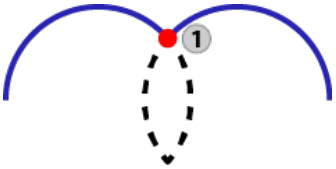
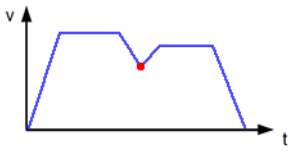
15.6.2.2.8.1 Intersection Transition

Click to view Intersection Transition

This occurs when the arcs intersect at two locations and the corner distance is equal to the arc length between the two intersections on the larger arc.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

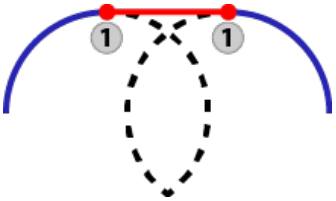
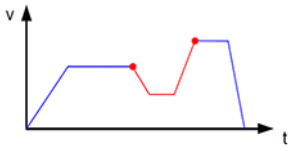
- The transition move is zero distance.
- If the buffer mode is **Buffered**, the path velocity goes to zero.
- All other buffer modes may cause a large acceleration and jerk.

Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing circular motion. **Black**: Trimmed path. 1. Intersection of the two arcs.	

15.6.2.2.9.2 Line Segment Transition

Click to view Line Segment Transition

- For certain values of corner distance, the transition arc has an infinite radius.
  - A line segment is used instead.

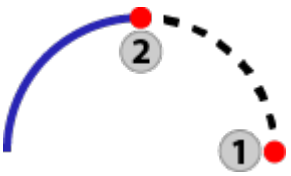
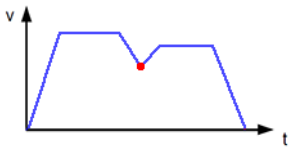
Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing circular motion. **Red**: Line Segment. 1. Start / endpoint of transition move.	

15.6.2.2.10.3 Same Circle: Opposite Direction Transition

Click to view Same Circle: Opposite Direction Transition

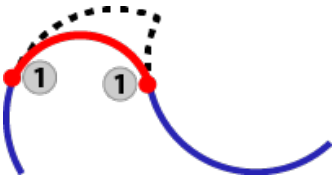
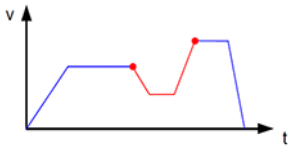
The arcs are shorted by an arc length equal to the corner distance.

- A zero distance transition is inserted if both arcs lie on the same circle and are in the opposite direction.
- If the buffer mode is **Buffered**, the path velocity goes to zero.
- All other buffer modes may cause a large acceleration and jerk.

Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing circular motion. 1. Original start / endpoint of arc move. 2. Start / endpoint of arc move.	

15.6.3 n-Degree: Arcs in Opposite Direction

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

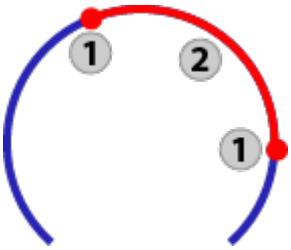
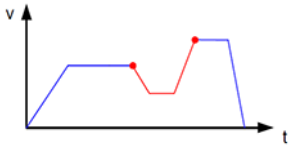
Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing circular motion. **Black**: Trimmed path. **Red**: Transition arc. 1. Start / endpoint of arc move.	

15.6.3.0.1.1 Same Circle: Same Direction Transition

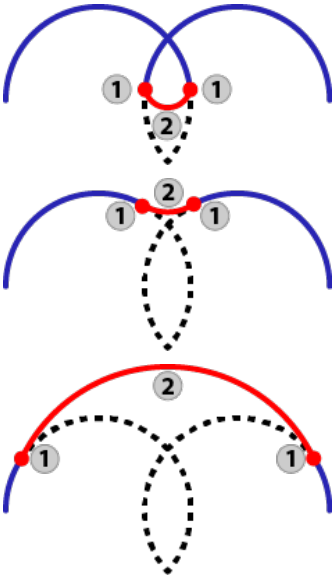
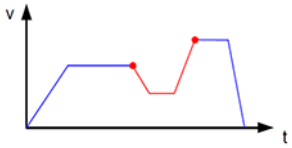
Click to view Same Circle: Same Direction Transition

A transition arc with a length of twice the corner distance is added if both arcs lie on the same circle and are in the same direction.

- This does not affect the path but does affect blending.

Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing circular motion. **Red**: Transition arc. 1. Start / endpoint of arc move. 2. Original start / end point.	

15.6.4 n-Degree: Arcs in Same Direction

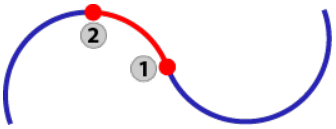
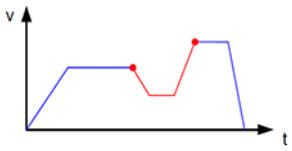
Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing circular motion. **Black**: Trimmed path. **Red**: Transition arc. 1. Start / endpoint of arc move. 2. Corner distance.	

15.6.4.0.1.1 Tangent Transition: Opposite Direction

Click to view Tangent Transition: Opposite Direction

A portion of the arc with the larger radius is replaced by an arc whose length is equal to the corner distance.

- If the arcs have the same radius, the incoming arc is treated as having a larger radius.
- This does not affect the path but does affect blending.

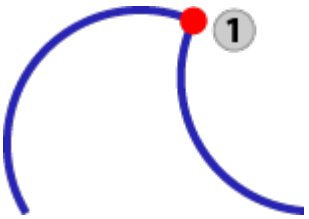
Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing circular motion. **Red**: Transition arc. 1. Original start / end point of moves. 2. Start / endpoint of arc move.	

15.6.4.0.2.2 Zero-distance Transition

Click to view Zero-distance Transition

A zero-distance transition move is inserted when the corner distance is zero.

- This only affects blending.
- If the buffer mode is **Buffered**, the path velocity goes to zero.
- All other buffer modes may cause a large acceleration and jerk.

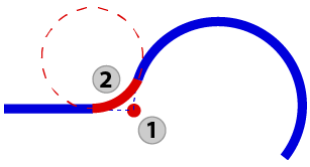
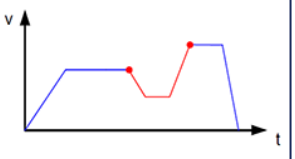
Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing circular motion. 1. Original start / endpoint of arc move.	User units / sec

15.6.4.0.3 Line-to-Arc and Arc-to-Line Transitions

With this transition mode, the linear move is shortened when one move is linear and the other is circular.

- The amount is specified by the corner distance.
- A circular arc connects the two moves, allowing for a smooth transition.
- The circular arc that connects the two moves derives its velocity from the transition parameter and the acceleration and deceleration values are derived from the next move.

15.6.5 n-Degree Transition

Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing linear and circular motion. **Red**: Transition arc. 1. Original start / endpoint of linear move. 2. Corner distance.	

15.6.5.0.1.1 Special Case Transitions

These are the special cases for Line-to-Arc and Arc-to-Line transitions:

- "Intersection Transition" (→ p. 847)
- "Tangent Transition" (→ p. 847)
- "Zero-distance Transition" (→ p. 847)

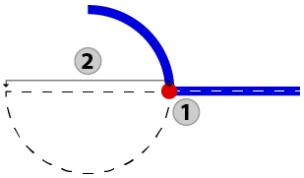
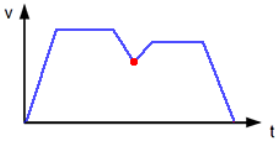
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.6.6 Intersection Transition

Click to view Intersection Transition

This is only when the arc intersects the line corner distance away from the point where the line and the arc meet.

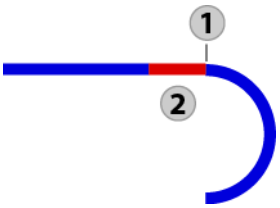
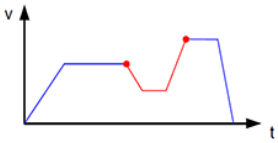
- The line intersects the arc a corner distance away from the beginning of the new move.
- The transition move is zero distance.
- If the buffer mode is **Buffered**, the path velocity goes to zero.
- All other buffer modes may cause a large acceleration and jerk.

Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Dashed**: Original shape. **Blue**: Incoming and outgoing linear and circular motion. **Red**: Zero distance transition. 1. Original endpoint of the 1st move. 2. Corner distance.	

15.6.7 Tangent Transition

Click to view Tangent Transition

- The linear move is shortened by the corner distance.
- A linear transition move is inserted to cover the distance that was removed.
- The line is tangent to the arc.

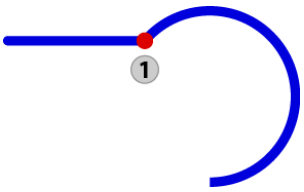
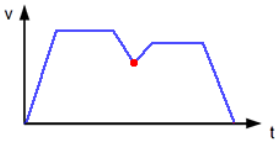
Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing linear and circular motion. **Red**: Transition line segment. 1. Original start / endpoint of linear move. 2. Corner distance.	

15.6.8 Zero-distance Transition

Click to view Zero-distance Transition

- A zero distance transition move is inserted.
  - This only affects blending.
- Special behavior for 0-distance transitions.
- If the buffer mode is **Buffered**, the path velocity goes to zero.
- All other buffer modes may cause a large acceleration and jerk.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

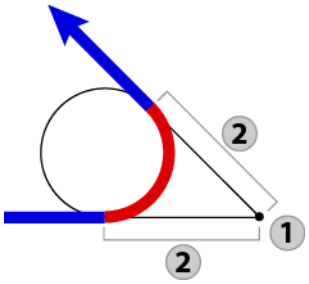
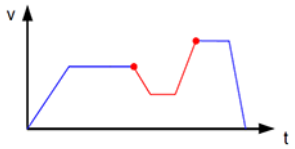
Motion Path - Line-to-Arc / Arc-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing linear and circular motion. **Red**: Zero distance transition. 1. Original endpoint of the 1st move.	

15.6.8.0.1 Line-to-Line Transitions

When both moves are linear, the current and next moves are shortened using this transition mode.

- The amount is specified by the corner distance.
- A circular arc connects the two moves, with exceptions.
  - This allows for a smooth transition.
  - See "n-Degree: Transition" (→ p. 848).
- The circular arc that connects the two moves derives its velocity from the transition parameter.
- The acceleration and deceleration values are derived from the next move.

15.6.8.0.2.1 n-Degree: Transition

Motion Path - Line-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing linear motion. **Red**: Transition arc. 1. Original endpoint of the 1st move. 2. Corner distance.	

15.6.8.0.3.2 Special Case Transitions

These are the special cases for line-to-line transitions:

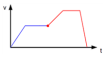
- "180-degree Transition" (→ p. 848)
- "Zero-degree Transition" (→ p. 849)
- "Zero-distance Transition" (→ p. 849)

15.6.9 180-degree Transition

Click to view 180-degree Transition

- The current move is shortened by the corner distance when the next move travels in the opposite direction.
- If the buffer mode is **Buffered**, the path velocity goes to zero.
  - All other buffer modes may cause a large acceleration and jerk.
- The new move is in the opposite direction as the old move.
- The sudden change in the direction of motion may result in large jerks to the axes.

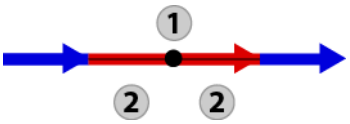
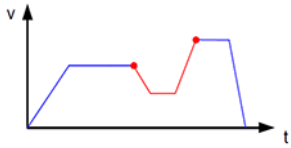
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

	**Blue**: Incoming linear motion. **Red**: Outgoing linear motion. 1. Original endpoint of the 1st move. 2. Corner distance.	

15.6.10 Zero-degree Transition

Click to view Zero-degree Transition

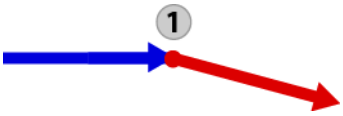
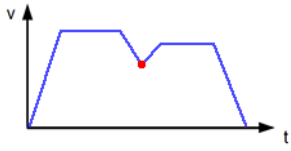
- The path remains unchanged.
- A linear transition move with the specified transition velocity is inserted.
- The new move continues in same direction as old move - continuous behavior.

Motion Path - Line-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing linear motion. **Red**: Transition line segment. 1. Original endpoint of the 1st move. 2. Corner distance.	

15.6.11 Zero-distance Transition

Click to view Zero-distance Transition

- A zero distance transition move is inserted.
  - This only affects blending.
- If the buffer mode is **Buffered**, the path velocity goes to zero.
  - All other buffer modes may cause a large acceleration and jerk.
- Motion passes the first move's endpoint.
- There is no transition arc.
- A sudden change in the direction of motion may result in large jerks to the axes.

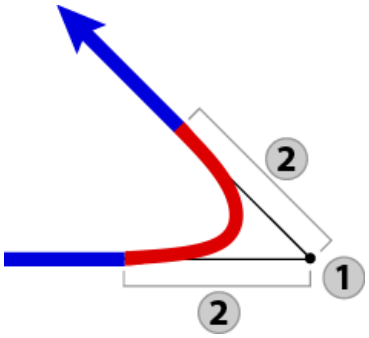
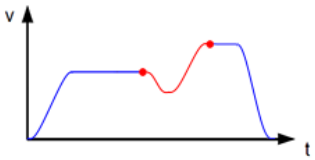
Motion Path - Line-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing linear motion. **Red**: Zero distance transition move. 1. Original endpoint of the 1st move.	

15.6.11.0.1 S-Curve Corner Line-to-Line Transitions

This topic describes the behavior when both moves are linear the current and next moves are shortened using the S-Curve Corner transition mode.

- The amount each linear move is shortened is specified by the corner distance.
- A spline with continuous acceleration is used to connect the two linear moves.
  - This allows for a smooth transition.
  - See "n-Degree: Transition" (→ p. 849).

15.6.11.0.2.1 n-Degree: Transition

Motion Path - Line-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing linear motion. **Red**: Transition arc. 1. Original endpoint of the 1st move. 2. Corner distance.	

15.6.11.0.3.2 Special Case Transitions

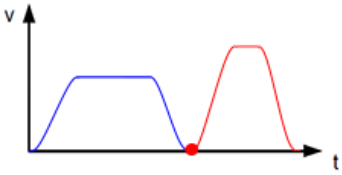
These are the special cases for line-to-line transitions:

- "180-degree Transition" (→ p. 850)
- "Zero-degree Transition" (→ p. 850)
- "Zero-distance Transition" (→ p. 851)

15.6.12 180-degree Transition

Click to view 180-degree Transition

- The current move is shortened by the corner distance when the next move travels in the opposite direction.
- The velocity drops to zero between the two moves.
- The new move is in the opposite direction as the old move.
- The velocity drops to zero to avoid large accelerations and jerks to the axes.

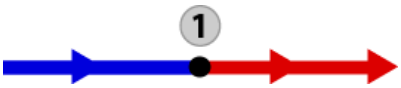
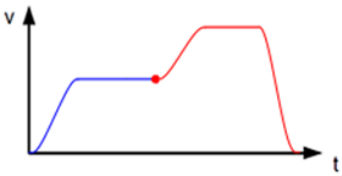
Motion Path - Line-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming linear motion. **Red**: Outgoing linear motion. 1. Original endpoint of the 1st move. 2. Corner distance.	

15.6.13 Zero-degree Transition

Click to view Zero-degree Transition

- The path remains unchanged.
- The corner distance does not affect the path or velocity profile.
- A transition element is not inserted.
- The velocity blending acts like BM_BLENDING_PREVIOUS is being used.
  - See "Blending Between Moves" (→ p. 822).
- The new move continues in same direction as old move - continuous behavior.
- No transition segment is inserted.

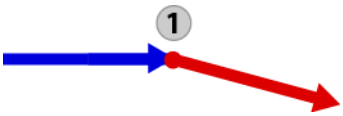
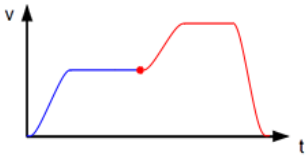
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Motion Path - Line-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming linear motion. **Red**: Outgoing linear motion. 1. Original endpoint of the 1st move.	

15.6.14 Zero-distance Transition

Click to view Zero-distance Transition

- The path remains unchanged and a transition element is not inserted.
- The velocity blending acts like BM_BLENDING_PREVIOUS is being used.
  - See "Blending Between Moves" (→ p. 822).
- Motion passes the first move's endpoint.
- No transition segment is inserted.
- A sudden change in the direction of motion may result in large jerks to the axes.

Motion Path - Line-to-Line	Key	Motion Velocity Profile
	**Blue**: Incoming and outgoing linear motion. **Red**: Zero distance transition move. 1. Original endpoint of the 1st move.	

15.7 Motion Techniques

This section explains advanced concepts and procedures related to **motion techniques** that are possible with the KAS-IDE.

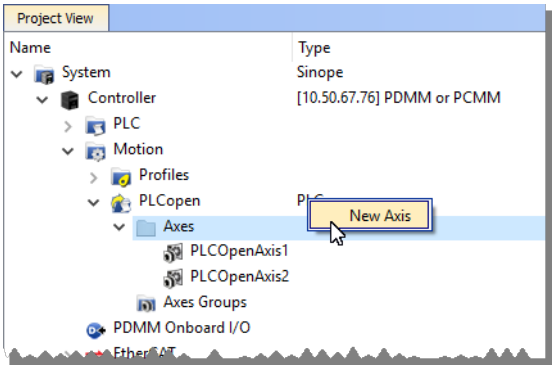
15.7.1 Create a Digitizing Axis

Digitizing axes are supported by PLCopen-based projects.

NOTE

This procedure is the general process.
The specifics will vary depending upon the EtherCAT network.

1. From the Project View, right-click the **Controller > Motion > PLCopen** item and select **New Axis**.



2. Give the axis a name.
   Spaces are not supported.
   See "Common Axis Parameters" (→ p. 199).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

3. Define the data in the **EtherCAT Bus** and **Feedback Settings** areas.
   See "Digitizing Axis Parameters" (➔ p. 200).
4. Define the "Axis Data Parameters" (➔ p. 202).

What happens when the device providing feedback to the digitizing axis is removed?

The feedback position becomes zero if the device providing the Feedback Position for the Digitizing axis is removed using the ECATDeviceAction function block.

The feedback position data coming from the device is automatically connected to the digitizing axis when the device is reconnected to the EtherCAT network.

See Also

- "Create a Digitizing Axis From Code" (➔ p. 852)
- "Create a Digitizing Axis with an Imported ENI File" (➔ p. 852)
- "Create a Digitizing Axis When EtherCAT Devices Are Not Connected" (➔ p. 854)
- "Create a Digitizing Axis When EtherCAT Devices Are Defined" (➔ p. 853)
- "Digitizing Axis - PLCopen Setup" (➔ p. 855) about digitizing axes.

15.7.1.1 Create a Digitizing Axis From Code

This procedure creates digitizing axes without using the GUI.

1. The MC_CreatePLCAxis function block is used to create the axis.
2. The MC_InitAxisFeedback function is used to initialize the feedback.

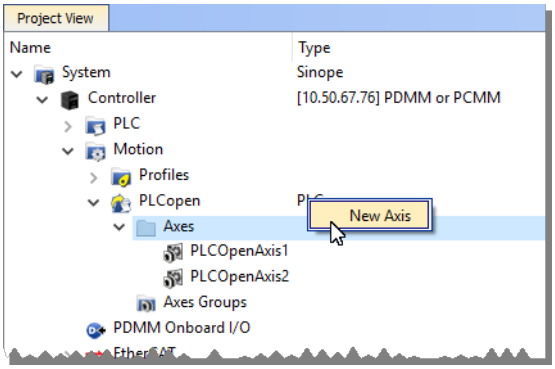
```
(* ----- Create PLCopen axis 'Digit1' *)
MC_CreatePLCAxis( 'Digit1', 'EtherCATDriver', 1001, 3, 1, 1, 1, 1, LREAL#0, 4
);
MC_InitAxisFeedback( 3, 24804, 2, 32 );
```

15.7.1.2 Create a Digitizing Axis with an Imported ENI File

This procedure assumes you are setting up a project when using an imported ENI file.

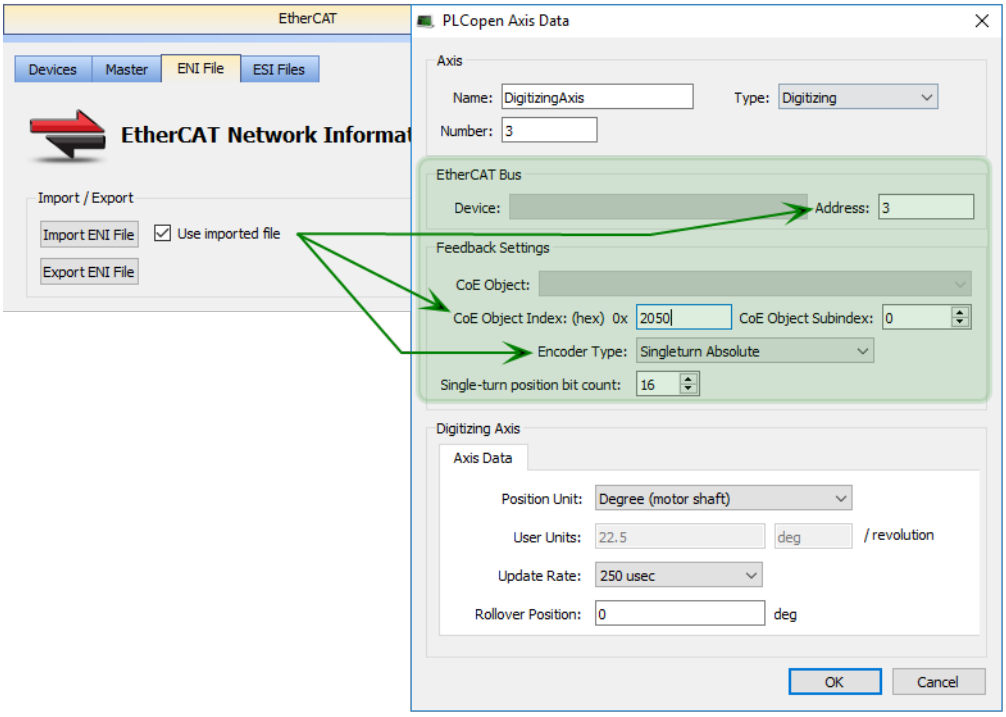
See "ENI File tab" (➔ p. 541) about using ENI files.

1. From the Project View, right-click the **Controller > Motion > PLCopen** item and select **New Axis**.



2. Give the axis a name.
   Spaces are not supported.
   See "Common Axis Parameters" (➔ p. 199).
3. Click the **Type** drop-down and select **Digitizing**.
4. Define the data in the **EtherCAT Bus** and **Feedback Settings** areas.
   The ENI file provides some information about the digitizing axis.
   The **Address**, **CoE Object Index**, **Subindex**, and the **Encoder Resolution** of the device to use are required.
   See "Digitizing Axis Parameters" (➔ p. 200).

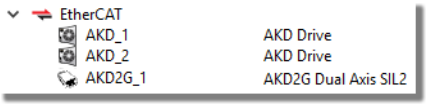
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



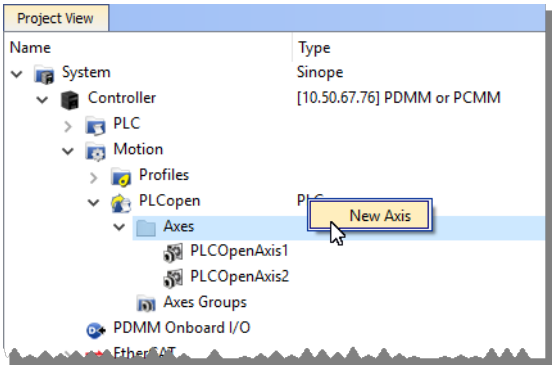
5. Define the "Axis Data Parameters" (→ p. 202).

15.7.1.3 Create a Digitizing Axis When EtherCAT Devices Are Defined

This process assumes that you have already scanned the EtherCAT network and assigned devices to axes.

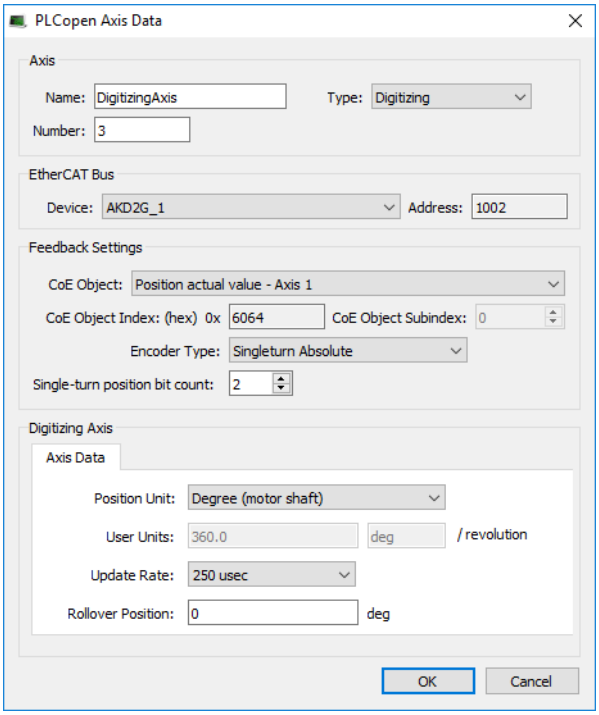


1. From the Project Tree, right-click on the *Controller* > *Motion* > *PLCopen* item and select **New Axis**.



2. Give the axis a name and click **OK**. Note that spaces are not supported and select "**Digitizing**" as the **Type**. See "Common Axis Parameters" (→ p. 199) for details.
3. Digitizing axes have *Bus* and *Feedback Settings* data which needs to be defined. Double-click on the newly defined axis. Using an AKD or AKD2G will automatically set certain parameters, and hide that data. For example the Encoder Resolution for AKD drives is automatically set to 20-bit. See "Digitizing Axis Parameters" (→ p. 200).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

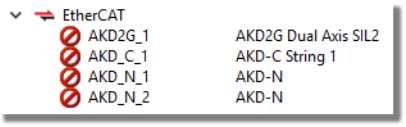


4. Define the "Axis Data Parameters" (→ p. 202).

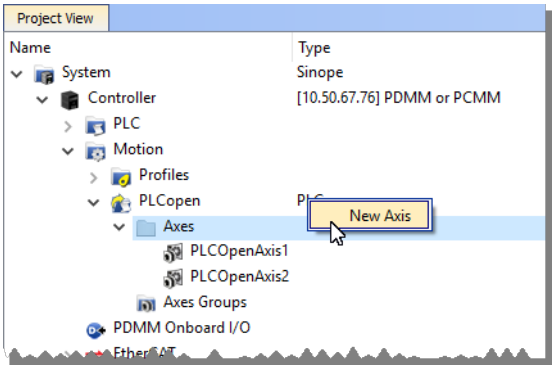
15.7.1.4 Create a Digitizing Axis When EtherCAT Devices Are Not Connected

This process assumes you are setting up a project and are not connected to the EtherCAT network.

In this example, several devices have been added to the EtherCAT branch of the Project View, including an AKD2G, an AKD-C, and two AKD-Ns.

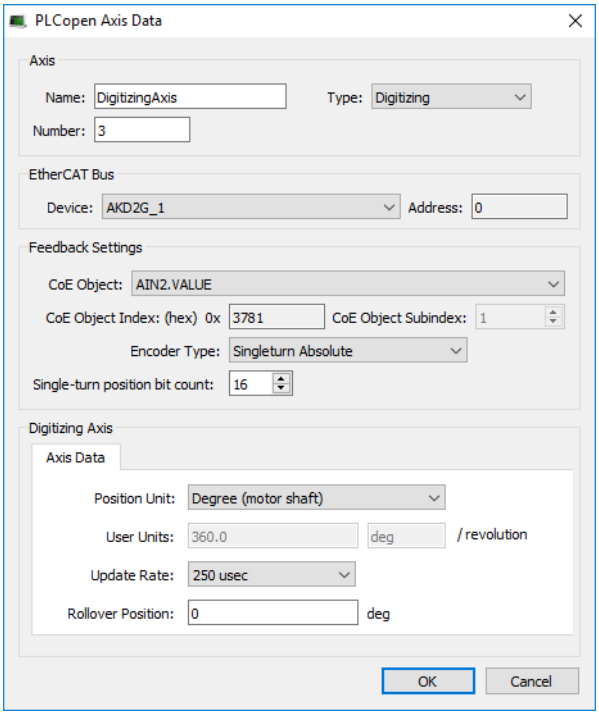


1. From the Project Tree, right-click on the *Controller* > *Motion* > *PLCopen* item and select **New Axis**.



2. Give the axis a name and click **OK**. Note that spaces are not supported and select "**Digitizing**" as the **Type**. See "Common Axis Parameters" (→ p. 199) for details.
3. Digitizing axes have *Bus* and *Feedback Settings* data which needs to be defined. Double-click on the newly defined axis and select the **Device** which will provide the feedback information and the feedback parameters, including the CoE information that will be used. See "Digitizing Axis Parameters" (→ p. 200) for details.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



4. Define the "Axis Data Parameters" (→ p. 202).

15.7.1.5 Digitizing Axis - PLCopen Setup

This topic provides the basics of setting up a digitizing axis and describes how to accomplish some common tasks.

See "Common Axis Parameters" (→ p. 199).

- A Digitizing Axis requires a feedback position.
  - The feedback position can come from any secondary feedback on the drive or an EtherCAT device.
  - Typically the AKD, S300, or S700 drives provide the feedback position via the CoE object **0x2050**. (Figure 19-5)

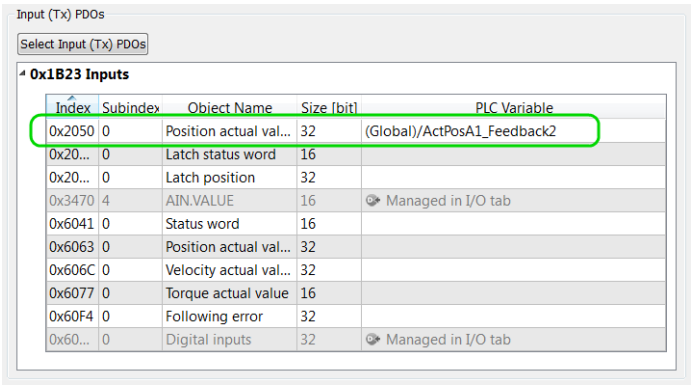


Figure 19-5: Example: 0x2050

- The AKD2G drive provides the feedback position via the CoE object 0x60E4. (Figure 19-6)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

0x1A03 Inputs

Index	Subindex	Object Name	Size [bit]	PLC Variable
0x60E4	3	3rd additional position actual value - Axis 1	32	(Global)/ActPos_Feedback3
0x60E4	4	4th additional position actual value - Axis 1	32	
0x60E4	5	5th additional position actual value - Axis 1	32	

Figure 19-6: AKD2G CoE Object 0x60E4

- The Digitizing Axis is created by:
  1. Setting the **Type** to **Digitizing**.
  2. Selecting the EtherCAT device to provide the feedback
  3. Setting the correct CoE object. (Figure 19-7)

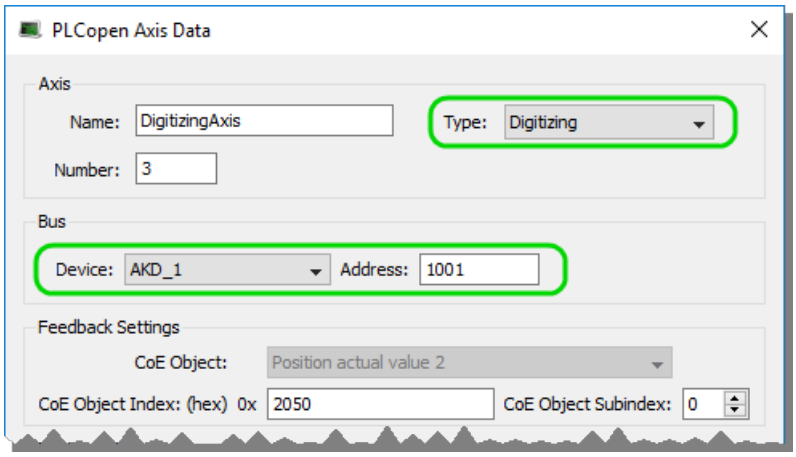


Figure 19-7: Create Digitize Axis Selections

15.7.1.5.1 First Generation AKD Drives

First generation AKD drives use certain pre-set values.

The actual position in user units is read with the MC_ReadActPos function block. (Figure 19-8)

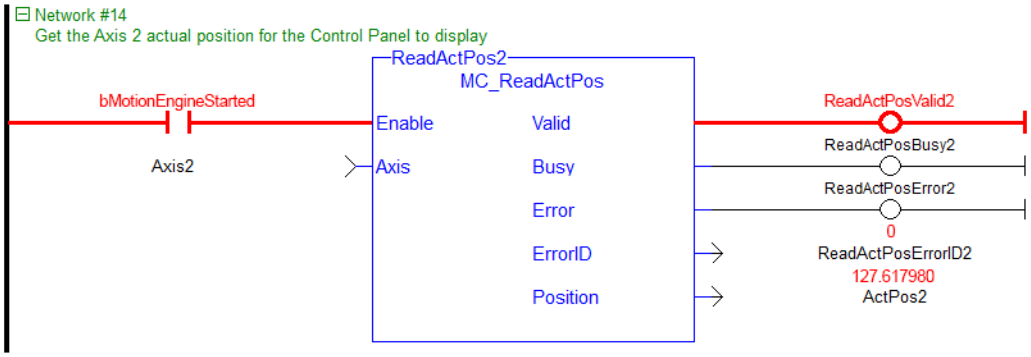


Figure 19-8: MC_ReadActPos Function Block

The position is offset using the MC_SetPos function block. (Figure 19-9)

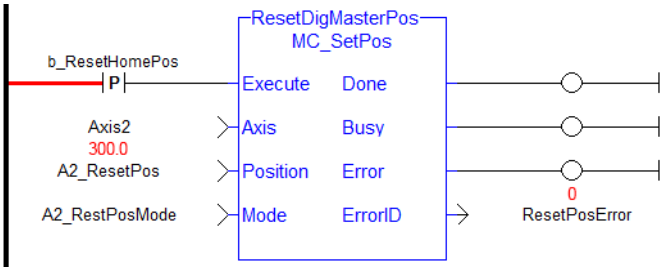


Figure 19-9: MC_SetPos Function Block

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

For AKD drives, the polarity of the feedback signal is changed using drive parameter FB2.DIR. (Figure 19-10)

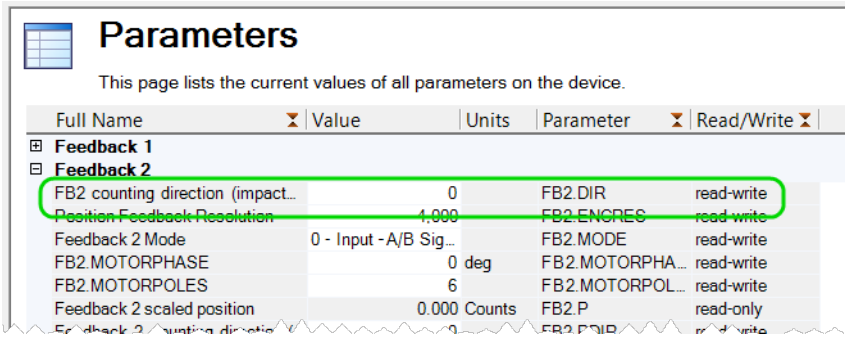


Figure 19-10: Drive Parameter FB2.DIR

NOTE

If an Endat 2.2 device is used, set FB3.DIR and change DRV.HANDWHEELSRC to **3** to tie the PDO index **2050** parameter read from FB2 to FB3.

What happens when the device providing feedback to the digitizing axis is removed?

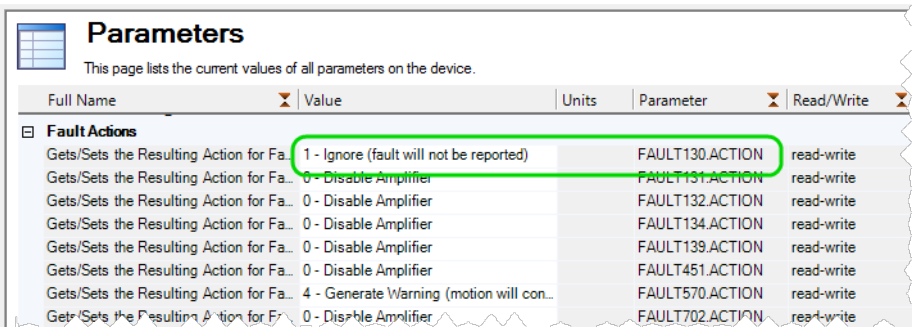
The feedback position becomes zero if the device providing the Feedback Position for the Digitizing axis is removed using the ECATDeviceAction function block.

The feedback position data coming from the device is automatically connected to the digitizing axis when the device is reconnected to the EtherCAT network.

15.7.1.6 Digitizing Axis - Encoders without a Z-Channel

If the encoder does not have a Z-channel, there may be some unrelated drive faults associated with the secondary feedback signal.

To disable this, change the setting for FAULT130.ACTION (or fault 131, 132, etc.) in the AKD parameters to 1.



15.7.2 Error Management

When a non-fatal error occurs and motion must be stopped quickly these procedure can be taken:

15.7.2.1 Axis Procedures

Step	Example Application Code
Send Stop Command for each axis.	MLAxisStop `(PipeNetwork.AXI_A1_Axis, TRUE, DEF_A1_StopDec);`

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Step	Example Application Code
Stop the Axis Motion Generator.	MLAxisMoveVel `(PipeNetwork.AXI_A1_Axis, 0.0);`

Wait for Axis to be stopped.	AxisStatus := MLAxisStatus
	`(PipeNetwork.AXI_A1_Axis);`
	`IF AxisStatus.11 THEN`
	`MLAxisStop`
	`(PipeNetwork.AXI_A1_Axis,FALSE,DEF_A1_`
	`StopDec);`

Turn power off(disable) all the axes.	MLAxisPower (PipeNetwork.AXI_A1_Axis,FALSE);
Disconnect Pipe Network from the axis.	MLCNVDisconnect (PipeNetwork.CNV_A1);

15.7.2.2 Machine Procedures

Step	Example Application Code
Stop Command at the master block level.	MLMstRun `(PipeNetwork.MASTER, 0.0);`

Wait for Master command to be stopped.	`IF A1_AckState = DEF_StateErrorStop`
	`AND A2_Ackstate = DEF_StateErrorStop`
	`AND (PipeNetwork.MASTER)`
	`THEN`
	`PrintF('*** ErrorStop M1=%i ***', M1_`
	`StatusWord,0,0,0);`
	`M1_AckState := DEF_StateErrorStop;`

IMPORTANT

This code is only applicable to Pipe Network applications.
It does not work in a PLCopen application.

This procedure for error management is based on the **Project Structure Guidelines** described in Application Software Structure - Implementation (→ p. 1011).

See Restarting Motion with Pipe Network (→ p. 884) about restarting the motion.

15.7.3 Fast Inputs

Fast (or high-speed) Inputs are digital inputs of a drive that are configured to latch the time at which they are triggered.

- Fast Inputs allow a high-speed application to get position information about the occurrence of an external event at a higher resolution than the cycle time.
- Because of the precise timing of external events, an application can improve its control algorithm, resulting in higher operating performance.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- A position or time capture can be triggered either by the positive (rising) edge or by the negative (falling) edge of the digital input.
  - The "AKD Drive Configuration" (➔ p. 448) has two capture engines which can be freely linked to any input.
  - These high speed inputs are used in applications which, when triggered, cause a drive position to be captured and reported back to the controller.

NOTE

Only digital inputs 1 and 2 can be used as Fast Inputs.

IMPORTANT

When using S300 or S700, Fast Inputs must be enabled by setting the drive keywords IN1MODE to 26 and IN2MODE to 26.
This can be achieve using DriveGUI configuration tool.

15.7.3.1 Fast Inputs with Pipe Network Motion

TIP

For the Axis and Trigger Pipe Network blocks, see MLAxisCfgFastIn.
This function arms a Fast Input and returns the latched timestamp when the Fast Input event occurs.

Registration Operation Related FB (Trigger FB)	Homing Operation Related FB (Axis FB)
MLTrigReadPos	MLAxisCfgFastIn
MLTrigIsTrigged	MLAxisIsTrigged
MLTrigSetEdge	MLAxisRstFastIn

When the input is triggered, the timestamp is latched.

- With EtherCAT, the timestamp sent to the KAS-IDE via the MLAxisTimeStamp or MLTrigReadTime function blocks is based on the distributed clock that manages the reference clock.
  - See "EtherCAT Distributed Clock (Synchronization)" (➔ p. 418).
- The KAS-IDE converts this timestamp into a relative offset inside the cycle.
- It is also possible to configure a Fast Input to latch the motor position instead of latching the time.
  - See "AKD Drive Configuration" (➔ p. 448).
- However, when working with KAS, time latching is more useful, because the positions of all the drives in the application can then be interpolated using the Trigger block with the MLTrigReadPos function block.

15.7.3.2 Fast Inputs with PLCopen Motion

For PLCopen, see MC_TouchProbe.

This function block arms a Fast Input and returns the latched position when the Fast Input event occurs.

Other PLCopen function blocks include:

Registration Operation	Homing Operation
MC_MachRegist	MC_Reference
MC_MarkRegist	MCFB_StepAbsSwitchFastInput
	MCFB_StepLimitSwitchFastInput

15.7.4 Interpret a Timestamp

The timestamp is based on the EtherCAT system time.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

For this value to make sense, the distributed clock must be activated in the drive (see "Distributed Clock tab" (→ p. 568)) and in the EtherCAT master.

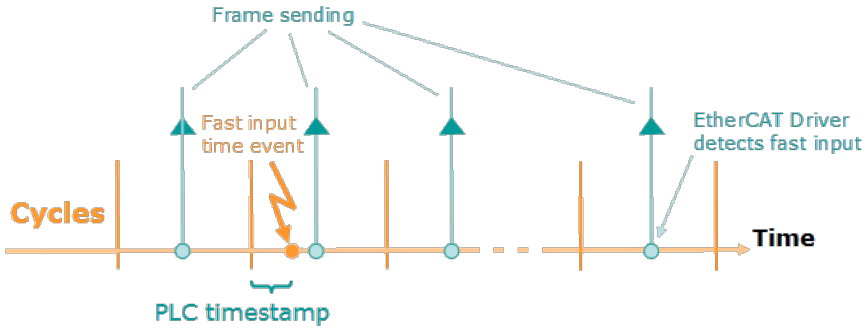


Figure 19-11: PLC Timestamp Related to Fast Input Event

- The timestamp returned is relative to the beginning of the cycle in which the Fast Input is triggered.
- It is called PLC timestamp because it is the value used in PLC programs.

15.7.5 Overtravel Conditions

The PLCopen and Pipe Network motion engines can respond to overtravel conditions from the Kollmorgen AKD Series Servo Drives.

- As a safety feature, Kollmorgen AKD Series Servo Drives ignore motion commands once an overtravel is detected in the direction of a hardware limit switch.
- The PLCopen or Pipe Network motion engine applies a stop action to axes when these overtravel conditions occur.
- While DS-402 Status word bit #11 (Internal Limit Active) is on, the controller follows the actual position of an axis.
    - This prevents a sudden position jump when the drive resumes following the controller demand position.

Notes

- Generation 1 AKD drives require setting a fieldbus parameter to enable this behavior.
    - See "Configuring Generation 1 AKD Drives" (→ p. 860).
- The default behavior of the motion engine can be modified by setting these Axis Boolean parameters:
    - Pipe Network = ML_AXIS_PARAM_IGNORE_DRIVE_LIM_STATUS
    - PLCopen = MC_AXIS_PARAM_IGNORE_DRIVE_LIM_STATUS
- When this parameter is set to TRUE, the motion engine:
    - Ignores the DS-402 Status word bit #11 from the drive.
    - Allows the application to take the appropriate action.
- By default, the motion engines attempt to stop axes connected to drives that are not part of the AKD or AKD2G family instead of the letting the drives do so.
    - If a drive overrides the controller demand input when DS-402 Status word bit #11 is active, these Axis Boolean parameters need to be set to TRUE:
        - Pipe Network = ML_AXIS_PARAM_DRIVE_OVERRIDE_ON_LIMIT_ACTIVE
        - PLCopen = MC_AXIS_PARAM_DRIVE_OVERRIDE_ON_LIMIT_ACTIVE

15.7.5.0.1 Configuring Generation 1 AKD Drives

When using generation 1 AKD drives, bit # 14 of the parameter FBUS.PARAM05 must be set high to allow KAS-IDE to respond to the hardware limit switches.

- See FBUS.PARAM1 TO FBUS.PARAM10.
- If FBUS.PARAM05 originally had a value of 0 (zero), typing `FBUS.PARAM05 16384` turns on bit # 14.
- Remember to save your drive parameters to flash memory once bit # 14 is set, otherwise the behavior turns off once the drive is power cycled.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.7.5.1 Coordinated Motion Behavior

An axis that is part of an axes group issues an emergency stop on the axes group when it experiences an overtravel condition.

- All axes remain enabled under servo control once the emergency stop action completes.
- This action puts the axes group into an error state.

To recover from this state:

1. Call MC_GrpReset on the axes group.
2. Check the states of the I/O points configured for the positive and negative limit switches on the drive associated with the axes to figure out which axis and which limit switch cause the error.
3. Command motion in the direction opposite to what was previously commanded.

15.7.5.2 Pipe Network Behavior

A Pipe Network axis issues a stop action on the axis when it experiences an overtravel condition.

- This is similar to calling MLAxisStop.
- bit #11 of MLAxisStatus is set and the axis remains enabled under servo control once the emergency stop action completes.
- The axis no longer accepts pipe positions.
- The pipe the axis is connected to continues functioning as before and any master will continue generating new positions.
- The axis which experienced overtravel simply no longer responds to new pipe positions.

To recover from this state, call MLAxisReAlign to realign the axis with the pipe.

See "Restarting Motion with Pipe Network" (➔ p. 884).

NOTE

If a hardware limit flag is being triggered at the program startup by a Pipe Network axis, MLAxisReAlign may need to be called immediately before motion occurs.
In this case, the distance to travel should be at least as long as the maximum distance the hardware limit flag can remain active.

15.7.5.3 PLCopen Behavior

A PLCopen axis issues an emergency stop of the axis when it experiences an overtravel condition.

- The axis remains enabled under servo control once the emergency stop action completes.
- This action puts the axis into the **ErrorStop** state.
- See "PLCopen State Machine" (➔ p. 383).

To recover from this state:

1. Call MC_ResetError on the axis.
2. Command motion in the direction opposite to what was previously commanded.

   If you are uncertain about which direction the axis was traveling in, check the states of the I/O points configured for the positive and negative limit switches on the drive.

15.7.6 PC Online Change

IMPORTANT

The project must be saved () and compiled () before doing an online change.

This section provides a detailed description of the PLC Online Change functionality.

See "Use the PLC Online Change" (➔ p. 867).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.7.6.1 What is Online Change

- Online Change enables you to update your PLC application on the fly, while it is running on the controller.
- You do not need to stop the controller, download the new code and start again.
- You only need to modify, recompile, and download the new code (e.g., Figure 19-12) and then ask the controller to switch the execution to the new application.

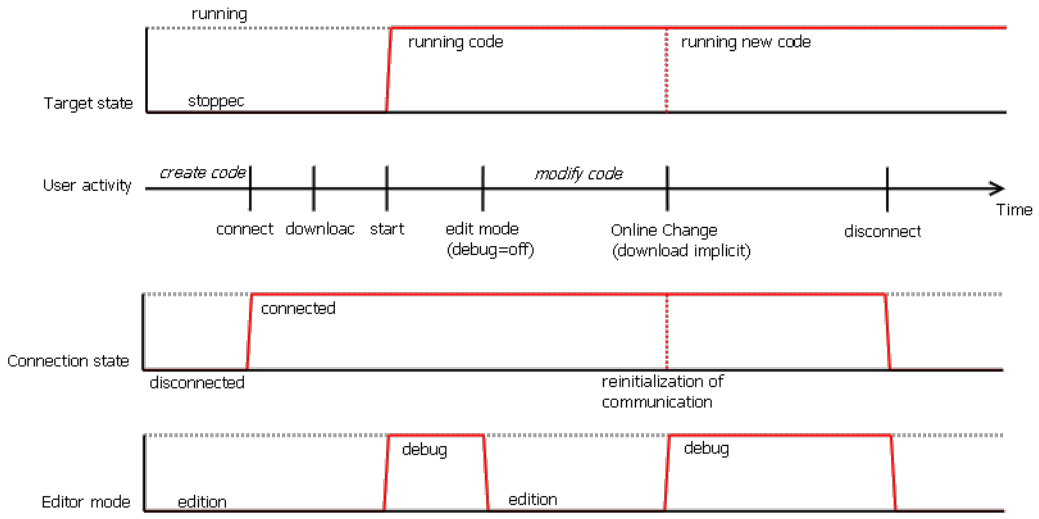


Figure 19-12: Online Change - Process Diagram

TIP

- This capability applies only to PLC code.
- This is not supported in the Pipe Network editor, the PLCopen axis, or any other part of the system.

NOTE

- Depending on the PLC code size, the time to perform the Online Change operation can take more than one cycle.
- In that case, you can miss one PLC cycle before the changeover becomes effective.
- This duration is also displayed in the Log window with this INFO level message: **Online Change done in X µs**.
- The INFO measurement corresponds to the duration for the code hotswap.
- The download and loading of new code in memory is not taken into account in this measurement because they occur when the previous code is still running.

This feature is used in these situations:

- **Development Phase**
  - Modify the application and apply these modifications incrementally without stopping the controller.
- **Update in production**
  - Update the running motion application (e.g., with a bug fix release) without stopping the whole production chain.

When Online Change is enabled, you can perform these kinds of changes on the fly:

- Rename a program.
- Change the code of a program.
- Change the condition of an SFC transition or the actions of an SFC step .
- Create, rename, or delete global and local variables.
- Create, rename, or delete global and local function block instances.
- Rename "Retain Variables" (➔ p. 292).

These changes **are not allowed**:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Create or delete a program.
- Add or remove Custom Libraries.
- Add new PLC axes, change or delete PLC axes.
- Add new cam profiles, change or delete existing cam profiles.
- Add new Pipe Network blocks, change or delete existing Pipe Network blocks.
- Add or remove variables in a Structure.
- Create a new Structure or a new UDFB.
- Change the local parameters and variables of a UDFB.
- Change the type or dimension (or string length) of a variable or function block instance.
- Create or delete Retain variables (their position in the runtime cannot be re-allocated).
- Change SFC charts: you cannot add or remove steps in the First Level of an SFC chart.
  - You can modify existing steps.
- Change the set of Input/Output or any modification that leads to an update in the EtherCAT Motion Bus configuration.
- Pulse (P or N) contacts and coils (edge detection).

  TIP

  Using Pulse contacts in FFLD does not give any error, but the behavior of the contact during the switch is not always safe.
  See click here... for a workaround.
- The WAIT / WAIT_TIME instructions must **not** be used.

  IMPORTANT

  The Online Change and Revert functions will fail while executing a WAIT.
- Loops in FBD with no declared variable linked.
  - In this case, you must explicitly insert a variable in the loop.
- The online change is denied by the KAS Runtime if some system resources are currently open when the change is performed.
  - A system resource can be either a file open by file management functions, or a socket open by TCP/UDP functions.
  - If an online change is performed while system resources are open, the online change operation fails with an error message indicating system resource used.

IMPORTANT

When Online Change is enabled and custom libraries are used, some errors can occur during the compilation.
This happens if you open your project on another PC or under a different user account in Windows.
To fix this limitation:
1. Deactivate the Online Change.
2. Save and then reopen the project.
3. Re-enable the Online Change.

NOTE

Your new application can contain more variables than the previous one.
A memory with sufficient pre-allocated space is defined for the eventual new variables.
If you exceed this limit, a warning message opens.

For limitation about breakpoint with Online Change, see "Online Change" (➔ p. 271).

Click to see the States and Transitions

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

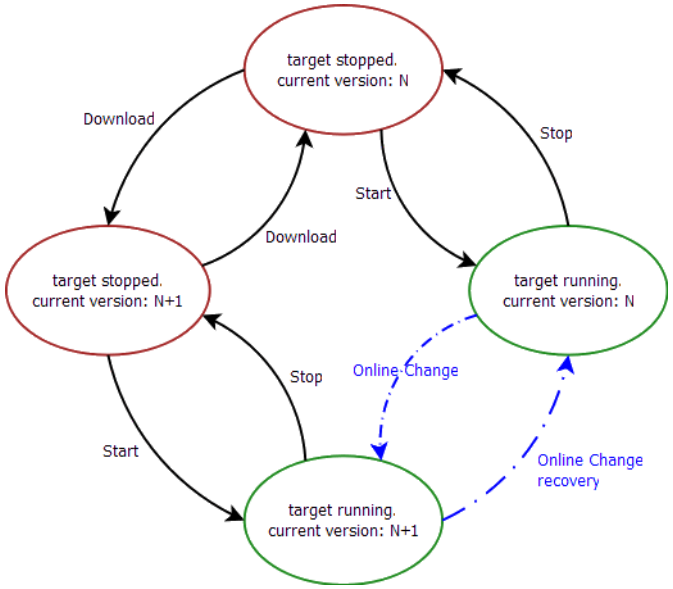


Figure 19-13: Online Change - States and Transitions

15.7.6.2 Activate an Online Change

To allow Online Change, open the Project Settings dialog box (Figure 19-14) and set the relevant parameters.

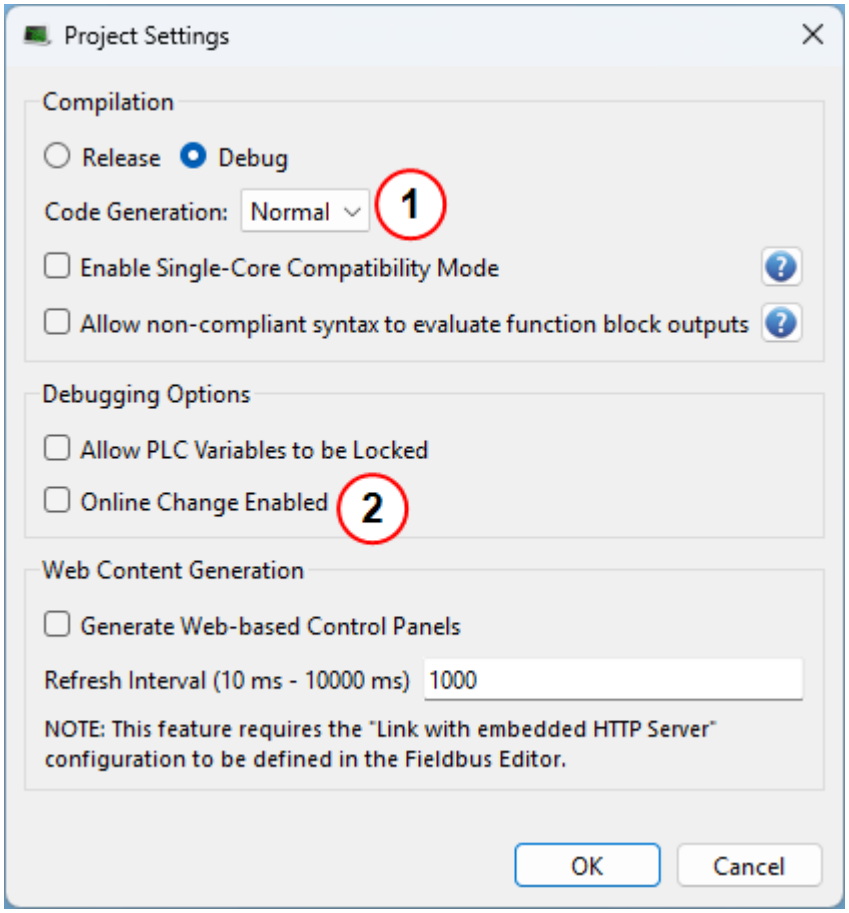


Figure 19-14: Project Settings dialog box - Online Change Enable

15.7.6.2.1 Set the Parameters

Verify the **Normal** code (1) is selected because Online Change is not possible with **Optimized** code.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

When **Optimized** code is selected, then Online Change is always deactivated.

Then you can compile the application; this now allows future changes on the fly.

This dialog box is used to enable or disable the **Online Change** feature (2).

IMPORTANT

If you deactivate the Online Change, the next PLC application generated is no more compatible for an online change, even if you re-activated the online change before the compilation.
As a result, you can only apply an Online Change to a running application under these conditions:

- The Online Change was already activated.
- You have never deactivated the Online Change between the compilation of the running application and the compilation of the new application.
  - Check the Controller Log window for any errors that occur.

15.7.6.2.2 Switch to Edit Mode

When you start the application, the Debug mode is automatically activated.

- You can see the values changing in the editors and the Dictionary , showing what is happening on the controller.
- In this mode, the editor is read-only, so you are not able to modify the code.

To edit your code, go out of the Debug mode and enter the Edit mode by clicking the ( ) button in the "Main (Tools) toolbar" (➔ p. 471).

Perform the Online Change

When your new code has compiled correctly, you can perform the Online Change. To do so, click the button. When you click this button, the KAS-IDE opens a window showing the execution of current actions (download, activation of new code).

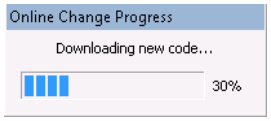


Figure 19-15: Online Change - Updating Controller Version

Once the Online Change is applied, the result is displayed in the window and you can click OK to acknowledge the operation and do a

Warm start.

15.7.6.2.3 Dictionary Behavior

When the Online Change is enabled, the dictionary shows:

- new variables in blue
- deleted variables in red

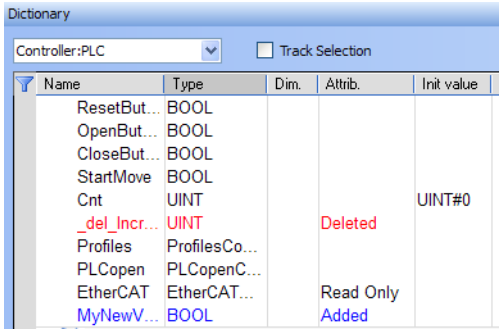


Figure 19-16: Online Change - Dictionary

TIP

The deleted variables can be used for new variables.

15.7.6.3 The Revert Button

The **Revert** button is for security purposes.

- It's used, after an Online Change, to revert the change quickly and return to the previous application.
  - That means switching the execution of the controller to the P-code that was running before the last Online Change.
  - The source code in the KAS-IDE is not replaced.
  - The WAIT / WAIT_TIME instructions can not be used with **Revert**.

After the Revert, the KAS-IDE automatically goes back to Edit mode and is disconnected from the controller.

The controller must be manually reconnected.

NOTE

You can go back to the previous version only when the Online Change feature is activated and while the controller is not stopped.

IMPORTANT

After a revert operation, the Online Change feature is **deactivated**.

The Revert button is active when you are connected and the controller is running.

Revert is not possible:

- If you did not perform an Online Change.
- If the controller has been restarted since the previous Online Change.
- After another Revert.
- During a WAIT.

15.7.6.4 Difference between Local and Controller Versions

When you restore a project with the Revert feature after an Online Change, KAS provides a tool to show the differences between two versions of the project.

This tool can help:

- In checking all modifications before the next Online Change.
- Compare the code with the last version after a Revert.

See "Compare PLC Programs" (➔ p. 277).

15.7.6.5 Pulse Limitations with Online Change

At the first cycle, the pulse evaluation is ignored, and the memory is updated.

This memory enables the pulse evaluation from the second cycle.

When we apply the Online Change between t0 ant t1, the cases where this method is not correct are the two following:

Falling Edge

When we want to detect a **falling** edge:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

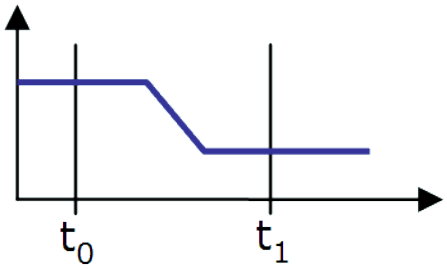


Figure 19-17: Pulse Limitations with Falling Edge

Rising Edge

When we want to detect a **rising** edge:

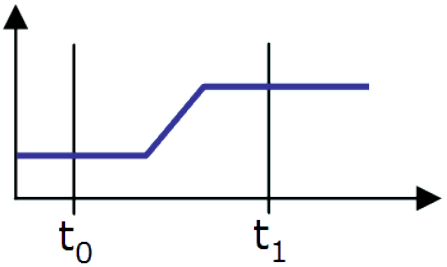


Figure 19-18: Pulse Limitations with Rising Edge

TIP

If you want to avoid this limitation, you must use declared instances of R_TRIG and F_TRIG function blocks.

15.7.6.6 Use the PLC Online Change

See "PC Online Change" (→ p. 861) for descriptions of the functionality.

15.7.6.6.1 Set an Application

1. Create a new PLC application.
2. Connect to a controller and scan for EtherCAT devices.
3. Add logic and function blocks to the application.
4. On the toolbar, click the  button to compile the project.

15.7.6.6.2 Enable Online Change

1. On the toolbar, click the  button.
   The Project Settings dialog box opens. (Figure 19-19)

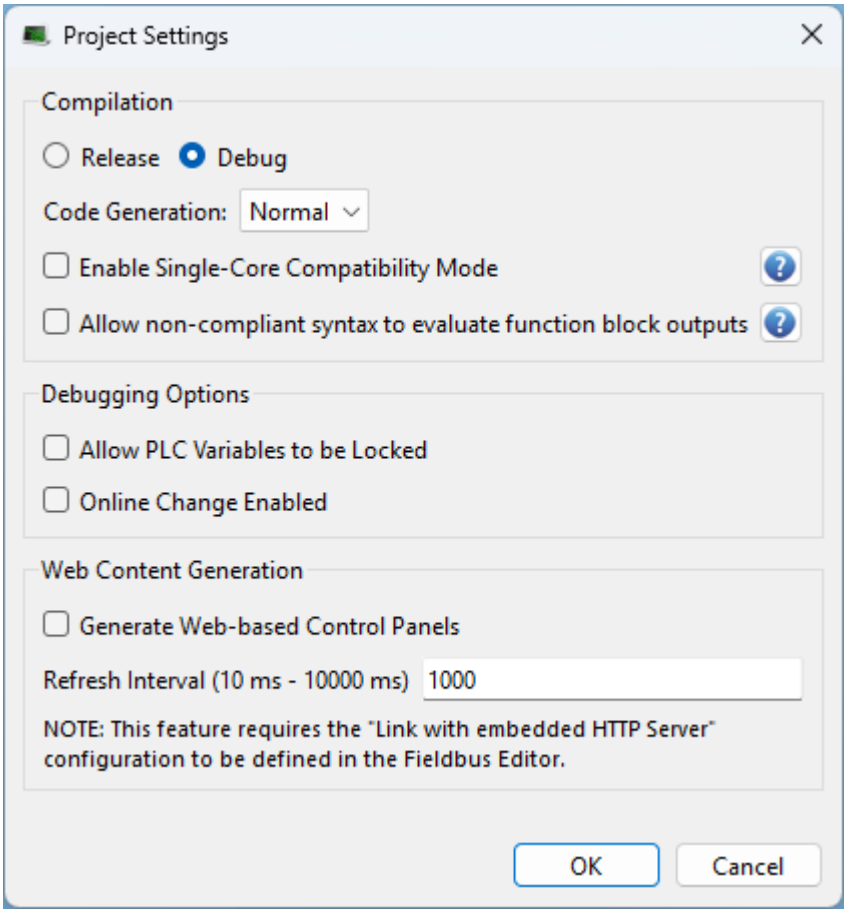


Figure 19-19: Project Settings dialog box

2. Select the **Online Line Change Enabled** check box to update the PLC application while it is running on the controller.
3. Click **OK** to save the changes or selections and close the dialog box.
4. Compile the project .
5. Connect and download the application to the device.
6. Start executing the application.

15.7.6.6.3 Use the Online Change

1. Enable the Toggle Edit/Debug mode button in the tool bar.
The application variables are not updated even though the application is running. (Figure 19-20)

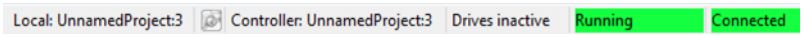


Figure 19-20: Online change, running and connected

2. Modify a local or global variable name in the Dictionary. (Figure 19-21)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

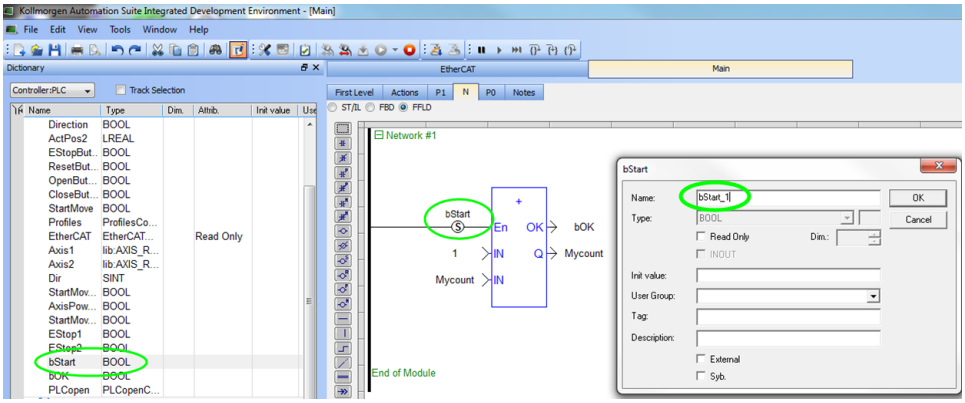


Figure 19-21: Change the name in the Dictionary

3. Compile the project .
4. Click the PLC Online Change download button on the tool bar to download the changes to the controller.
   The status bar has changed from Running to Paused during the download and back to Running once the download has finished.
   The applications variables are updating.

NOTE

If Online Change is disabled while the application is running, even temporarily, downloading will fail.

15.7.6.6.4 Revert the Online Change

1. Click the PLC Online Change Revert button on the tool bar.
   A message appears stating that revert was successful.
2. Note the reverted state of the application in the KAS-IDE. (Figure 19-22)

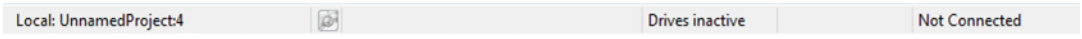


Figure 19-22: Reverted state

3. Connect to the device.
4. Note the state of the application and the device. (Figure 19-23)

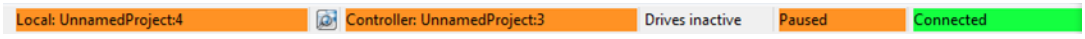


Figure 19-23: Application state

5. From this state, Online Change can only be used again after:
   a. Stopping the application.
   b. Making changes.
   c. Recompile.
   d. Download the application to the controller.

15.7.7 PLCopen Homing

15.7.7.1 PLCopen Homing Description

The homing features provided in PLCopen create tools for homing of PLCopen axes.

Homing can be performed using:

- The MC_Reference function block.
- The Custom Homing Library UDFBs.
- Your own written homing cycles.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.7.7.1.1 Utilizing MC_Reference

- The application specifies a position for an axis to be assigned to a reference position.
  - It then invokes the MC_Reference function block to generate motion to move the axis to the reference location.
- The AKD capture engine (previously set up by the application via SDO commands) captures the position of the reference location.
- Based on the reference position and the captured actual position, the coordinate system is shifted to correlate the desired reference position to this location.

15.7.7.1.2 Writing your Own Homing Cycles

- UDFBs can be written to provide specific canned homing cycles based on feedback type and homing sequences (e.g., homing off of limit switches, encoder markers, homing to zero or null positions etc.) by proper configuration of the AKD capture engine, the MC_Reference and MC_SetpositionMC_SetPos function blocks.

15.7.7.1.3 Utilizing Custom Libraries

- A library already contains a set of homing UDFB.
- To add the library to your project, see "Create and Use Custom Libraries" (➔ p. 244).
- Contact "Support and Services" (➔ p. 1073).

15.7.7.2 PLCopen Homing Methods

The common homing methods (among others) can be performed in PLCopen.

- This section details the setting of the AKD parameters and the PLCopen function blocks to accomplish these methods.
- PLCopen is not limited to these methods; the capture engine is very configurable.

15.7.7.2.1 Home using the Current Position

- Homing using the current position is simply accomplished using the MC_SetPositionMC_SetPos function block.
- Using this function block, the current position can be set to any value.

15.7.7.2.2 Find Input

Homing using a drive input is accomplished by configuring the AKD capture engine, and then using the MC_Reference function block.

These capture engine parameters need to be configured, along with the input parameters in the MC_Reference.

- Capture Event has to be set to ignore preconditions (0).
- Capture edge – capture edge is programmed in the MC_Referece block.
- Capture Trigger must be set to the desired drive input (0-6).
- Capture mode must be set to capture position (0).
- Capture preselect is not used.
- Capture Precondition edge is not used.
- MC_Reference inputs:
  - Trigger_Ref.InputID must be set to 0 or 1 to select which AKD capture engine to use.
  - Trigger_Ref.Direction must be set to Rising (1) or Falling (2) to select Capture Edge.
  - Trigger_Ref.Trigid is not required.
  - Position input must be programmed to the desired position at the switch.
  - Option input must be programmed to 0 for use latched position.

15.7.7.2.3 Find Input then Find Zero Angle

Homing using a drive input along with the zero angle is similar to Find Input except the position is defined at the zero angle of the feedback device, rather than the switch location.

It is typically used for resolver feedback.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Capture Event must be set to ignore preconditions (0).
- Capture edge – capture edge is programmed in the MC_Referece block.
- Capture Trigger must be set to the desired drive input (0-6).
- Capture preselect is not used.
- Capture Precondition edge is not used.
- Capture mode must be set to capture position (0).
- MC_Reference inputs:
  - Trigger_Ref.InputID must be set to 0 or 1 to select which AKD capture engine to use.
  - Trigger_Ref.Direction must be set to Rising (1) or Falling (2) to select switch capture edge.
  - Trigger_Ref.Trigid is not required.
  - Position input must be programmed to the desired position at the null closest to the switch.
  - Option input must be programmed to identify the number of poles the resolver has.

15.7.7.2.4 Find Input then Find Index

Homing using a drive input along with the index is similar to Find Input except the position is defined at the index pulse of the feedback device, rather than the switch location.

- It is typically used for incremental encoder feedback.
- To accomplish this, a precondition is used in the capture engine.
  - Specifically, the input is the precondition, and the index is the event.
  - The reference method looks for the switch first, and then the index pulse.
- Capture Event must be set to the desired switch operation.
  - Typically set to 1 to require the edge of the switch. Set to 2 or 3 if the state of the switch is required.
- Capture preselect must be set to the desired drive input (0-7).
- Capture edge – capture edge of index pulse is programmed in the MC_Referece block.
- Capture Trigger must be set to the desired index input (10 = primary index,11 = tertiary index).
- Capture mode must be set to capture position (0).
- MC_Reference inputs:
  - Trigger_Ref.InputID must be set to 0 or 1 to select which AKD capture engine to use.
  - Trigger_Ref.Direction must be set to Rising (1) or Falling (2) to select Capture Edge.
  - Trigger_Ref.Trigid is not required.
  - Position input must be programmed to the desired position at the index pulse.
  - Option input must be programmed to 0 for use latched position.

15.7.7.2.5 Find Index

Homing using a drive index pulse is accomplished by configuring the AKD capture engine, and then using the MC_Reference function block.

These capture engine parameters need to be configured, along with the input parameters in the MC_Reference.

- Capture Event must be set to ignore preconditions (0).
- Capture edge – capture edge is programmed in the MC_Referece block.
- Capture Trigger must be set to the desired index input (10 = primary index,11 = tertiary index).
- Capture mode must be set to capture position (0).
- Capture preselect is not used.
- Capture Precondition edge is not used.
- MC_Reference inputs:
  - Trigger_Ref.InputID must be set to 0 or 1 to select which AKD capture engine to use.
  - Trigger_Ref.Direction must be set to Rising (1) or Falling (2) to select Capture Edge.
  - Trigger_Ref.Trigid is not required.
  - Position input must be programmed to the desired position at the index pulse.
  - Option input must be programmed to 0 for use latched position.

15.7.7.3 AKD Capture Engine Configuration

The AKD capture engine provides a broad range of capabilities for configuration of the capture events.

- It is capable of configuring preconditions to allow the application programmer to specify sequential events or conditions that must be met before the capture event can be triggered.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- The capture Engine in the AKD is configured with SDO #0x3460 (subindexes 1 to 10).
- The AKD supports two capture engines (0 and 1); the application programmer must configure the desired engine.

Sub Index #	Function
1	Trigger for capture engine 0.
2	Trigger for capture engine 1.
3	Mode for capture engine 0.
4	Mode for capture engine 1.
5	Capture Event for capture engine 0.
6	Capture Event for capture engine 1.
7	Precondition edge for capture engine 0.
8	Precondition edge for capture engine 1.
9	Preselect for capture engine 0.
10	Preselect for capture engine 1.

15.7.7.3.1 AKD Configuration Parameters

These are the configuration parameters for the AKD capture engines.

See either CAP0.PRESELECT or CAP1.PRESELECT.

Parameter	Description
Capture Edge	Programmed in the MC_Reference function block.
Capture Event SDO object #0x3460 subindex engine 0 = 5/engine 1 = 6	• 0 = Ignore preconditions. • 1 = Trigger edge after the precondition edge. • 2 = Trigger edge while precondition = 1. • 3 = Trigger edge while precondition = 0.
Capture Mode SDO object #0x3460 subindex 3/4	• 0 = Capture position. • 1 = Capture internal time. • 2 = Capture EtherCAT distributed time (DCT). • 3 = Capture zero angle position.
Capture Precondition Edge SDO object #0x3460, subindex 7/8	• 0 = Reserved. • 1 = Precondition with rising edge. • 2 = Precondition with falling edge. • 3 = Precondition with rising and falling edges.

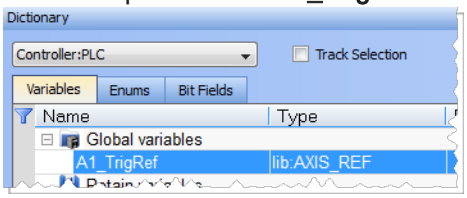
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Parameter	Description
Capture Preselect SDO object #0x3460 subindex 9/10	• 0 = General input 1. • 1 = General input 2. • … • 6 = General input 7. • 7 = rs485 input 1. • 8 = rs485 input 2. • 9 = rs485 input 3. • 10 = Primary index. • 11 = Tertiary index.
Capture Trigger SDO object #0x3460 subindex 1/2	• 0 = General input 1. • 1 = General input 2. • … • 6 = General input 7. • 7 = rs485 input 1. • 8 = rs485 input 2. • 9 = rs485 input 3. • 10 = Primary index. • 11 = Tertiary index.

15.7.7.4 Example: Fast Homing with the PLCopen Motion Engine

This example shows how to use the axes' motion library functions to detect the positive edge of the first Fast Input in the drive, and read its position. MC_TouchProbe and the Trig_Ref library function are used to accurately perform a homing procedure.

1. Set up an instance of the TrigRef function in the Dictionary.
   This example used the **A1_TrigRef** name.



2. Define the TrigRef values.

```
A1_TrigRef.InputID := 0;     // configure Drive Capture Engine Used
A1_TrigRef.TrigID := 0;      // configure which Drive to capture an input
A1_TrigRef.Direction := 1;   // configure Signal transition Direction of
capture
A1_TrigRef.TrigMode := 1;    // configure Time or Position Capture
```

3. Activate the TouchProbe function.

```
Inst_MC_TouchProbe( True, AxisID, A1_TrigRef, FALSE, 0, 0 );
```

NOTE

The position capture can be time-based (`A1_TrigRef.TrigMode := 0`) or position-based (`A1_TrigRef.TrigMode := 1`).

TIP

Fast Homing based on the drive's high speed capture mechanism is also supported by these Kollmorgen UDFBs in the IDE Function Block Library:
MCFB_StepAbsSwitchFastInput and MCFB_StepLimitSwitchFastInput

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.7.8 Pipe Network Homing

UDFBs can be written to provide specific canned homing cycles based on feedback type.

Contact "Support and Services" (➔ p. 1073) for more information.

15.7.8.1 Example: Fast Homing with the Pipe Network Motion Engine Axis Pipe Block

This example explains how to use the motion library functions of the axes when you want to detect the positive edge of the first Fast Input in the drive and read its associated timestamp.

MLAxisIsTrigged can be used to accurately perform a homing procedure in a user-defined procedure.

Procedure

1. MLAxisCfgFastIn (PipeNetwork.AXIS1, MLFI_FIRST, MLFI_RISING_EDGE)
   - Configure Fast Input 0 of AXIS1 to be triggered on the positive edge.

     NOTE

     Since tMLFI_FIRST and MLFI_SECOND correspond to the physical Fast Inputs 1 and 2.
     The drive must be configured to link fast input 1 with engine 0, and fast input 2 with engine 1.
   - The first argument indicates the Axis pipe block in the Pipe Network that represents the drive to be configured.
   - The second argument identifies which of the two Fast Inputs of the drive is configured (can be 0 or 1).
   - The third argument can indicate detection of positive edge when set to 1 and detection of negative edge when set to 2.
     - If set to 0, Fast Input is disabled.
2. MLAxisIsTrigged(PipeNetwork.AXIS1, 0, 1)
   - This function returns true if Fast Input 0 of AXIS1 has been triggered on the positive edge.
   - The meaning of the arguments is the same as in **MLAxisCfgFastIn**.
3. MLAxisTimeStamp (PipeNetwork.AXIS1, 0, 1)
   - This function returns the time in microseconds when the Fast Input was triggered on the positive edge.
   - This time is relative to the start of the drive cycle time.
     - Its value is explained in "Interpret a Timestamp" (➔ p. 859).
   - The meaning of the arguments is the same as in **MLAxisCfgFastIn**.
4. MLAxisRstFastIn (PipeNetwork.AXIS1, MLFI_FIRST)
   - This function resets the Fast Input 0 of AXIS1.
   - The reset keeps the configuration of the Fast Input, but it rearms it so it can be triggered again.
   - The meaning of the first two arguments is the same as in **MLAxisCfgFastIn**.
5. Follow-up Motion

This is typical code used in a homing procedure:

```
CASE StepCounter OF

0:

MLAxisRstFastIn(PipeNetwork.Feeder,MLFI_FIRST);

MLAxisMoveVel(PipeNetwork.Feeder,250.0); //Jog Feeder Axis to search for
sensor input

StepCounter := 1;

1:

IF MLAxisIsTrigged(PipeNetwork.Feeder,MLFI_FIRST,MLFI_RISING_EDGE)
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
   THEN MLAxisAbs(PipeNetwork.Feeder,MLAxisCmdPos(PipeNetwork.Feeder));
//Stop motion when sensor is reached
   StepCounter := 2;
END_IF;

2:

IF MLAxisGenIsRdy(PipeNetwork.Feeder)
   THEN MLAxisWritePos(PipeNetwork.Feeder,0); //Set Feeder Axis position to
zero
   StepCounter := 3;
END_IF;
```

- Fast Homing based on the drive's high speed capture mechanism is also supported by these Kollmorgen KUDFBs in the IDE Function Block Library:
  - MLFB_HomeFindHomeFastInput
  - MLFB_HomeFindHomeFastInputModulo
  - MLFB_HomeFindLimitFastInput
  - MLFB_HomeFindLimitFastInputModulo

The Axis Position can be derived from the Time Stamp using this code:

```
TimeStamp := any_to_lreal(MLAxisTimeStamp( iAxisID, any_to_dint
(ibFastInputNumber), 1+any_to_dint(ibHomeSwitchMode) ))/1000000;
CalculatedTriggerPosition := (CurrentVelocity-
PreviousVelocity)/2/iCycleTime*TimeStamp*TimeStamp+PreviousVelocity*TimeStamp
+PreviousPosition;
```

15.7.9 PLCopen Registration

Registration is a technique used to maintain the positional accuracy in repetitive processes.

- It uses a Fast Input switch, typically a photo eye, to measure product position and adjust the axis (or axes) to compensate for variations.
- The basic forms of registration are:
  - "Master/Slave Registration" (➔ p. 877)
  - "Single-Axis Registration" (➔ p. 878)

15.7.9.1 Decide which Function Blocks to Use for Registration

These are the methods to set up Registration, each using a different set of function blocks.

Method 1 is easy to set up, thus we recommend trying it first.

	Method 1: Using MC_TouchProbe	Method 2: Using MC_MarkRegist or MC_MachRegist
Key Function Blocks	• MC_MoveSuperimp • MC_TouchProbe • Trigger_Ref structure	• MC_MachRegist • MC_MarkRegist • MC_MoveSuperimp • MC_ReadParam • Trigger_Ref structure

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

	Method 1: Using MC_TouchProbe	Method 2: Using MC_MarkRegist or MC_MachRegist
Details	This method incorporates: • MC_TouchProbe to capture the drive position using the high speed capture engine in the drive. • The Trigger_Ref structure is used to set up the drive's capture engine. • MC_TouchProbe outputs a signal when a capture has occurred and also the position captured. • Additional code is created to make motion corrections (typically with MC_TouchProbe) based on the captured position value compared to a reference position or distance.	This method incorporates: • MC_MarkRegist or MC_MachRegist which, in addition to capturing a position from a drive, automatically calculates the position correction. • This can either: ◦ Make the offset move (linear acceleration / deceleration). ◦ Outputs the calculated position correction to be done by another function block (typically with MC_TouchProbe). • The Trigger_Ref structure is used to set up the drive's capture engine.
Advantages	Basic implementation. • Fast to develop due to having fewer input values to determine . • Simple function block functionality leads to less debug time.	Incorporates additional functionality including: • Automatically calculates the number of good and bad registration marks. • The option to only make corrective moves based on a good registration mark. • Automatically make the corrective motion (linear acceleration / deceleration only). • Position scaling differences are internally calculated if the axis in which the position is captured and the axis in which the motion compensation is made are different.
Example	PLCopen Registration Example Project	Rotary Knife Simulated Application

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.7.9.2 Master/Slave Registration

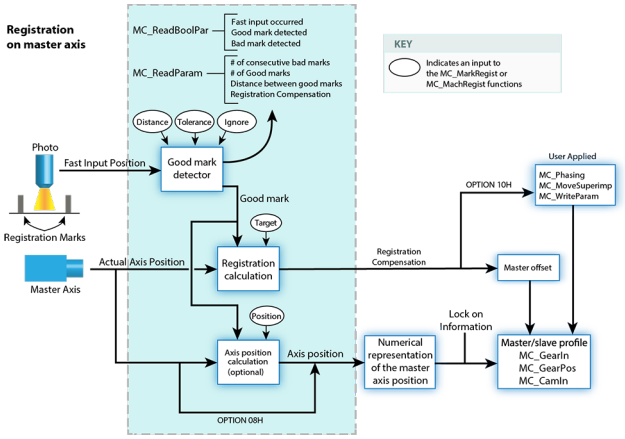
Master/slave registration is performed on an axis running a master/slave move (e.g., MC_CamIn or MC_GearIn).

- It can be performed by tracking the position of the master axis (Master Registration) or tracking the position of the slave axis (Slave Registration) or both.
- This type of registration:
  - Adjusts the positional relationship between the master and slave axes to accommodate for variations in the distance between products.
  - Keeps the process synchronized to the product over many repetitions.

15.7.9.2.1 Master Registration

Master registration is performed by having the Fast Input switch trigger on a registration mark controlled by the master axis.

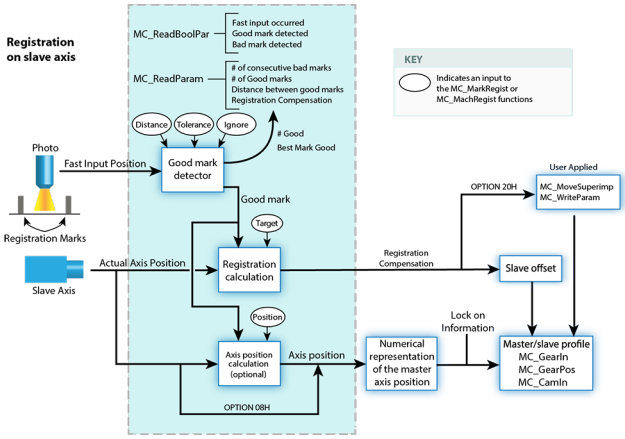
- When the Fast Input latches the position of the master axis at this mark, the distance between this position and the position of the previous mark is compared to an expected distance.
- This difference is added to the slave axis's master offset to adjust the position of the slave axis with respect to the position of the master.



15.7.9.2.2 Slave Registration

Slave registration is performed by having the Fast Input switch trigger on a registration mark controlled by the slave axis.

- When the Fast Input latches the position of the slave axis at this mark, the distance between this position and the position of the previous mark is compared to an expected distance.
- This difference is added to the slave axis's slave offset to adjust the position of the slave axis with respect to the position of the master.



This "Registration" (➔ p. 878) image is an example of a printing application using registration.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

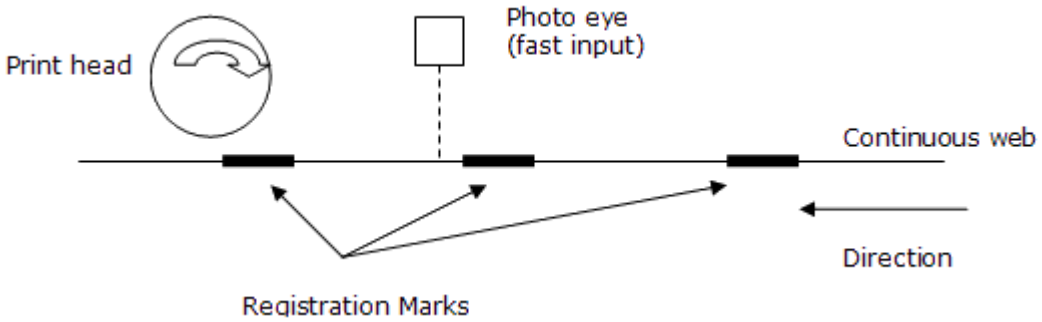


Figure 19-24: Registration

- The axis controlling the web is the master and the axis controlling the print head is the slave.
  - When the photo eye detects a registration mark on the web, the master position is latched.
- The amount of registration compensation required is calculated by comparing the actual distance between the marks to the expected distance.
  - That difference is added to the slave axis's master offset.
  - This adjusts the positional relationship between the web and the print head so that each print on the web is placed accurately.

15.7.9.3 Single-Axis Registration

Single-axis registration is performed on an axis running a discrete move such as MC_MoveAbsolute or MC_MoveRelative.

- When the Fast Input latches the position of the product, the axis position is reset, typically to 0 (zero).
- This resets the axis's coordinates for each product to accommodate for variations in the distance between products and keep the process synchronized to the product over many repetitions.

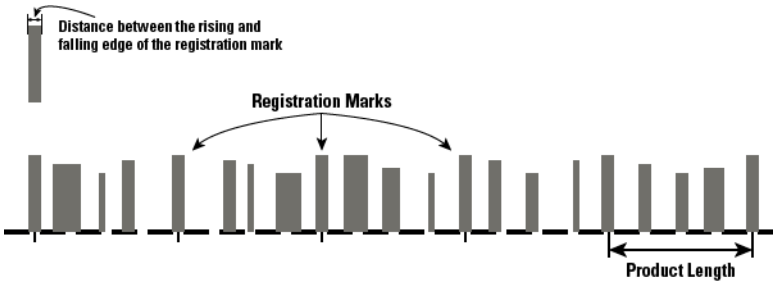
15.7.9.4 Registration Application Guide

In many closed-loop servo systems it is often necessary to maintain synchronization and accurate positioning repeatedly throughout a process. This can be difficult when the product or process itself is inconsistent. Using registration helps you to overcome this difficulty.

Many factors can contribute to inconsistency. Some examples are:

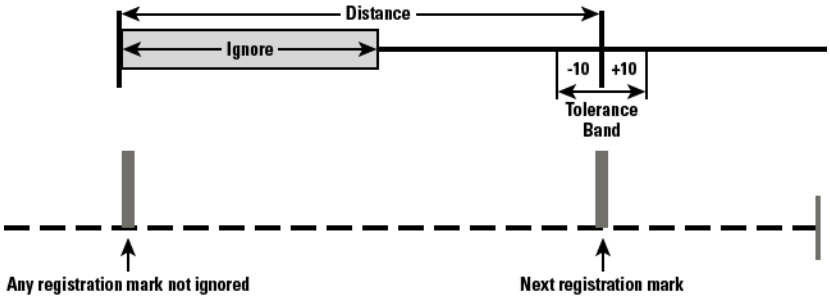
- Working with non-rigid material which may stretch or shrink during processing.
- Working with the mechanics of a system where the revolution of a feedback device may give you, for example, 5975 counts on one revolution and 5974 on the next.
- Unevenly spaced products on a web or belt.
- Materials which are sensitive to temperature, humidity, pressure, etc.

To overcome the various product and process inconsistencies registration capabilities may be required and can be applied on any servo or digitizing axis, and with any move type. Typically in these applications, sensors are used to detect the position of the product. With non-rigid materials, which may stretch or shrink or are unevenly spaced, a photo eye can detect registration marks on the material. With rigid products or processes a proximity sensor can detect leading or trailing product edges for material spacing.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

With registration, the registration sensor is wired to the fast input on the servo drive providing the means for the registration trigger that will capture the axis position at the instance of the registration event. When the event occurs, the system will calculate a registration correction to compensate for the inconsistency and then can apply the correction to the registration axis and/or change the numerical representation of the registration axis position.



This is important in applications such as printing, packaging, and converting where the process must be precisely coordinated and any non-rigid material cannot be depended upon to retain dimensional relationships. These applications usually involve master/slave moves. Not only does registration provide correction but the fast input signals can also be used as repeatable references to which the master and all subsequent slaves continually synchronize. Two main types of registration are "Mark to Mark Registration" (➔ p. 879) and "Mark to Machine Registration" (➔ p. 879).

15.7.9.4.1 Mark to Mark Registration

Mark to Mark Registration is implemented with the function block MC_MarkRegist and is based on the desired distance between two registration marks. This distance is used to qualify a “good” mark and then calculate a registration correction. Examples of the different mark to mark registration variations include:

- Clear Lane Registration
- Print Registration
- Product Registration
- Rotary Registration

Clear Lane Registration is the most common type of registration used in industry. A dedicated lane on the material is reserved solely for registration purposes. Only the registration marks in the lane will trigger the sensor and fast input on the servo drive. The distance between one registration mark and the next mark is the basis for registration correction calculations.

Print Registration is less common than Clear Lane Registration. Print registration involves picking out a distinct distance between print features. Here too, this distance is the basis for registration correction calculations. Print registration is used when it is not practical to have a clear lane.

Product Registration uses cycle position where the relationship of the product position in the process cycle is important. Product registration can occur synchronously or asynchronously. In synchronous product registration, products typically flow continuously through the process. Registration will measure and calculate small adjustments required to move each product to the correct position. The distance between products is the basis for registration correction calculations. Asynchronous applications may require the axis to sit and wait for the product to pass the sensor and, when it does, initiate a move and also apply registration correction based on the position of the product.

Rotary Registration is used to maintain a fixed axis position reference point on a rotary axis that has non-integer feedback in one revolution. If the axis position is not adjusted, the actual axis position would walk away from the desired position every revolution because of the non-integer number of feedback units per revolution.

15.7.9.4.2 Mark to Machine Registration

Mark to Machine Registration is implemented with the function block MC_MachRegist and is based on a desired target position on a machine axis. Like mark to mark registration, mark to machine registration uses the desired distance between two registration marks to qualify a “good” mark (using Clear Lane, Print, and Product registration). But instead of using the mark to mark length, it uses an axis position as its basis for correction.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.7.10 Pipe Network Registration and Fast Homing

15.7.10.1 Example: Fast Homing with the Pipe Network Motion Engine Axis Pipe Block

This example explains how to use the motion library functions of the axes when you want to detect the positive edge of the first Fast Input in the drive and read its associated timestamp.

MLAxisIsTrigged can be used to accurately perform a homing procedure in a user-defined procedure.

Procedure

1. MLAxisCfgFastIn (PipeNetwork.AXIS1, MLFI_FIRST, MLFI_RISING_EDGE)
   - Configure Fast Input 0 of AXIS1 to be triggered on the positive edge.

     NOTE

     Since tMLFI_FIRST and MLFI_SECOND correspond to the physical Fast Inputs 1 and 2.
     The drive must be configured to link fast input 1 with engine 0, and fast input 2 with engine 1.
   - The first argument indicates the Axis pipe block in the Pipe Network that represents the drive to be configured.
   - The second argument identifies which of the two Fast Inputs of the drive is configured (can be 0 or 1).
   - The third argument can indicate detection of positive edge when set to 1 and detection of negative edge when set to 2.
     - If set to 0, Fast Input is disabled.
2. MLAxisIsTrigged(PipeNetwork.AXIS1, 0, 1)
   - This function returns true if Fast Input 0 of AXIS1 has been triggered on the positive edge.
   - The meaning of the arguments is the same as in **MLAxisCfgFastIn**.
3. MLAxisTimeStamp (PipeNetwork.AXIS1, 0, 1)
   - This function returns the time in microseconds when the Fast Input was triggered on the positive edge.
   - This time is relative to the start of the drive cycle time.
     - Its value is explained in "Interpret a Timestamp" (➔ p. 859).
   - The meaning of the arguments is the same as in **MLAxisCfgFastIn**.
4. MLAxisRstFastIn (PipeNetwork.AXIS1, MLFI_FIRST)
   - This function resets the Fast Input 0 of AXIS1.
   - The reset keeps the configuration of the Fast Input, but it rearms it so it can be triggered again.
   - The meaning of the first two arguments is the same as in **MLAxisCfgFastIn**.
5. Follow-up Motion

This is typical code used in a homing procedure:

```
CASE StepCounter OF

0:

MLAxisRstFastIn(PipeNetwork.Feeder,MLFI_FIRST);

MLAxisMoveVel(PipeNetwork.Feeder,250.0); //Jog Feeder Axis to search for
sensor input

StepCounter := 1;

1:

IF MLAxisIsTrigged(PipeNetwork.Feeder,MLFI_FIRST,MLFI_RISING_EDGE)
   THEN MLAxisAbs(PipeNetwork.Feeder,MLAxisCmdPos(PipeNetwork.Feeder));
//Stop motion when sensor is reached
   StepCounter := 2;
END_IF;
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
2:

IF MLAxisGenIsRdy(PipeNetwork.Feeder)
   THEN MLAxisWritePos(PipeNetwork.Feeder,0); //Set Feeder Axis position to
zero
   StepCounter := 3;
END_IF;
```

- Fast Homing based on the drive's high speed capture mechanism is also supported by these Kollmorgen KUDFBs in the IDE Function Block Library:
  - MLFB_HomeFindHomeFastInput
  - MLFB_HomeFindHomeFastInputModulo
  - MLFB_HomeFindLimitFastInput
  - MLFB_HomeFindLimitFastInputModulo

The Axis Position can be derived from the Time Stamp using this code:

```
TimeStamp := any_to_lreal(MLAxisTimeStamp( iAxisID, any_to_dint
(ibFastInputNumber), 1+any_to_dint(ibHomeSwitchMode) ))/1000000;
CalculatedTriggerPosition := (CurrentVelocity-
PreviousVelocity)/2/iCycleTime*TimeStamp*TimeStamp+PreviousVelocity*TimeStamp
+PreviousPosition;
```

15.7.10.2 Example: Registration Position Capture with Pipe Network Trigger Block

A Pipe Network Trigger block can be used for registration applications to capture a pipe position based on a timestamp which is set when a drive's fast input changes state.

A Trigger block is placed in the Pipe Network precisely where the pipe position needs to be captured.

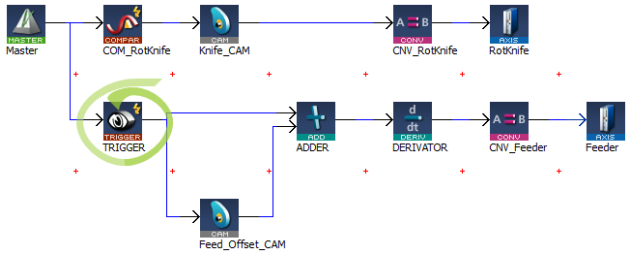


Figure 19-25: Example of using a Pipe Network Trigger block for position capture.

- The Pipe Network motion engine captures the pipe position by using the timestamp received from the drive and interpolating two consecutive pipe position updates.
- The timestamp will have a delay getting into the Pipe Network from the drive (typically 1-2 cycles) due to sensor delay and the communication latency through the EtherCAT network (typically 1-2 cycles).

```
Corrected timestamp: = Fast input timestamp - DelayCompensation
```

- To account for this delay, MLTrigWriteDelay provides delay compensation to the pipe position returned in MLTrigReadPos based on the expected delay.

The Trigger block is first configured by clicking on the block in a pipe network and setting these parameters:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Function	Description
INPUT_AXIS	• Defines the axis whose Fast Input is used. This name is the same given to the corresponding axis block in the Pipe Network
INPUT_ID	• Indicates which one of the two available Fast Inputs in that particular axis is used. • The value can be MLFI_FIRST or MLFI_SECOND for the trigger block to be triggered on the arrival of the first or the second input respectively. • Specify one of these constants: • MLFI_FIRST or MLFI_SECOND for the trigger block to be triggered on the arrival of the first or the second input respectively. • See "Use the Defines List" (➔ p. 523).
TRIGGER_ MODE	• Indicates if the trigger block responds to the rising edge or the falling edge of the Fast Input. • Specify either one of these constants: • MLFI_RISING_EDGE • MLFI_FALLING_EDGE • See "Use the Defines List" (➔ p. 523).

15.7.10.2.1 Configuration of the Trigger Block

After configuring the Trigger block, the order of calls to its motion library functions are:

1. MLTrigWriteDelay (DINT,TriggerID,LREAL,delay)
   - This function sets the delay compensation.
   - Typically starts with a value of 2 usec + (1 or 2) EtherCAT cycles.
2. MLTrigSetEdge (PipeNetwork.TRIGGER1,MLFI_FIRST,MLFI_RISING_EDGE)
   - This function reconfigures the edge of a trigger lock.
   - This function only needs to be called if the desired edge is different than the edge specified in configuration of the trigger block or if the edge is different than the previous capture.
3. MLAxisCfgFastIn (PipeNetwork.AXIS1, MLFI_FIRST, MLFI_RISING_EDGE)
   - This function call is necessary at least one time, even if the Trigger pipe block is configured properly.
4. MLTrigIsTrigged (PipeNetwork.TRIGGER1)
   - This function returns TRUE if the Fast Input associated to the Trigger pipe block given as argument has been triggered.
5. MLTrigReadPos (PipeNetwork.TRIGGER1)
   - This function returns the position of the Pipe Network at the time that the Fast Input associated with the Trigger pipe block was issued.

   NOTE

   You have to correct the position by taking into account the delay due to the number of cycles needed to read the timestamp of the Fast Input. You can use MLTrigWriteDelay to address this issue.
6. MLTrigReadTime (PipeNetwork.TRIGGER1)
   - This function returns the time associated with the Fast Input as explained in "Interpret a Timestamp" (➔ p. 859).
   - This function is of lesser importance compared to MLTrigReadPos.
7. MLTrigClearFlag (PipeNetwork.TRIGGER1)
   - This function rearms the Trigger pipe block.
8. MLAxisRstFastIn.(PipeNetwork.AXIS1, MLFI_FIRST)
   - This function rearms the Axis pipe block.

After the Trigger block pipe position is determined and calculations are made in the Registration program, a corrective move is typically done using a Cam Pipe Network block for precision motion control.

See Motion / Pipe Network - CAM Profile.

15.7.11 Practical Applications - Camming

These topics provide procedures to use when programming camming.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.7.11.1 Cam on the Fly

In addition to creating and modifying a Cam Profile using the IDE, an application programmer can also create or modify a new Cam Profile directly from their application.

- This allows new Cam Profiles to be defined while the application is still online, without stopping the machine to load a new application.
- An application programmer might use this feature to modify their application at runtime to adjust their Cam Profiles for varying product shapes and sizes.

Procedure

1. Before the motion engine is started, reserve memory for the profile that will be created on the fly.
   This is done using MLProfileCreate.
2. Define the profile properties and profile data points.
   This can be done at any time but must be done prior to calling MLProfileBuild.
3. Build the profile with MLProfileBuild.
   This takes a number of cycles and the profile will note be ready until the "Done" flag is set to TRUE.
4. The newly created profile can now be used exactly like a profile built in the IDE; it can be used in any number of camming relationships using any of the Pipe Network and PLCopen Cam Profile functions/function blocks.

Once you have created a Cam Profile using MLProfileBuild, if you need to modify that Cam Profile you must first release the existing Cam Profile. The following steps will release the Cam Profile.

1. Ensure the profile is not in use (Deactivate the existing profile). In Pipe Network perform a MLCamSwitch on an active Pipe to a different Profile or deactivate the pipe. In PLCOpen, perform an MC_CamOut on the profile or abort a profile move.
2. Call MLProfileRelease with the desired ProfileID as its input to release the profile so it can be modified.
3. When the "Done" flag is set, the ProfileID will be free. Modify the existing profiles data and then rebuild the profile MLProfileBuild.

NOTE

If a cam profile is Active, by having called MC_CamTblSelect and then MC_CamIn in PLCopen or MLCamSwitch in PipeNetwork, it cannot be released. In this situation use a second non-active cam to release, build (or rebuild), then activate.

An application is limited to 256 Cam Profile IDs. If additional profiles are desired, an existing profile can be released with MLProfileRelease. This frees its ProfileID so it can be used in the construction of a new profile. Releasing a profile is simple, and is performed with the following steps.

1. Ensure that the profile is not in use (Deactivate the existing profile). See for more details.
2. Call MLProfileRelease with the desired ProfileID as its input to release the profile so it can be modified.
3. When the "Done" flag is set, the ProfileID will be free and the old profile can be used for a new profile, calling either MLProfileInit or MLProfileBuild.

15.7.11.2 Position an Axis Before Starting Camming

The function block MC_CamStartPos is used to position a slave axis in its starting position for a MC_CamIn move with a slave absolute profile.

To position the slave axis for the MC_CamIn move, the typical programming sequence is:

1. With the master axis at standstill, call **MC_CamStartPos** to determine the start position for the slave axis.
2. Call MC_MoveAbsolute to move the slave axis to its start position.
3. Call **MC_CamIn** with `StartMode = 0` (Start mode).
   The MC_CamIn inputs **MasterOffset**, **SlaveOffset**, **MasterScaling**, and **SlaveScaling** should have the same values as used in the call to **MC_CamStartPos**.

See "CAM Profile Switching" (→ p. 251).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.7.11.3 Resume Camming After an E-Stop

The MC_CamResumePos function block is used to return a slave axis to its profile position after an event (such as an E-stop) caused the slave axis to go off path.

To return a slave axis to its MC_CamIn profile position, the typical programming sequence is:

1. With the master axis at standstill, call **MC_CamResumePos** to determine the profile position for the slave axis.
2. Call MC_MoveAbsolute to move the slave axis to the position calculated by **MC_CamResumePos**.
3. Call **MC_CamIn** with `StartMode = 1` (Resume mode).

15.7.12 Restarting Motion with Pipe Network

An advantage of the Pipe Network is to minimize machine downtime and reduce material waste when a non-fatal error occurs. After stopping the motion with MLAxisStop command, it can be restarting by using the MLAxisReAlign function block.

IMPORTANT

MLAxisReAlign must be called after the MLAxisStop command, otherwise all motion commands are ignored

For each axis:

Step	Example Application Code

Check Axis Status	`AxisStatus := MLAxisStatus(PipeNetwork.AXI_A1_Axis);`
	`IF AxisStatus.6 THEN`
	`  StepCounter := 1;`
	`END_IF;`

Turn axis back on (re-enable)	`IF MLAxisPower(PipeNetwork.AXI_A1_Axis, PowerUp) THEN`
	`  StepCounter := 2;`
	`END_IF;`

Calculate position difference between the Reference and Actual Positions	`DeltaPos := (MLAxisCmdPos(PipeNetwork.AXI_A1_Axis) - MLAxisReadActPos(PipeNetwork.AXI_A1_Axis));`

Determine how far to move	`IF DeltaPos > LREAL#0.5*DEF_A1_PosPeriod THEN`
	`  DeltaPos := DeltaPos - DEF_A1_PosPeriod;`
	`ELSE`
	`  IF DeltaPos < LREAL#-0.5*DEF_A1_PosPeriod THEN`
	`  DeltaPos := DeltaPos + DEF_A1_PosPeriod;`
	`  END_IF;`
	`END_IF;`
	`MLAxisReAlign(PipeNetwork.AXI_A1_Axis, 1000.0, 1000.0, 100.0, DeltaPos);`
	`StepCounter := 3;`

Wait for move to be completed	`IF MLAxisReAlgnRdy(PipeNetwork.AXI_A1_Axis) THEN`
	`  StepCounter := 4;`
	`END_IF;`

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

For the machine:

Step	Example Application Code
Execute multi-axis move	MLMstRun MLMstRun(PipeNetwork.MASTER, 500);

15.7.13 Superimposed Axes with PLCopen

This feature allows the application program to superimpose the moves of multiple axes (Superimposed Axes) on top of the move of another axis (Receiving Axis).

- This is performed internally by adding the command deltas of the Superimposed Axes to the command delta of the Receiving Axis.
- A maximum of four different Superimposed Axes can be superimposed upon a Receiving Axis.

15.7.14 Torque Feed-forward

The torque feed-forward tells the controller what forces is required to move the axis in an arbitrary trajectory.

These are the major features of torque feed-forward:

- Torque feed-forward results in virtually instantaneous response of the system.
- Feedback control loops (using PID loop or similar) take a finite amount of time before reacting.
- Torque feed-forward relies on an imperfect model of the system.
  - This means that the feed-forwards need help from the feedback control loop in order to get accurate motion.
- Torque feed-forward can make the bulk of the move very quickly, while the feedback control loops correct the small errors that remain.
  - As a result, a faster settling time can be achieved than if torque feed-forward was not used.
- There is a common misconception that torque feed-forward is similar to control loops and result in instability.
  - Torque feed-forward is open loop, so it cannot suffer from closed loop instability.
- Torque feed-forward is typically less sensitive to being misadjusted than closed loop parameters.
- Feedback control systems can be excited into instability by grossly misadjusted torque feed-forward.
  - However, the amount of misadjustment in the torque feed-forward necessary to cause such instability is very rare.

15.7.15 Tuning Controller Performance for EtherCAT Communication Latency

The response time for reading axis demand position data varies between different drive models.

- The response times also vary in reporting back TxPDO data such as actual position and time-based capture values.
- These different response times can lead to different lag in applications which use camming, gearing, and registration features.
- The KAS-IDE has several configuration parameters which allow you to tune your system to work best.

Example: Drive A has a 1 cycle latency for TxPDO data, and Drive B has 2 cycles latency.

The `MC_AXIS_PARAM_VELCOMP_FACTOR` parameter is used to specify 1 (for Drive A), and 2 (for Drive B) cycles.

The features and corresponding parameters are listed below.

TIP

If the drives in an existing system are changed (e.g., during an upgrade from AKD to AKD2G drives), the parameters need to be retuned to maintain optimal performance.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Feature	Parameters	Read Me For More Information
PLCopen Camming PLCopen Gearing	Axis parameters: • MC_AXIS_PARAM_ VELCOMP_FACTOR • MC_AXIS_PARAM_ VELCOMP_FILTER	• "Axis Parameters" (→ p. 347) • "Velocity Compensation" (→ p. 378)
PLCopen TouchProbe PLCopen Registration	Axis parameter: MC_AXIS_PARAM_SENSOR_ DELAY	• MC_TouchProbe and Time-Based Capture (KDN Article) • "PLCopen Registration" (→ p. 875)
Pipe Network External Feedback	Use a Phaser block to compensate for the delay	Motion Library - Phaser
Pipe Network Trigger Block	Trigger Delay	• MLTrigReadDelay • MLTrigWriteDelay

NOTE

The AKD drive supports an object to configure the EtherCAT input data for immediate or 1 cycle delay (default for backwards compatibility).

15.8 Motion Bus and Fieldbuses

Depending on the fieldbus used in your project (i.e., EtherCAT, PROFINET, etc.), you must use the configuration tools.

- "EtherCAT" (→ p. 887)
- "EtherNet/IP" (→ p. 887)
- "Modbus and TCP/IP" (→ p. 887)
- "PROFINET" (→ p. 888)
- "OPC UA" (→ p. 888)

The controller can be connected to multiple Ethernet devices simultaneously by connecting a switch to the top of the controller:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

PCMM2G

X31 or X32 port on the top of the controller:

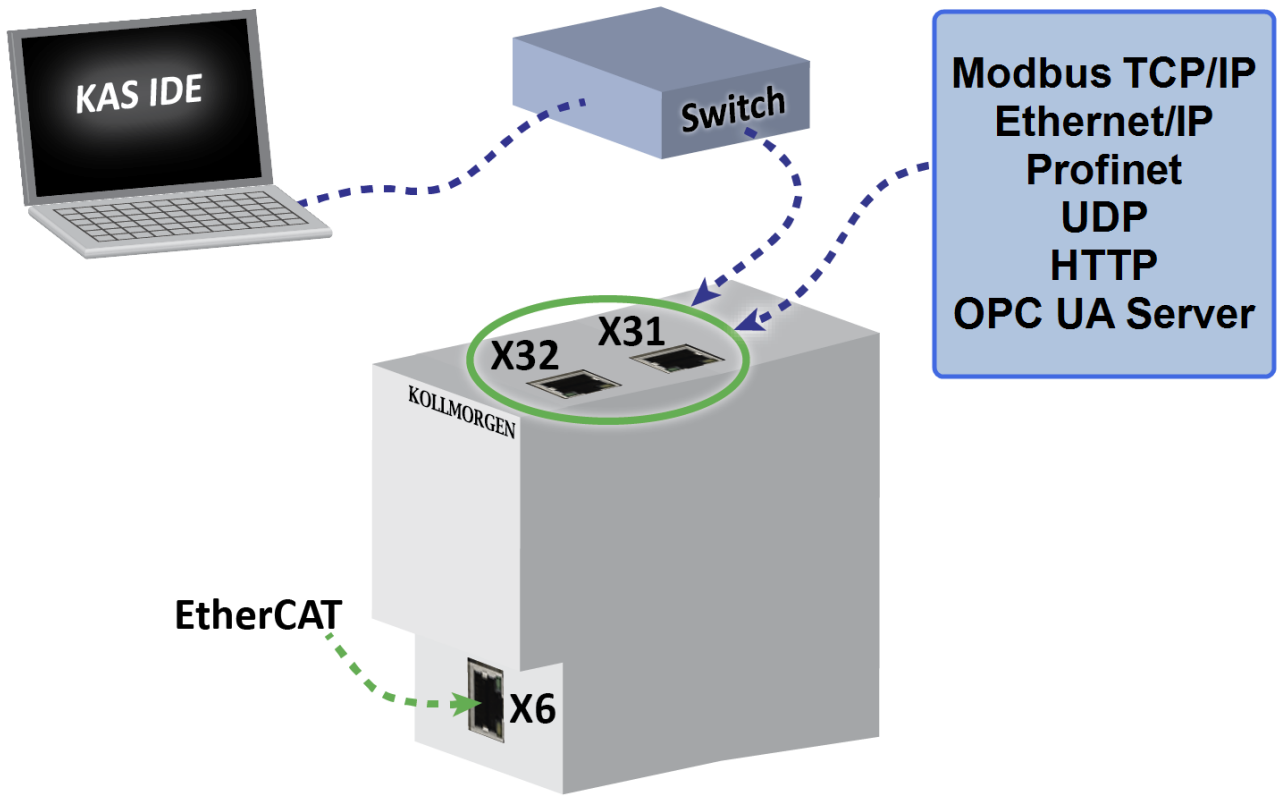


Figure 19-26: PCMM2G X31 and X32 switch connections.

TIP

See the "Controller Performance Guidelines" (→ p. 1006) on how the CPU load is affected when transmitting and receiving data between network devices and PLC variables.

15.8.1 EtherCAT

- For configuration, see "Configure EtherCAT Motion Bus" (→ p. 164).
- For I/O mapping, see "Map Input and Output to Variables" (→ p. 526).
- For error management, see "EtherCAT Error Messages" (→ p. 808).
- For EtherCAT XML Device Description, see the Beckhoff website.

15.8.2 EtherNet/IP

The KAS Runtime includes a fully integrated EtherNet/IP Adapter driver and Scanner driver for exchanging data with EtherNet/IP tag-based devices such as PLCs.

- EtherNet/IP Adapter (server), Scanner (client), Tag Client, and FlexIO/Point IO configurations are supported.
  - Use drag and drop for mapping PLC variables to PROFINET.
- Setting up "EtherNet/IP IO Scanner (Client)" (→ p. 898).
- Setting up "EtherNet/IP Adapter (Server)" (→ p. 895).
- Setting up "EtherNet/IP Tag Client" (→ p. 901).

15.8.3 Modbus and TCP/IP

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Setting up "Modbus Slave" (➔ p. 907).

15.8.4 PROFINET

This fieldbus is PROFIBUS over Ethernet.

We define one controller and have one or more devices.

Configuration

- For configuring the controller, see "Configure a PROFINET IO Controller" (➔ p. 910).
- For configuring devices, see "Configure a PROFINET IO Device" (➔ p. 915).

15.8.5 OPC UA

The Open Platform Communications Unified Architecture (OPC UA) is a standardized, platform-independent communication protocol developed by the OPC Foundation.

- It is designed for secure, reliable, and interoperable machine-to-machine communication in industrial automation environments.
- The KAS Runtime includes a fully integrated OPC UA Server driver for exchanging data with OPC UA client devices (e.g., PLC, HMI, etc.)
- See "OPC UA Server" (➔ p. 930).

15.8.6 I/O Mapping (for PROFIBUS Fieldbus)

This procedure describes how to map inputs and outputs to PLC variables on the Profibus fieldbus.

The mapping can be done from the Dictionary, but also with the "Custom Input/Output Editor" (➔ p. 708).

NOTE

See (➔ # 526, Map Input and Output to Variables) for more information about remote IOs on EtherCAT Motion Bus.

To map a variable from the Dictionary to a physical input or output:

1. Open the Variable list editor available in the **Dictionary** toolbox.
2. Right-click on the variable to be mapped.
3. Select the **Variable I/O Mapping** command in the menu to open the mapping dialog.

   By default the setting is NONE which means that the variable is a standard variable.
4. Select I/O (instead of NONE) and the I/O configuration panel appears:

This form is used to configure the different types of I/Os supported by KAS by defining these parameters:

Field	Description
Type	Defines the I/O type of fieldbus: CIFDriver for Profibus.
Direction	Specifies if the variable is an Output or an Input.
BitSize	• Defines the length of the frame to be mapped. • See "Data Types" (➔ p. 286) for information about length.
DeviceId	Defines the address of the I/O communication card located on to the target device (i.e., PAC).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Field	Description
SlaveId	Defines the address of the I/O node on the fieldbus ring (also see "Communication and Fieldbus" on page 79) For EtherCAT, a fixed address is assigned to each slave node that follows this convention: • First slave item on the network has address 1001. • Second slave item has address 1002, and so on.
ModuleId	For the current variable, defines the address identifier (ID) in the slice
BitOffset	Set to the first bit in the module of the slice which is mapped

To map a variable on Profibus, define these fields:

Field	Definition
I/O type	Enter CIFDriver.
DeviceId	Set to 0.
SlaveId	Set to the id of the I/O node.
ModuleId	Set to the id of the slice.
BitOffset	Set to the first bit of the slice which has to be mapped.

NOTE

For some drivers, you can select CUSTOM.

See "Modify an Input/Output" (➔ p. 709) for more information about the format of the **Description** field.

15.8.7 Add an Unsupported EtherCAT Device

NOTE

This procedure is for **advanced** users only.

When your project contains EtherCAT devices **not** supported by KAS, you must create the configuration using an external tool, and perform these steps:

1. Get the AKD device description ESI file from the official AKD distribution.
2. Verify all the device description ESI files are available for the external tool.
3. Use the external tool and do all the configuration, including these points:
   a. Set the Cycle Time.
   b. Turn on the distributed clocks option for all slave drives in order to share a global system time through EtherCAT.
   c. Assign PDO to each drive (inputs and outputs).
   d. Set the mode of operation of the drives into **position** mode.
   e. Insert variable names and do the mapping (see "Map PLC Variables" (➔ p. 890)).
4. Use the external tool to export the ENI description file.
5. In KAS, import the ENI file describing all the EtherCAT devices included in the project.
   See "ENI File tab" (➔ p. 541).

IMPORTANT

Importing an external ENI file overrides all EtherCAT project device information and configuration settings in the KAS-IDE.
These views and configurations are **not** applicable when using an imported ENI file:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Project View: All devices located under the EtherCAT node.
- EtherCAT Device View tabs:
  - General Properties
  - PDO Selection/Mapping
  - Distributed Clock
  - CoE Init-Commands
- Slice I/O Properties
- Mapping PLC Variables to Slice I/O or PDO objects.

Information displayed in the views may not match the imported ENI file.

15.8.7.1 Modify the EtherCAT Image in Cyclic Mode

In the application program, when integrating non-standard EtherCAT devices, use these function blocks to update the EtherCAT frame.

- ECATReadData
- ECATWriteData

See "EtherCAT Motion Bus Concepts" (➔ p. 416).

15.8.7.2 Configure the EtherCAT Device

Use these Functions Blocks:

- ECATReadSdo
- ECATReadSdoData
- ECATWriteSdo
- ECATWriteSdoData

15.8.7.3 Map PLC Variables

When you use an XML network description file generated with an external configurator, you need to add special tags to the PDO names to ensure the PLC variables can be mapped to IO channels.

The tags must comply with this convention:

```
@Scope.VariableName+StartBit-Size
```

Field	Description
@	Prefix with character **@** the PLC variable names of each of the image attributes that must be mapped.
Scope	Scope can be: • (Global) • ("Retain Variables" (➔ p. 292)) • Parentheses must surround the scope when it is Global or Retain. • Program Name • Even for the case of nested child SFC programs, the variables still belong to a unique, well-defined subprogram.
+StartBit	Optional: Integer that defines the bit from which the data must be written or read from the PLC variable.
-Size	Optional: Integer that defines the number of consecutive bits in the image which must be copied to/from the PLC variable. When present, this setting has precedence over the <BitSize> tag of the XML file.

15.8.7.3.1 Examples

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
(Global).MachineState
(Global).bLedStatus:0-1
(Global).bLedStatus:1-1
(Global).myINT:+4
main.variable:3+4-8
```

NOTE

This convention is applicable for simple variables.
KAS does not yet support mapping for **Structs and Arrays**.

15.8.8 Set Up FSoE Master and an AKD2G with SafeMotion Monitor

This topic describes how to:

- Manually set up FSoE PDOs.
- Copy links for a FSoE Master and an AKD2G with SafeMotion Monitor.

This example uses a simple EtherCAT topology consisting of:

- A Controller (PxMM or PCMM2G).
- A BBH SCU safety master (**Device_1**).
- An AKD2G with SMM Basic safety drive (**AKD2G_1**).

TIP

If using a BBH SCU-1-EC and SafePLC2 software, the FSoE PDOs can be automatically setup by the KAS-IDE and SafePLC2.
See "SafePLC2 Projects" (➔ p. 219).
For all other third party FSoE Masters, use this procedure.

15.8.8.1 1. Create Safety Application

Use the 3rd-party Safety PLC software to create a safety application for the FSoE master.

Example: The application will contain a FSoE master and an AKD2G with Axis1 and Axis2 as slave devices.

1. Set the AKD2G Connection IDs to match the the AKD2G SafeIDs.
2. Compile and download the safety application to the FSoE Master.
3. Run the safety application in FSoE Master.
   The state is **Run Intern** and the FSoE state is **8**.

15.8.8.2 2. KAS-IDE Project Set Up

To finish the example, a project is created in the KAS-IDE using the 2-axis Pipe Network template.

1. The IP address of the controller is set in the project.
2. Scan the EtherCAT network to discover the devices. (Figure 19-27)

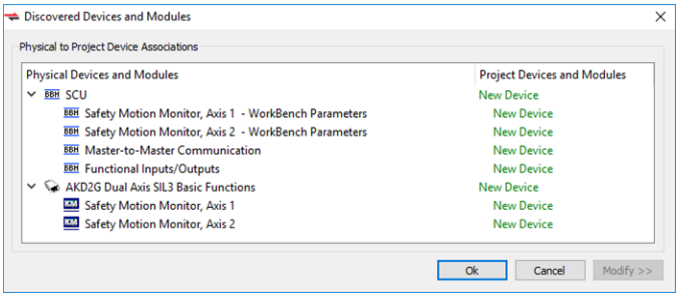


Figure 19-27: Discovered devices

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

3. We are prompted to recompile the project due to changes in the EtherCAT topology.
   We respond No because we have not set up the PDO copy links.
4. To connect the PDOs we open the "FSoE / PDO Connections tab" (➔ p. 546) which is populated with Input and Output PDOs. (Figure 19-28)

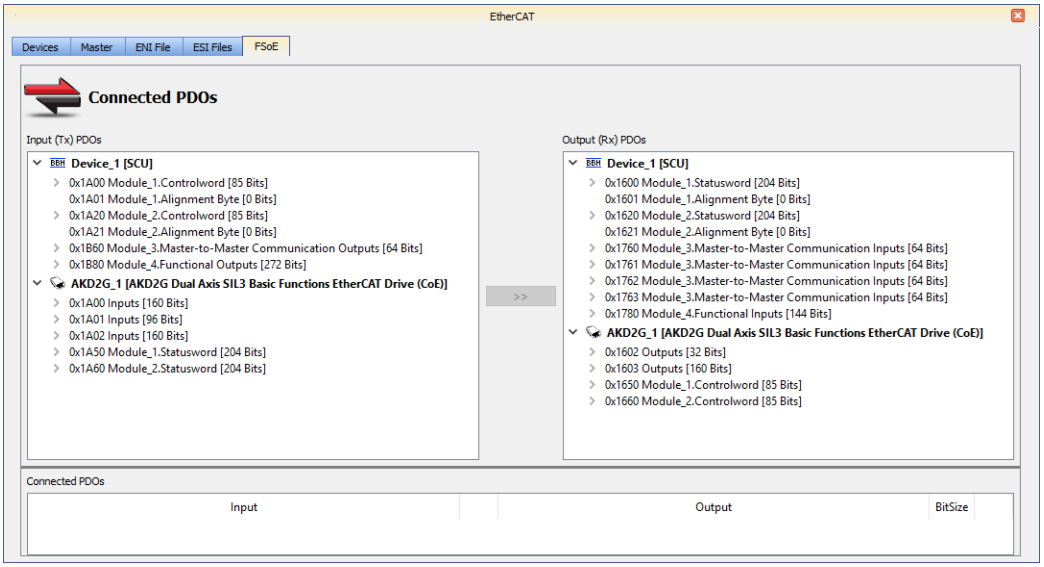


Figure 19-28: Connected PDOs

5. To connect PDOs we choose an input, an output, and click the **Connect** button.
   Connections are displayed at the bottom of the tab. (Figure 19-29)

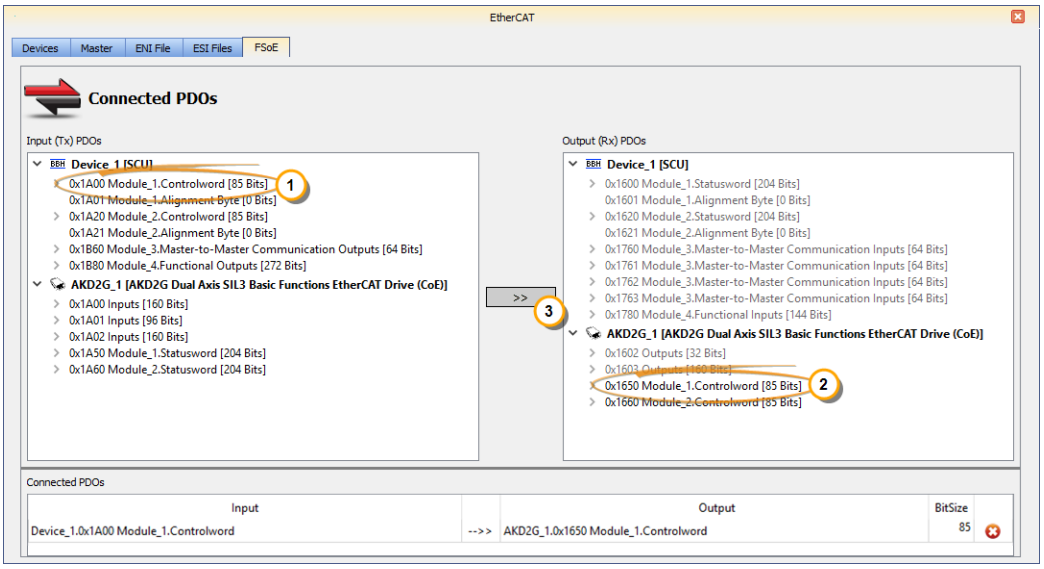


Figure 19-29: Chosen input and output

6. We make these connections:
   - Device_1[SCU] PDO 0x1A00 to AKD2G_1 PDO 0x1650
   - Device_1 [SCU] PDO 0x1A20 to AKD2G_1 PDO 0x1660
   - AKD2G_1 PDO 0x1A50 to Device_1 [SCU] PDO 0x1600
   - AKD2G_1 PDO 0X1A60 to Device_1 [SCU] PDO 0x1620
7. Compile, download, and run the KAS application.
   The BBH master has changed the state to Run and the FSoE state is 4.

The Safety Master and the Safety Slave are now exchanging data over EtherCAT.

See Also

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "AKD2G Safety Parametrization Using FSoE with SCU-1-EC and PxMM / PCMM2G" (➔ p. 893)
- FSoEParamsInit
- "Troubleshoot FSoE Safety Parameters" (➔ p. 813)

15.8.9 AKD2G Safety Parametrization Using FSoE with SCU-1-EC and PxMM / PCMM2G

The safety features in the AKD2G safety drive (Basic and Advanced) can be activated through either Fail Safe over EtherCAT (FSoE) or digital input or both.

The source of activation is configured inside the safety parameters and it needs to match the system requirements.

This topic describes how to configure and enable safety functions through BBH's SafePLC and KAS-IDE applications.

Requirements

Hardware	• SCU-1-EC ("BBH FSoE Master" (➔ p. 767)). • PCMM2G • AKD2G safety drive.
Software	• SafePLC2 (BBH Safety Software). • KAS-IDE - v3.05 (or later).

Procedure

These steps are for safety parameterization via the direct method using the KAS-IDE and BBH SafePLC2 software.

1. Create a PLCopen-based project in the KAS-IDE.
2. Scan the EtherCAT network.
3. Configure the safety modules for the AKD2G devices.
   - Choose either the **SRA Parameters** or **SRA Parameters configured via WorkBench** module, based on the application need.
4. When using **SRA Parameters**, add the FSoEParamsInit function block to the project.
   This function block enables the controller to read the safety parameters from the FSoE Master (SCU-1-EC) and write them to the AKD2G.
5. Follow either of these procedures:
   - If SafePLC2 is installed on the same computer, use Direct Method Connection.
   - If SafePLC2 is installed on a different computer than the KAS-IDE, use Import Method Connection.
6. Compile and download the SafePLC application to the BBH safety master (SCU-1-EC) and run the application.
7. Compile and download the KAS application to the controller.
8. Start the KAS application.
   See "Device Control" (➔ p. 267).
9. Execute the FSoEParamsInit function block when the application is ready and EtherCAT is in OP state.

See Also

- "SafePLC2 Projects" (➔ p. 219)
- "Set Up FSoE Master and an AKD2G with SafeMotion Monitor" (➔ p. 891)
- FSoEParamsInit
- "Troubleshoot FSoE Safety Parameters" (➔ p. 813)

15.8.10 Fieldbus Editor

15.8.10.1 EtherNet/IP Controller Data Communications

This information applies to both the Scanner and Adapter:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- A maximum of four different connections:
  - One Adapter Connection with one O -> T Assembly and one T -> O Assembly.
  - A maximum of three Scanner Connections each with one O -> T Assembly and one T -> O Assembly.
  - Total: A maximum of eight Assemblies (4 O -> T and 4 T -> O).
- Connections can be:
  - All to one product (with unique IP addresses).
  - To a maximum of four products with four unique IP Addresses.
- Each assembly can transfer a maximum of 500 bytes.
- Maximum communications is 8 x 500 = 4000 bytes.
- Assemblies and connections can be mixed in any suitable manner.
  - Examples:
    - Set up the controller to talk to four different Adapters with each one having a O -> T Assembly and T-A Assembly.
    - Set up the controller:
      - As a Scanner (Client) to connect to three different EtherNet/IP Servers (Adapters) each with one O -> T Assembly and one T -> O Assembly.
      - As a Server (Adapter) to connect to a Scanner (Client) with one O -> T Assembly and one T -> O Assembly.

15.8.10.2 EtherNet/IP Controller EDS File

The KAS installation contains an EtherNet/IP EDS file for the Kollmorgen controller.

This file may be needed by other controllers, (e.g., PLCs, PCs, etc.) to configure the EtherNet/IP communication with a controller.

The `KAS_Controller_EIP.eds` file is located in the `\Astrolabe\Bin\EDS` directory, where the KAS software was installed.

By default, the EDS file is located here:

```
C:\Program Files (x86)\Kollmorgen\Kollmorgen Automation
Suite\Astrolabe\Bin\EDS\KAS_Controller_EIP.eds
```

Revision	Compatibility	Change
1.0	KAS v2.7 only	Initial revision.
1.1	KAS v2.8 only	Updated vendor information.
1.2	KAS v2.9 only	Support a maximum of four input and output assemblies when KAS Runtime is configured as adapter (server), up from 1.

15.8.10.2.1 Using EDS Files

An EDS file may be required by a third-party tool when the KAS controller is configured as an EtherNet/IP Adapter.

- KAS adapter input/output images are configurable.
- A configuration matching the adapter must be defined on the scanner side.

NOTE

The data limit is:
500 bytes of data maximum O(originator)->T(target) and
500 bytes of data maximum T(target) -> O(originator).
This is based on the EtherNet/IP specification.

The data block sizes in EDS files are predefined in bytes: 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 500.

Use the closest existing configuration size that exceeds the needed amount of data.

Example: If 40 bytes is sent O->T and 64 bytes is sent T->O, two 64 byte values must be selected.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

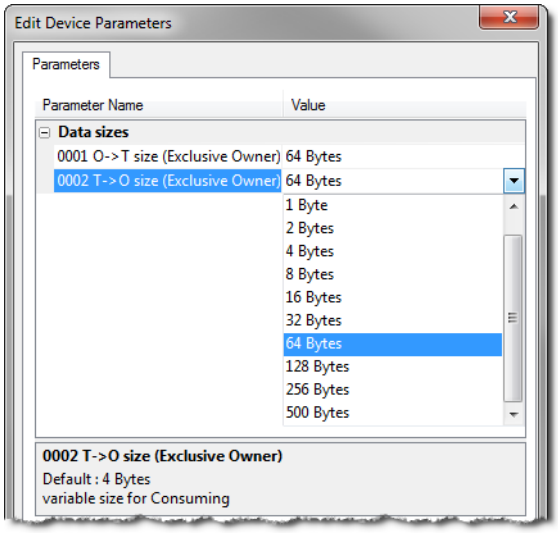


Figure 19-30: Example of setting byte sizes in a third-party Network Configurator.

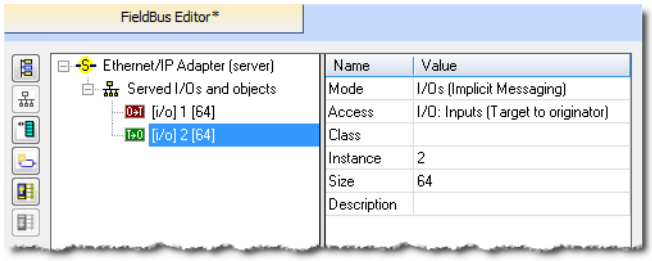


Figure 19-31: Example of setting byte sizes in the KAS-IDE Fieldbus Editor

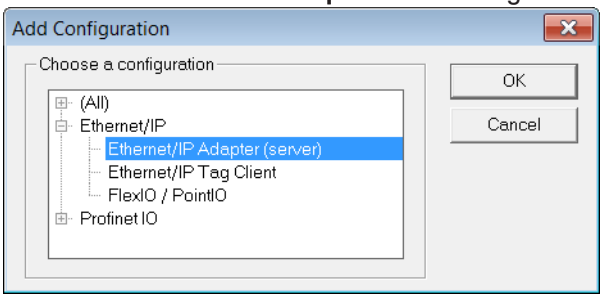
15.8.10.3 EtherNet/IP Adapter (Server)

The KAS Runtime includes fully integrated EtherNet/IP server driver for exchanging CIP I/O assemblies as an EtherNet/IP adapter in your applications.

15.8.10.3.1 Data Exchange - Configuration

A dedicated configuration tool is integrated in the KAS-IDE.

1. Double-click the Fieldbus node in the project explorer to open it.
2. Click the **Insert Configuration** icon to add the Fieldbus configuration.
3. Select the **EtherNet/IP Adapter** in the configuration selector.



The configuration is represented as a tree:

- EtherNet/IP I/O Scanner
  - Served I/Os and objects
    - I/O Assembly or Vendor Specific Object (*)
      - Exchanged Variable (*)

(*) The items with this mark can appear several times in the configuration.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.8.10.3.2 Configuration

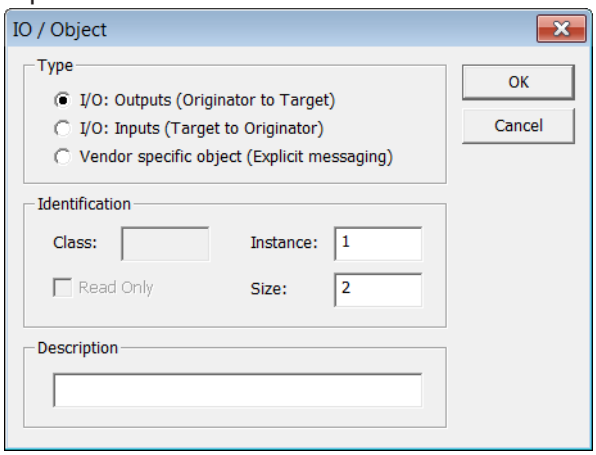
These items can be configured at the root level:

Identifier	Meaning
Use LAN2	Obsolete
IP Address	IP address of the Ethernet adapter used.

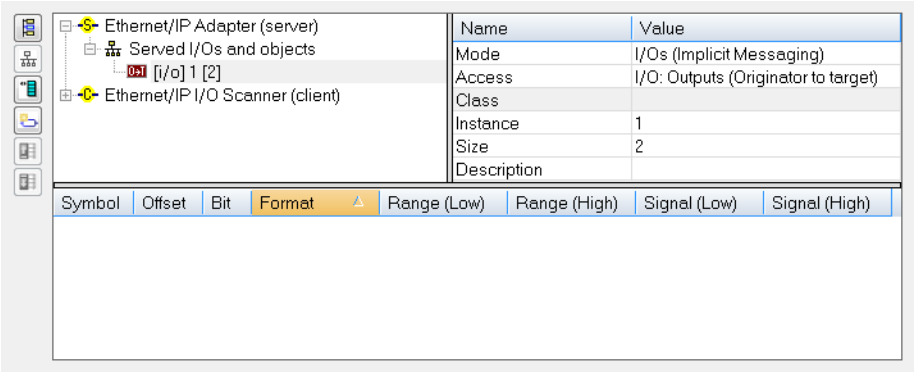
Click the Insert Master icon to declare a server (adapter device).

Each server is identified by its IP address and an optional Description text.

Select the **Served I/Os and objects node**, then click the Insert Slave icon to declare a CIP I/O assembly or a vendor specific object. Up to 4 input and 4 output assemblies are supported by the KAS Runtime, even though it is possible to create more in the KAS-IDE.



Each assembly is identified by:

Identifier	Meaning
Mode	The kind of CIP object. Either: • I/O assembly • Vendor specific object
Access	In case of a vendor specific object, this property defines the access rights: • Read/Write = free access • Read Only = the client (scanner) cannot write the object data
Class	CIP class in case of a vendor specific object. This field should be ignored in case of an I/O assembly. 
Instance	Instance of the CIP assembly or object.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Identifier	Meaning
Size	Data size in bytes.
Read Only	Specify that a vendor-specific object is only readable by clients.
Description	Optional description text.

When defining a vendor specific objects, these attributes are available for scanners:

- 1 (get only) = size of the object data.
- 3 (get/set) = object data.

Then map IEC 61131-3 variables on the data of the assembly, for each variable you must specify:

Identifier	Meaning
Symbol	The name of the IEC 61131-3 variable
Offset	Offset in bytes in the assembly data
Bit	Bit offset in the selected byte if format is "Bit"
Format	Format of the data in the assembly

TIP

Drag a variable from the Dictionary directly to a slave item.

NOTE

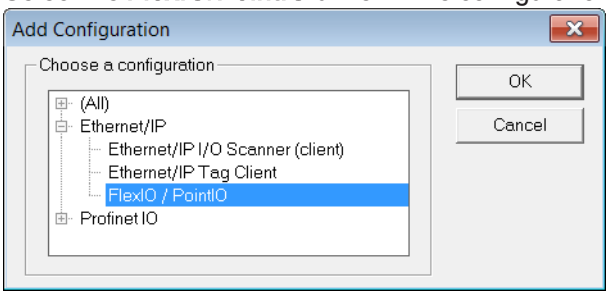
The data limit is:
500 bytes of data maximum O(originator)->T(target) and
500 bytes of data maximum T(target) -> O(originator).
This is based on the EtherNet/IP specification.

15.8.10.4 EtherNet/IP FlexIO / PointIO

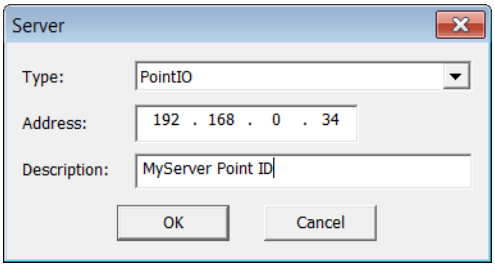
Before establishing the connection to the POINT IO, these modules require configuration.

This is done through the WEB interface of the POINT IO bus coupler.

1. Double-click the Fieldbus node in the project explorer to open it.
2. Click the Insert Configuration icon to add the Fieldbus configuration.
3. Select the **FlexIO/PointIO** driver in the configuration selector.



4. Click the Insert Master icon and select PointIO



5. Click the Insert Slave icon
   Only modules in the list are supported.
   When inserting, the module variables can be declared automatically by checking Declare variables and set a prefix.

NOTE

Modules need to be inserted in the right order.

Configuration is ready and you can download the application to the KAS Runtime.

15.8.10.5 EtherNet/IP IO Scanner (Client)

The KAS Runtime includes a fully integrated EtherNet/IP client driver for exchanging CIP I/O assemblies as an EtherNet/IP scanner in your applications.

- "Data Exchange Configuration" (→ p. 898)
- "Server Configuration" (→ p. 898)
- "Edit the Configuration Data Block" (→ p. 899)
- "Assembly Configuration" (→ p. 899)
- "Configuring Variables" (→ p. 900)

15.8.10.5.1 Data Exchange Configuration

A dedicated configuration tool is integrated in the KAS-IDE.

1. Double-click the **Fieldbus** node in the project explorer to open it.
2. Click the **Insert Configuration** icon to add the Fieldbus configuration.
   The **Add Configuration** dialog box opens.
3. Select the **EtherNet/IP I/O Scanner** in the configuration selector. (Figure 19-32)

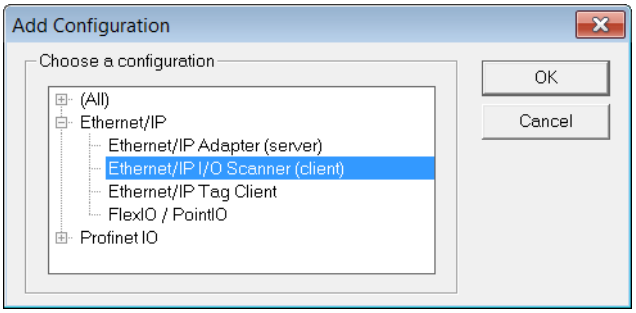


Figure 19-32: Add Configuration dialog box

The configuration is represented as a tree:

- EtherNet/IP I/O Scanner
  - Server (an EtherNet/IP adapter device) (*)
    - I/O Assembly (Originator to Target)
      - Exchanged Variable (*)
    - I/O Assembly (Target to Originator)
      - Exchanged Variable (*)

NOTE

(*) The items with this mark can appear several times in the configuration.

In addition to the IO configuration, explicit messaging can be performed using these function blocks:

- eipReadAttr
- eipWriteAttr.

15.8.10.5.2 Server Configuration

Click the Insert Master icon to declare a server (slave adapter).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Each server is identified by its IP address and an optional Description text. (Figure 19-33)

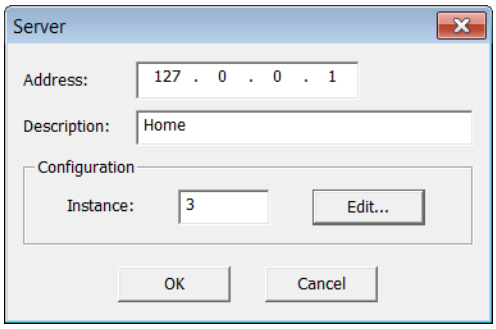


Figure 19-33: Server dialog box

NOTE

- The Server IP address can be represented as an alias and modified externally, without using the KAS-IDE, using a custom controller configuration file.
- See "Configure Custom Project Settings" (➔ p. 159).

These blocks of data are exchanged with the server for each connection:

- Outputs (originator to target)
- Inputs (target to originator)
- Configuration: a static block sent to the server at connection time.

The instance number for the configuration may be specified in the properties of the server.

15.8.10.5.3 Edit the Configuration Data Block

While most devices do not expect any configuration it may happen that some data is required. For this, double click the server in the tree and click the **Edit...** button in the dialog box.

- The configuration is entered as a list of items.
  - Use the **Type** button to specify the data types and number of items. (Figure 19-34)
- Structured data types defined in the KAS-IDE are supported.
  - Enter values in the grid.
- If you exchange multiple byte integers (e.g., WORD), you must specify whether the device expects little or big endian formatting.

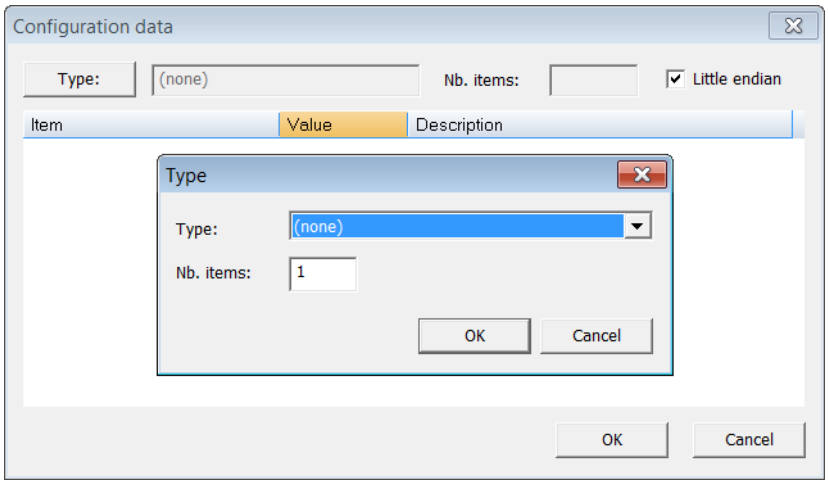


Figure 19-34: Configuration data - Type dialog box

15.8.10.5.4 Assembly Configuration

Click the **Insert Slave** button ( ) to declare a CIP I/O assembly.

Each assembly is identified by:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Identifier	Meaning
Type	Direction of the I/O assembly. Either: • Originator to Target (outputs). • Target to Originator (inputs).
Instance	Instance of the CIP assembly.
Config.instance	The instance number of the configuration assembly. The default value is 100.
Size	Data size in bytes.
Connection type	Type of the CIP connection. Can be Point To Point or MultiCast.
Priority	CIP priority: • Low • High • Scheduled • Urgent
32-bit header	Check this option if a 32-bit header is to be sent on notifications.
RPI(ms)	Minimum period for notification of changes in milliseconds. The range is 10 - 10000 ms.
Description	Optional description text.

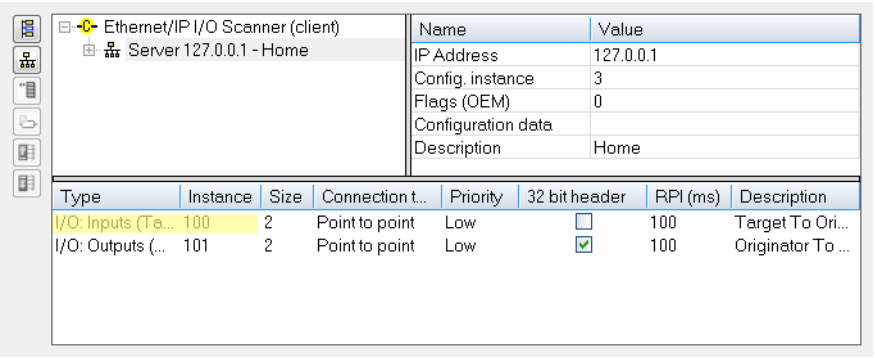


Figure 19-35: Example: EtherNet/IP Scanner (client)

15.8.10.5.5 Configuring Variables

IEC 61131-3 variables may be mapped on the data of the assembly.

Each variable must have these specifications:

Identifier	Meaning
Symbol	The name of the IEC 61131-3 variable.
Offset	Offset in bytes in the assembly data.
Bit	Bit offset in the selected byte if format is Bit.
Format	Format of the data in the assembly

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Identifier	Meaning
Mode	Kind of data exchanged through the variable: • Data Exchange: A type of input or output data in the assembly. • Server OK: Indicates the status of the IP connection to the server. • I/O connection OK: Indicates the status of the CIP I/O connection. • Last UCMM error: CIP code fo the last error occurred during a connection.

NOTE

The data limit is:
500 bytes of data maximum O(originator)->T(target) and
500 bytes of data maximum T(target) -> O(originator).
This is based on the EtherNet/IP specification.

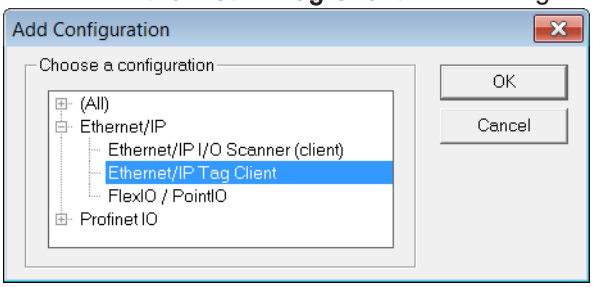
15.8.10.6 EtherNet/IP Tag Client

The KAS Runtime includes fully integrated EtherNet/IP client driver for exchanging tags with EtherNet/IP tag based devices such as PLCs.

15.8.10.6.1 Data exchange - Configuration

A dedicated configuration tool is integrated in the KAS-IDE.

1. Double-click the Fieldbus node in the project explorer to open it.
2. Click the **Insert Configuration** icon to add the Fieldbus configuration.
3. Select the **EtherNet/IP Tag Client** in the configuration selector.



The configuration is represented as a tree:

- EtherNet/IP Tag Client
  - Server (an EtherNet/IP adapter device) (*)
    - Tag (generally an array) (*)
      - Exchanged variable (*)

(*) The items with this mark can appear several times in the configuration.

Driver and configurator are optimized for exchanging arrays (tags declared as arrays in the PLC). However it is also possible to exchange single tags.

15.8.10.6.2 Configuration

Click the Insert Master icon to declare an server (slave device). Each server is identified by its IP address and an optional description text.

Then you need to configure tags such as declared in the PLC:

- The easiest way is to right-click on the server in the tree and select the **Add ARRAY Tag** command in the contextual menu. Then you enter the properties of the tag request and the symbol of the corresponding array to be used in your IEC 61131-3 application. Configuration of the tag and mapping of all array items is performed automatically.
- Alternatively, click the Insert Slaver icon to declare the tag and map some variables later on.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

A tag request is identified by:

Identifier	Meaning
Tag name	The name of the tag such as declared in the PLC.
PLC Slot	PLC slot number.
Mode	Read or Write. The same tag can be configured twice for both reading and writing.
Nb Elements	Number of array items to read or write.
Offset	O-based index of the first item to read or write in the array.
Tag data type	Data type of the tag such as declared in the PLC. Available Types are: • BOOL • Single Boolean variable on 1 byte - 00=FALSE / FF=TRUE. • DINT (32-bit signed integer) • DWORD (32-bit string) • DWORD • Should be selected if the tag is declared in the PLC as an array of bits. • INT (16-bit signed integer) • SINT (8-bit signed integer)
Period(ms)	Specify in this parameter a period for continuously sending the request. Enter 0 (zero) for a request sent on-demand.
Timeout	Request timeout in milliseconds.

IEC 61131-3 variables are mapped on the data of the tag.

For each variable you must specify:

Identifier	Meaning
Symbol	The name of the IEC 61131-3 variable.
Offset	Offset in bytes in the assembly data.
Bit	Bit offset in the selected byte if format is Bit.
Format	Format of the data in the assembly.
Mode	The kind of data exchanged through the variable: • Data Exchange: A piece of input or output data in the assembly. • Server OK: Indicates the status of the IP connection to the server. • Send Request Now: Used as a command for activating the request. • [transaction counter]: Increased each time the request is sent. • [general status]: CIP error code (0 = OK). • [extended status]: CIP extended error code (0 = OK).

The tag is read or written:

- Periodically if a non zero period is specified in the tag configuration.
- When a variable configured as **Send Request Now** becomes TRUE.

In the case of a command variable, the variable is automatically reset to FALSE when the request is sent.

NOTE

The data limit is:
500 bytes of data maximum O(originator)->T(target) and
500 bytes of data maximum T(target) -> O(originator).
This is based on the EtherNet/IP specification.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.8.10.7 Link with Embedded HTTP Server

This configuration:

- Activates the Web Diagnostic Panels feature.
- Allows users to list the PLC variables that can be exposed by the KAS Web server and their access rights.

Procedure

1. In the Project Explorer, double-click the Fieldbus node to open the Fieldbus Editor.
2. On the vertical toolbar, click the **Insert Network** (Configuration) button ( ). (Figure 19-36)

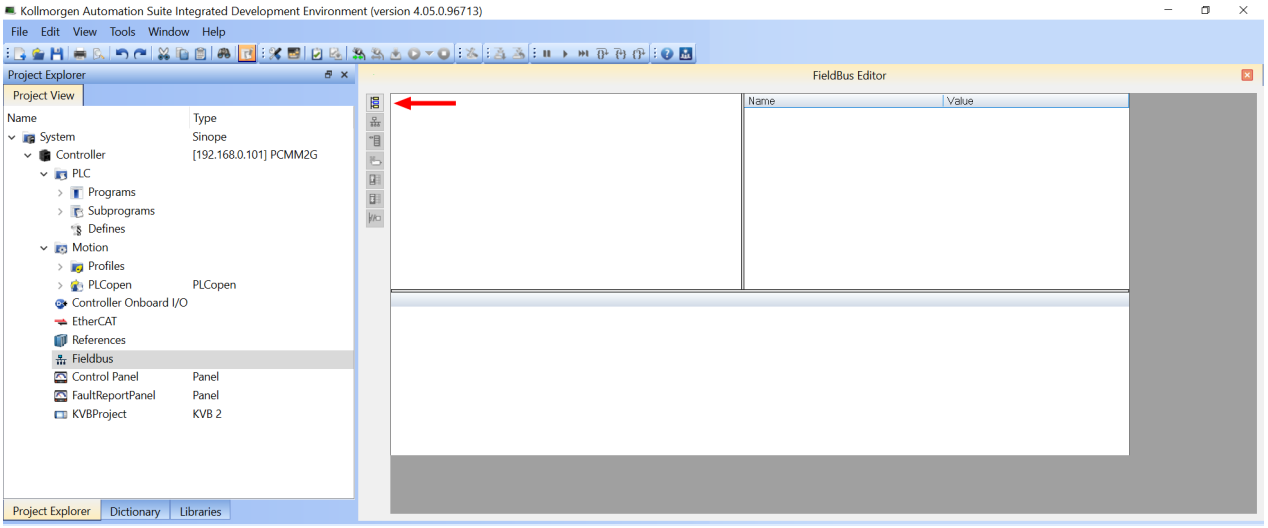


Figure 19-36: Insert Network (Configuration) button

The **Add Configuration** dialog box opens.

3. In the **Select a configuration** tree, select **Link with embedded HTTP Server**. (Figure 19-37)

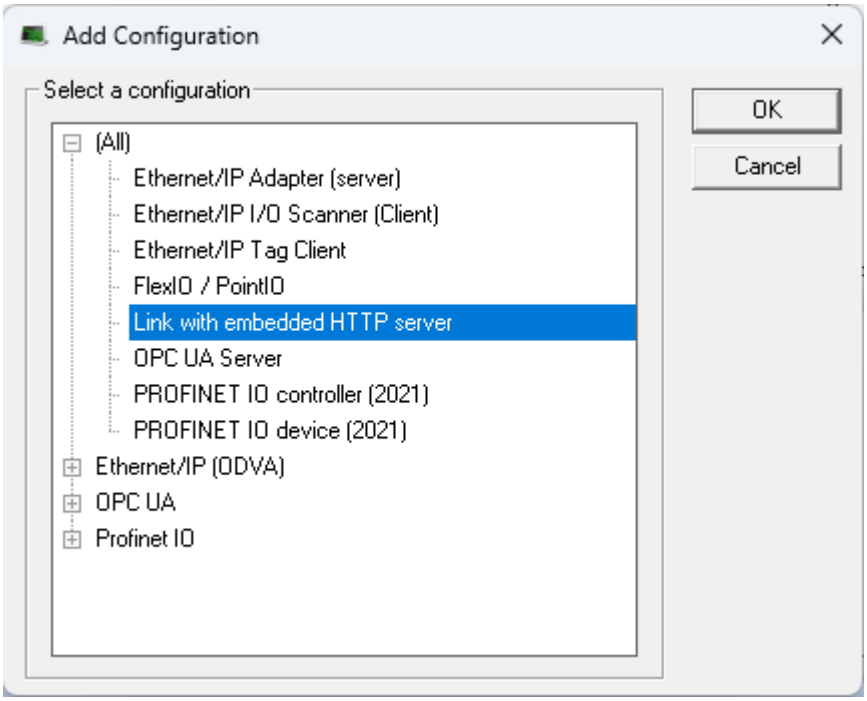


Figure 19-37: Add Configuration dialog box - Link with embedded HTTP Server selected

4. Click **OK** to save the changes or selections and close the dialog box.
The "Link with embedded HTTP Server dialog box" (→ p. 904) dialog box opens. (Figure 19-38)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

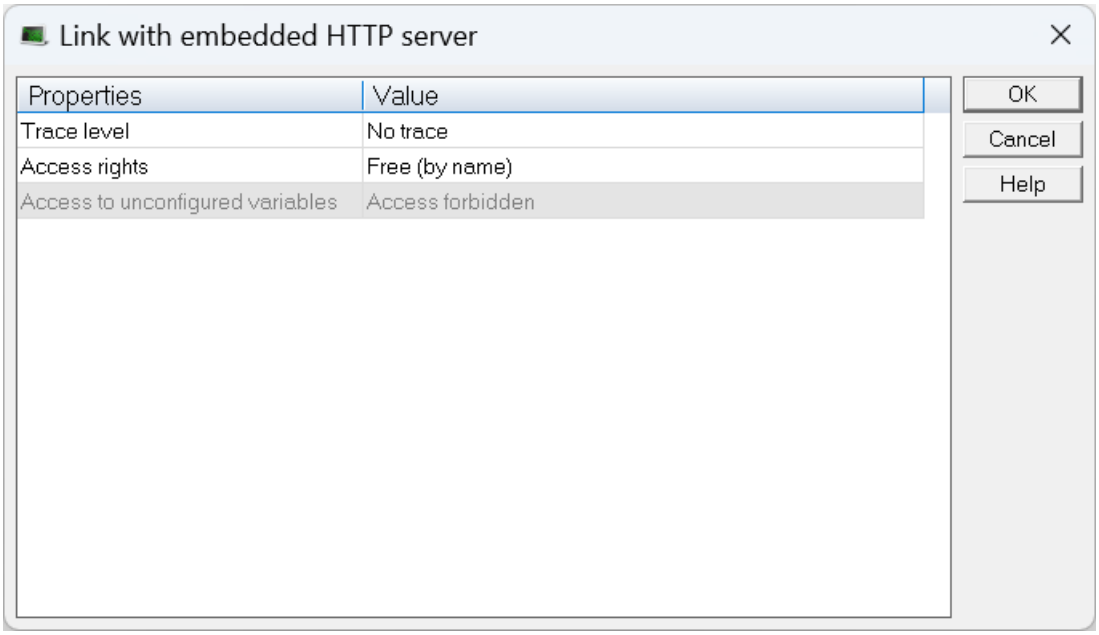


Figure 19-38: Link with embedded HTTP Server dialog box

5. In the **Trace Level - Value** column cell, double-click the cell to open the **Trace Level - Value** list box.
6. In the **Trace Level - Value** list box, select the option that defines the type of information written to the controller log output.
   The options are:
   - No trace (default)
   - Information
   - Debug
7. In the **Access Rights** row, double-click the **Value** column cell to switch the selection between **Free (by name)** or **Configured variables only**.
   - **Free (by name)**: When selected, the KAS Web server gives full Read / Write access to any PLC variable present in the KAS project.
   - **Configured variables only**: When selected, the PLC variables must be enumerated in the configuration tree.

   IMPORTANT

   Kollmorgen recommends using the **Configured variables only** option to avoid any possible unintended remote access to PLC variables.
8. In the **Access to unconfigured variables** row, double-click the **Value** column cell to open the **Access to unconfigured variables - Value** list box.
9. In the **Access to unconfigured variables - Value** list box, select the option that defines the default access rights for any variable not explicitly configured in this tree.
   The options are:
   - Read-only
   - Read / Write
   - Access forbidden (default)

   IMPORTANT

   Kollmorgen recommends accepting the default option of **Access of forbidden** for **Access to unconfigurable** setting element to avoid the risk of unintended access of application variables from external applications remotely
10. Click **OK** to save the changes or selections and close the dialog box.

See Also

Web Diagnostic Panels

15.8.10.7.1 Link with embedded HTTP Server dialog box

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Usage

The Link with embedded HTTP Server dialog box is used to enable viewing Control Panels in a web browser.

See:

- "Link with Embedded HTTP Server" (➔ p. 903)
- Web Diagnostic Panels

Access

1. In the Project Explorer, double-click the Fieldbus node to open the Fieldbus Editor.
2. On the vertical toolbar, click the **Insert Network** (Configuration) button ( ). (Figure 19-39)

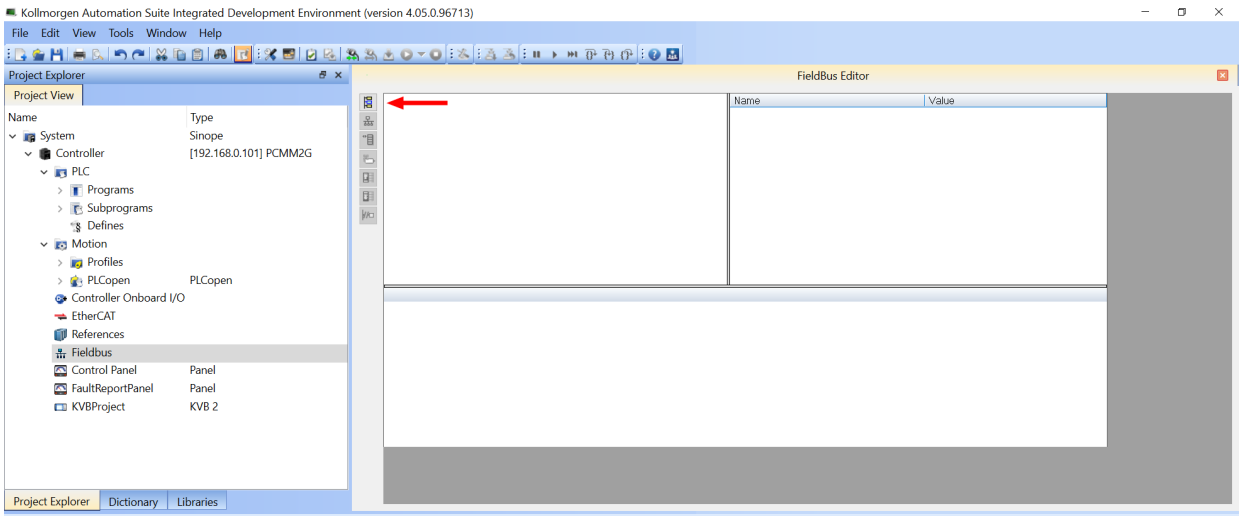
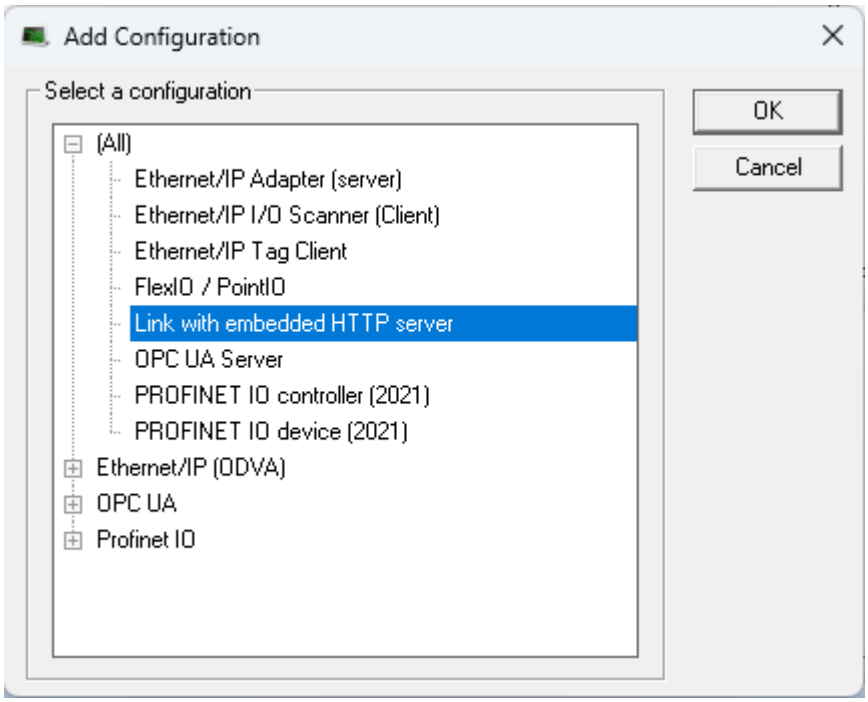


Figure 19-39: Insert Network (Configuration) button

The **Add Configuration** dialog box opens.

3. In the **Select a configuration** tree, select **Link with embedded HTTP Server**. (Figure 19-40)



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Figure 19-40: Add Configuration dialog box - Link with embedded HTTP Server selected

4. Click **OK** to save the changes or selections and close the dialog box.
   The **Link with embedded HTTP Server** dialog box opens. (Figure 19-41)

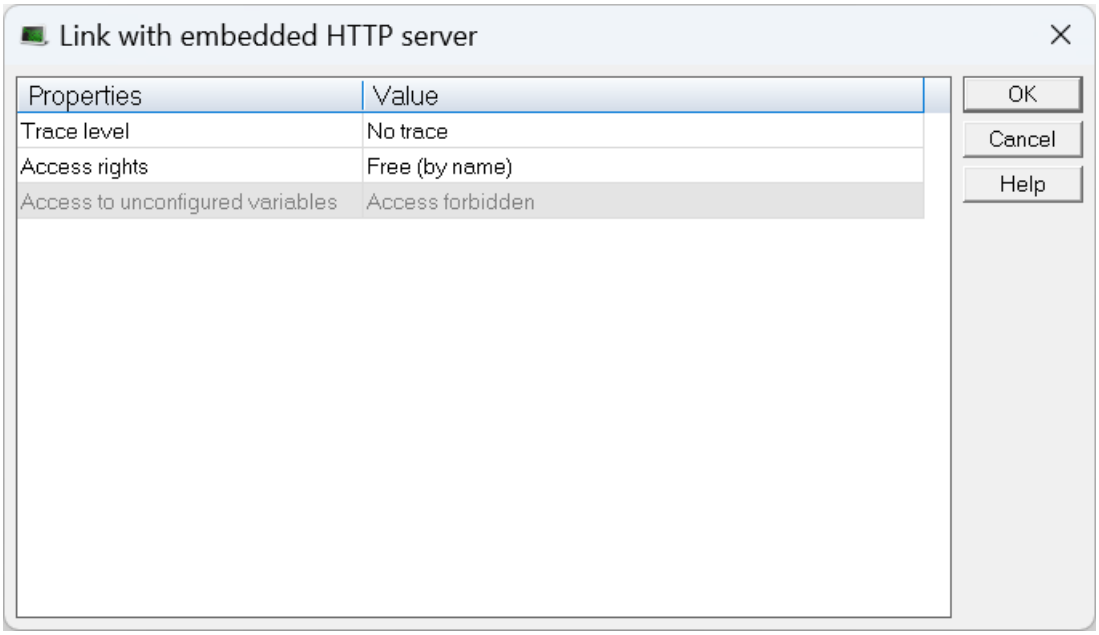


Figure 19-41: Link with embedded HTTP Server dialog box

Dialog box Description

Section	Element	Description
Properties column		The **Properties** column lists the properties to define for this variable.
Value column		The **Value** column shows the selected Trace Level, Access Rights, and Access to unconfigured variables options.
Trace Level row	**Trace Level** - **Value** list box	• In the **Trace Level** - **Value** column cell, double-click the cell to open the **Trace Level** - **Value** list box. • In the **Trace Level** - **Value** list box, select the option that defines the type of information written to the controller log output. The options are: • No trace (default) • Information • Debug

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Section	Element	Description
Access Rights row	**Access Rights - Value** cell	• The selection in the **Access Rights - Value** cell defines the PLC variables accessible from the KAS Web server and their access rights. • In the **Access Rights** row, double-click the **Value** column cell to switch the selection between **Free (by name)** or **Configured variables only**. The options are: • **Free (by name)**: When selected, the KAS Web server gives full Read / Write access to any PLC variable present in the KAS project. • This is the default selection. • **Configured variables only**: When selected, the PLC variables must be enumerated in the configuration tree. **IMPORTANT** • When **Configured variables only** is selected, click **OK** to save the selection and close the **Link with embedded HTTP Server** dialog box. • Re-open the dialog box to use the options in the **Access to unconfigured variables - Value** list box.
Access to unconfigured variables row	**Access to unconfigured variables - Value** list box	• In the **Access to unconfigured variables** row, double-click the **Value** column cell to open the **Access to unconfigured variables - Value** list box. • In the **Access to unconfigured variables - Value** list box, select the option that defines the default access rights for any variable not explicitly configured in this tree. The options are: • Read-only • Read / Write • Access forbidden (default) **NOTE** • This setting is only available when the **Configured variables only** option is selected in the **Access Rights - Value** cell. • If the **Access to unconfigured variables - Value** cell is grayed out when **Configured variables only** access rights is selected, close the **Link with embedded HTTP Server** dialog box and re-open the dialog box.
	OK button	Click **OK** to save the changes or selections and close the dialog box.
	Cancel button	Click **Cancel** to discard all changes or selections and close the dialog box.
	Help button	Click **Help** to open the help topic for this dialog box, panel, or window.

15.8.10.8 Modbus Slave

- "Protocol Specification" (→ p. 908)
- "Data Exchange Configuration" (→ p. 908)
- "Modbus Slave Configuration" (→ p. 909)
- "Data Types" (→ p. 910)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The KAS Runtime includes fully integrated slave functions for enabling Modbus communication on a serial link or Ethernet.

This communication is done in the background, asynchronously, at the cycle time (20-1000 milliseconds) specified in the Controller Properties. See "Configure the Controller" (➔ p. 156) for more information.

- Variables defined in the HMI to describe the interface are passed to the controller this way.
  - See "Map Variables to HMI" (➔ p. 216) for more information.
- This means there is no data coherency in the data exchange because the variables read by the Modbus do not come from the same PLC cycle.
- This data has a rather low priority.
  - It is interpreted by human feedback and should never be noticed by the user.

NOTE

Kollmorgen HMIs are limited to communicating no more frequently than every 100 milliseconds.

TIP

It is possible to have two Modbus master devices communicating with a controller at the same time, accessing the same variables.
Example: Having two separate HMI panels running the same program on a single, large machine.

15.8.10.8.1 Protocol Specification

The protocol supported is Open Modbus on Ethernet.

These are the supported Modbus function codes:

Code	Description
1	Read coils.
2	Read bit inputs.
3	Read holding registers.
4	Read input registers.
5	Write 1 coil.
6	Write 1 register.
15	Write **n** coils.
16	Write **n** registers.

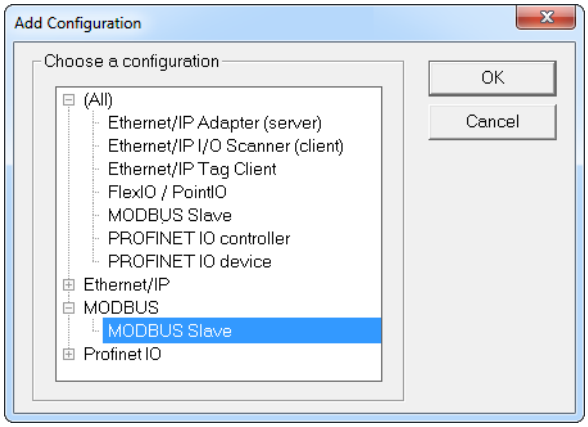
- As a default, the first valid address for each kind of data is 1.
- If you use Modbus devices with other addressing conventions, you can change the base offset for each kind of data using the **Tools** > **Addresses** menu command.

15.8.10.8.2 Data Exchange Configuration

The KAS-IDE has a dedicated configuration tool. To run it,

1. In the **Controller Properties** dialog box, select the **Other Modbus devices** option.
   See "Configure the Controller" (➔ p. 156) for more information.
2. Double-click the **Fieldbus** node in the project explorer to open it
3. Click the **Insert Configuration** icon to add the Fieldbus configuration.
4. Select the **Modbus Slave** in the configuration selector.
5. Click **OK** to continue.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



The Modbus Slave configuration is represented as a tree:

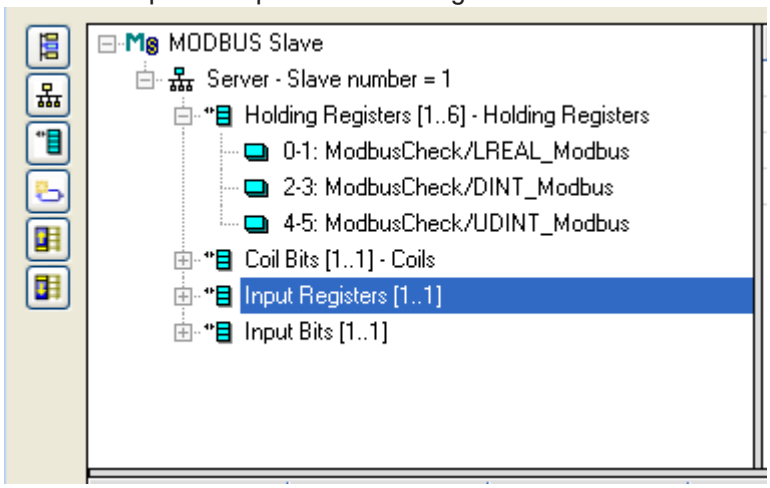
- Modbus Slave
  - Slave number (variables that can be accessed from external Modbus masters)
    - Input bits data block (read by masters)
      - Variable (*)
    - Input words data block (read by masters)
      - Variable (*)
    - Coil bits data block (forced by masters)
      - Variable (*)
    - Holding bits data block (forced by masters)
      - Variable (*)

(*) The items with this mark can appear several times in the configuration.

15.8.10.8.3 Modbus Slave Configuration

You need to configure the Modbus Slave to make variables visible from external Modbus masters (e.g., SCADA systems).

This is a simple example of slave configuration:



- Double click the **Server** item to setup the Modbus slave number that will identify the runtime application.
- When the local server is selected, use the **Insert Slave/Data Block** menu command to insert Modbus data blocks.
- These kinds of block are available:
  - Coil Bits: bits forced by by external masters (function 5 or 15).
  - Holding Registers: words forced by external masters (function 6 or 16).
  - Input Bits: bits read by external masters (function 2).
  - Input Registers: words read by external masters (function 4).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

TIP

For optimal controller performance use Input Registers where possible.
Input Registers require 15-25% less CPU time to exchange data than **Holding Registers** do.

Each data block is identified by a Modbus base address and a number of items (bits or words).

NOTE

Read and write requests sent by Modbus masters are denied if the range specified in the request does not fit within a data block defined in the configuration.
Requests overlapping two data blocks are denied.
Example: If you configure a block of 16 words starting at address 1 and another block of 16 words starting at address 17, a request for read or write of 32 words starting at address 1 is denied and an address error exception is reported.

When a server data block is selected, use the **New symbol** command to map a variable to an item of the data block.

Each variable is identified by a valid symbol of a variable in the open project and an offset in the data block according to Modbus addressing.

Exchanges

- For exchanging:
  - Boolean variables through Modbus words, a hexadecimal mask is available to define to which bit of a word a variable is attached.
    - Example: Enter the mask **0001** to map a Boolean variable to the less significant bit of a word.
  - 32-bit variables (e.g., DINT, REAL, etc.), select to map the variable on two consecutive words.
  - 64-bit variables(e.g., LINT, LREAL, LWORD, ULINT), select to map the variable on four consecutive words

Optional: Sort the variables of each data block according to their offset using the **Sort symbols** menu command at any time.

15.8.10.8.4 Data Types

You can freely map a variable of any data type to a Modbus item.

The Runtime automatically converts the value to the type of the variable.

15.8.10.9 PROFINET Configuration

These topics provide information about PROFINET configuration.

- "Configure a PROFINET IO Controller" (➔ p. 910)
- "Configure a PROFINET IO Device" (➔ p. 915)
- "PROFINET - Additional Features" (➔ p. 920)
- "PROFINET - Data Types" (➔ p. 922)
- "PROFINET - Error and Status Codes" (➔ p. 922)
- "PROFINET - Resolve Errors" (➔ p. 928)

NOTE

Unlike an EtherCAT fieldbus with a dedicated Ethernet interface, the PROFINET fieldbus communicates through the same Ethernet interface as Modbus and other non-fieldbus communication.
There is no prioritization of PROFINET over the other communication protocols.
Frequent HTTP or IDE Oscilloscope communication can cause delay in the PROFINET communications and, ultimately, PROFINET frame loss.

15.8.10.9.1 Configure a PROFINET IO Controller

The IOs of the PROFINET network must be connected to the variables using a PROFINET IO controller.

The KAS-IDE contains a fully integrated configurator for a PROFINET IO RT controller or device.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The PROFINET maximum data size is 1440 bytes Input length and 1440 bytes Output length.
To calculate the size, use the slot configuration view to count the number of bits per slot and convert to bytes.
Example: 7 slots containing 16 outputs of 32-bits each, is 7 x 16 x 32 = 3584 bits = 448 bytes.

IMPORTANT

Referring to the PROFINET standard, the units of a PROFINET network are named as IO Controllers and IO Devices.

TIP

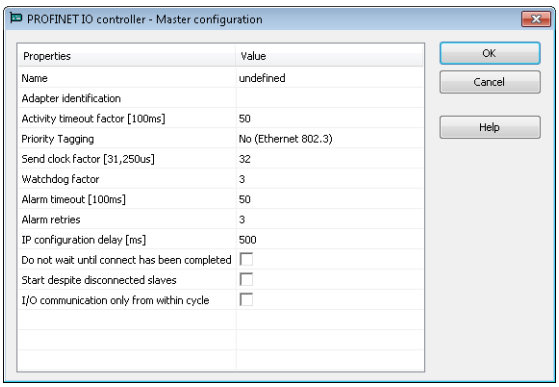
Install the **CD Prot** driver to use additional features of the PROFINET controller fieldbus editor (e.g., Browse Network for Slaves).
The installer is in `<C:\Program Files (x86)\Kollmorgen\Kollmorgen Automation Suite [version number]\Bin\CDProtDriver>`.
The computer must be rebooted after installing the driver.

NOTE

PROFINET is only supported on PCMM2G controllers.

Configuration Procedure

1. Click **Insert > Insert Master/Port** to open the PROFINET IO controller dialog.



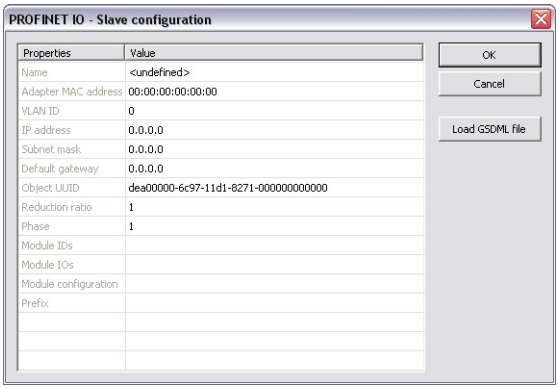
These parameters can be changed:

Parameter	Meaning
Name	• A device name consists of labels. • The labels must follow these conventions: • 1 or more labels, separated by [.]. • Total length is 1 to 240. • Label length is 1 to 63. • Labels consist of [a...z and 0...9]. • Labels do **not** start or end with a hyphen [-]. • The first label does **not** start with **port-xyz** or **port-xyz-abcde** with a,b,c,d,e, x, y, z = 0...9. • Device names do not have the form n.n.n.n, n = 0...999. • Labels only start with **xn-** if RFC 3490 is applied.
Adapter identification	• The PROFINET adapter identification for the controllers are: • PCMM2G: **eth1**. When the application is compiled, based on the controller type, the PROFINET adapter identification is changed to match the required name.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Parameter	Meaning
Activity timeout factor [100ms]	• Timeout for the connection establishment to the devices. • Maximum time between beginning of connection establishment and the first cyclic data exchange. • Timeout factor based on 100ms.
Priority Tagging	• **Yes**: cyclic data exchange without priority tag. • **No**: data exchange with priority tag.
Send clock factor [31,250µs]	• Send the clock factor. • Multiple of 31,250µs (32 = 1ms).
Watchdog factor	Watchdog factor (default 3): • The watchdog factor defines how many frames may be missing until the device is set back. Watchdog triggers: • a) frames may be lost (e.g., due to bad cabling). • b) frames may arrive delayed due to blocking situations in the network (e.g., due to a non-separated network). • The watchdog may be triggered as soon as earlier file transfer occurs between the controller and the IDE when the default value is used. • Increase this value if you encounter frequent watchdog triggers.
Alarm timeout [100ms]	Alarm timeout (default 50).
Alarm retries	Number of alarm retries (default 3).
IP configuration delay [ms]	• IP configuration delay (default 500). • Defines the time to wait whether some devices are not ready after start.
Do not wait until connect has been competed	• Click the **Do not wait until connect has been competed** checkbox to start the application immediately. • Clear the **Do not wait until connect has been competed** checkbox to start the application after all devices are connected.
Start despite disconnected slaves	• Click the **Start despite disconnected slaves** checkbox to start the application despite configured but not found devices. • Clear the **Start despite disconnected slaves** checkbox to start the application if all configured devices were found only.
I/O communication only from within cycle	• Click the **I/O communication only from within cycle** checkbox to run IO communication from within the VM-cycle. • Clear the **I/O communication only from within cycle** checkbox to run IO communication outside the VM-cycle.

2. Mark the controller and click **Insert > Insert Slave/Datablock** to add a PROFINET IO device.



These parameters can be changed:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

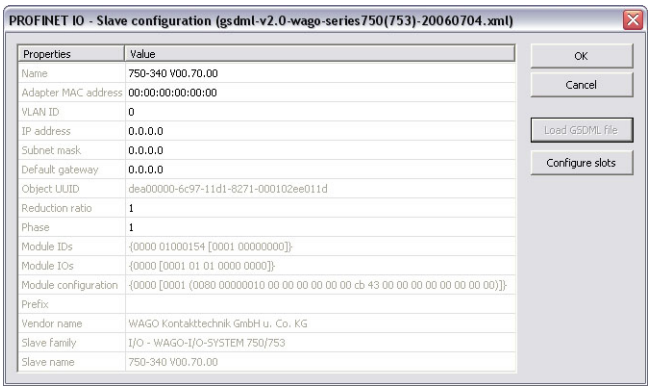
Parameter	Meaning
Name	• A device name consists of labels. • The labels must follow these conventions: • 1 or more labels, separated by [.]. • Total length is 1 to 240. • Label length is 1 to 63. • Labels consist of [a...z and 0...9]. • Labels do **not** start or end with a hyphen [-]. • The first label does **not** start with **port-xyz** or **port-xyz-abcde** with a,b,c,d,e, x, y, z = 0...9. • Device names do not have the form n.n.n.n, n = 0...999. • Labels only start with **xn-** if RFC 3490 is applied.
Adapter MAC address	MAC address of the PROFINET IO device.
VLAN ID	Virtual LAN ID.
IP address	IP address of the device.
Subnet mask	Subnet mask for the IP address of the device.
Default gateway	Default gateway.
Object UUID	UUID of the device.
Reduction ratio	Reduction ratio (default 16): • The Reduction ratio defines the frequency for data to be exchanged with the device. • The transfer rate is calculated by <Send clock factor> * 31,250 µs * <Reduction ratio>. • The Send clock factor is a master parameter with a default value of 32. • All known devices work with this Send clock factor. • Example: • With the default setting of: • 16, the data transfer rate is 48ms. • 2, the data transfer rate is 2ms. • Most devices support Reduction ratio settings of 1, 2, 4, 8, 16, 32, etc.
Phase	Phase
Module IDs	Module IDs of the device modules.
Module IOs	Module IOs of the device modules.
Module configuration	Module configuration.
Prefix	Prefix for the variables.

NOTE

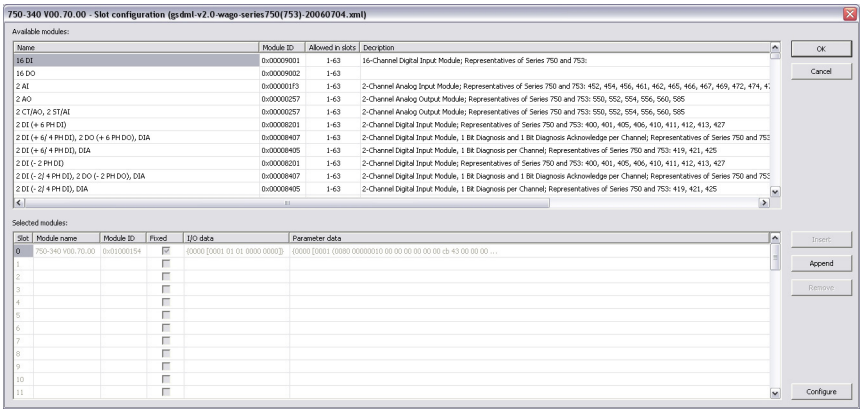
With the calculation of **<Watchdog factor> x <Reduction ratio> x <Send clock factor> x 31,250µs**, you get the time that may expire between two frames until the device is set back.
Example: For the default settings **3 * 32 * 31,250µs** the connection is set back after 3ms of missing frames. For office networks this time is rather low.
It is recommended to us a Reduction ratio of 32 and a Watchdog factor of 24 for such networks.
Example: The connection is reset after 24 x 32 x 32 x 31,250µs = 768ms.
If this data exchange rate is to low, separate the PROFINET IO network from the office network (e.g., by a router).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

3. Click **Load GSDML file** to import the necessary GSDML file.



4. Do the slot configuration after the GSDML file import.



5. Select the modules in the upper list.
With the buttons **Insert** and **Append** the modules are copied to the lower list.

   You can not configure each module.
Only modules with some sub modules respectively with a sub module with parameter data can be configured.
Mark the according module in the lower list and click the **Configure** button.

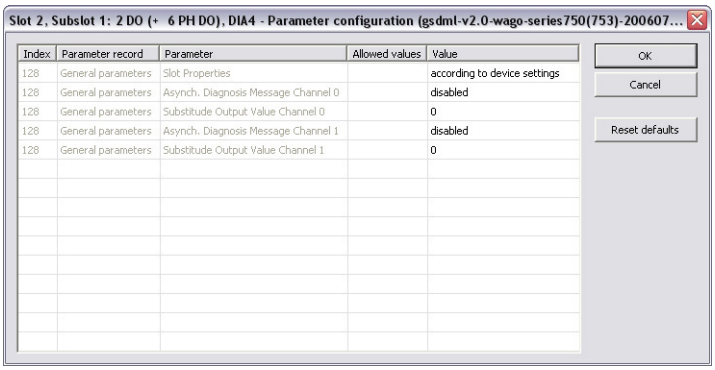
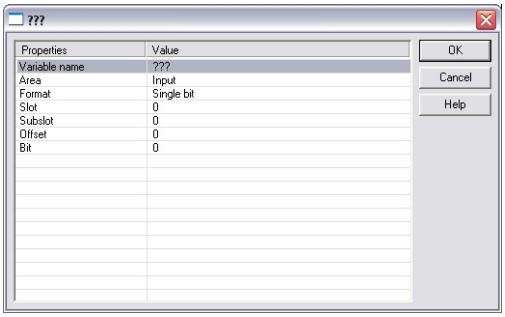


Figure 19-42: Example of configuring sub-modules.

6. Connect the variables with the I/Os.
Use **Insert > Insert/Set Variable** to append a variable to a device.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



These parameters can be changed:

Parameter	Description
Variable name	Variable name following the IEC 61131-3 syntax.
Area	• Output, Output IOCS, Output IOPS. • Input, Input IOCS, Input IOPS. • Device status or PNIO status.
Format	• 32-bit float • Signed: • 8-bit integer • 16-bit integer • 32-bit integer • Single bit • Unsigned: • 8-bit integer • 16-bit integer • 32-bit integer
Slot	Slot Number.
Subslot	Subslot Number.
Offset	Offset
Bit	Bit

NOTE

The offset of a variable is relative to a sub module.
Thus also depending from a slot and subslot.
The offset of the first variable of a sub module is always 0 (zero).

7. All settings can be changed in the grid too.
The information refers to the items of the selected item in the configuration tree.

Variable name	Area	Format	Slot	Subslot	Offset	Bit
_0_Slot14_Subslot1_Output0	Output	Signed 16 bit integer	14	1	0	0
_0_Slot15_Subslot1_Output0	Output	Signed 16 bit integer	15	1	0	0
_0_Slot16_Subslot1_Output0	Output	Signed 16 bit integer	16	1	0	0
_0_Slot17_Subslot1_Output0	Output	Signed 16 bit integer	17	1	0	0
_0_Slot18_Subslot1_Output0	Output	Signed 16 bit integer	18	1	0	0
_0_Slot19_Subslot1_Output0	Output	Signed 16 bit integer	19	1	0	0
_0_Slot20_Subslot1_Output0	Output	Signed 16 bit integer	20	1	0	0
_0_Slot2_Subslot1_Output0	Output	Unsigned 8 bit integer	2	1	0	0
_0_Slot3_Subslot1_Input0	Input	Unsigned 8 bit integer	3	1	0	0
_0_Slot4_Subslot1_Input0	Input	Unsigned 8 bit integer	4	1	0	0
_0_Slot5_Subslot1_Input0	Input	Unsigned 8 bit integer	5	1	0	0
_0_Slot6_Subslot1_Input0	Input	Unsigned 8 bit integer	6	1	0	0
_0_Slot7_Subslot1_Input0	Input	Unsigned 8 bit integer	7	1	0	0
_0_Slot8_Subslot1_Input0	Input	Unsigned 8 bit integer	8	1	0	0
_0_Slot8_Subslot1_Output0	Output	Unsigned 8 bit integer	8	1	0	0
_0_Slot1_Subslot1_Input0	Input	Unsigned 16 bit integer	1	1	0	0
_0_Slot14_Subslot1_Output1	Output	Signed 16 bit integer	14	1	2	0
_0_Slot15_Subslot1_Output1	Output	Signed 16 bit integer	15	1	2	0
_0_Slot16_Subslot1_Output1	Output	Signed 16 bit integer	16	1	2	0
_0_Slot17_Subslot1_Output1	Output	Signed 16 bit integer	17	1	2	0

If the GUI Views is online with a target system, the table shows the real-time data of the variables.

15.8.10.9.2 Configure a PROFINET IO Device

The Runtime manages a mapping table containing the PROFINET IO Inputs and Outputs.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The KAS-IDE contains a fully integrated configurator for a PROFINET IO RT controller or device.

The PROFINET maximum data size is 1440 bytes Input length and 1440 bytes Output length.
To calculate the size, use the slot configuration view to count the number of bits per slot and convert to bytes.
Example: 7 slots containing 16 outputs of 32-bits each, is 7 x 16 x 32 = 3584 bits = 448 bytes.

IMPORTANT

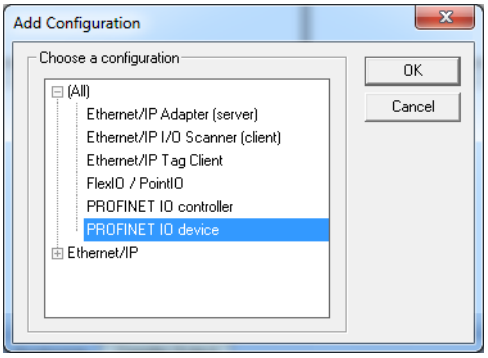
Referring to the PROFINET standard, the units of a PROFINET network are named as IO Controllers and IO Devices.

NOTE

PROFINET is only supported on PCMM2G controllers.

Configuration Procedure

1. Open the fieldbus configuration window.
2. Right-click in the window and select **Insert > Insert Network**.
   The **Add Configuration** dialog opens.



3. Select **PROFINET IO device** and click **OK**.

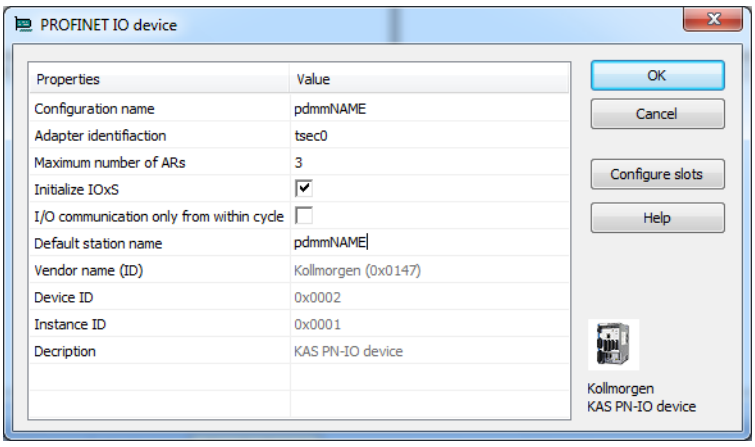
   The configuration is represented as a tree:

   - Profinet IO Configuration
     - Profinet IO device (*)
       - Group (*)
         - Variable (*)

   (*) These items can appear several times in the configuration (depending on the bus topology).

   The I/Os of the PROFINET network must be connected to the variables via a PROFINET IO device.

4. Right-click the PROFINET IO device network and click **Insert Master/Port**.
   This starts the declaration of a PROFINET device.
   The **PROFINET IO device** dialog opens:



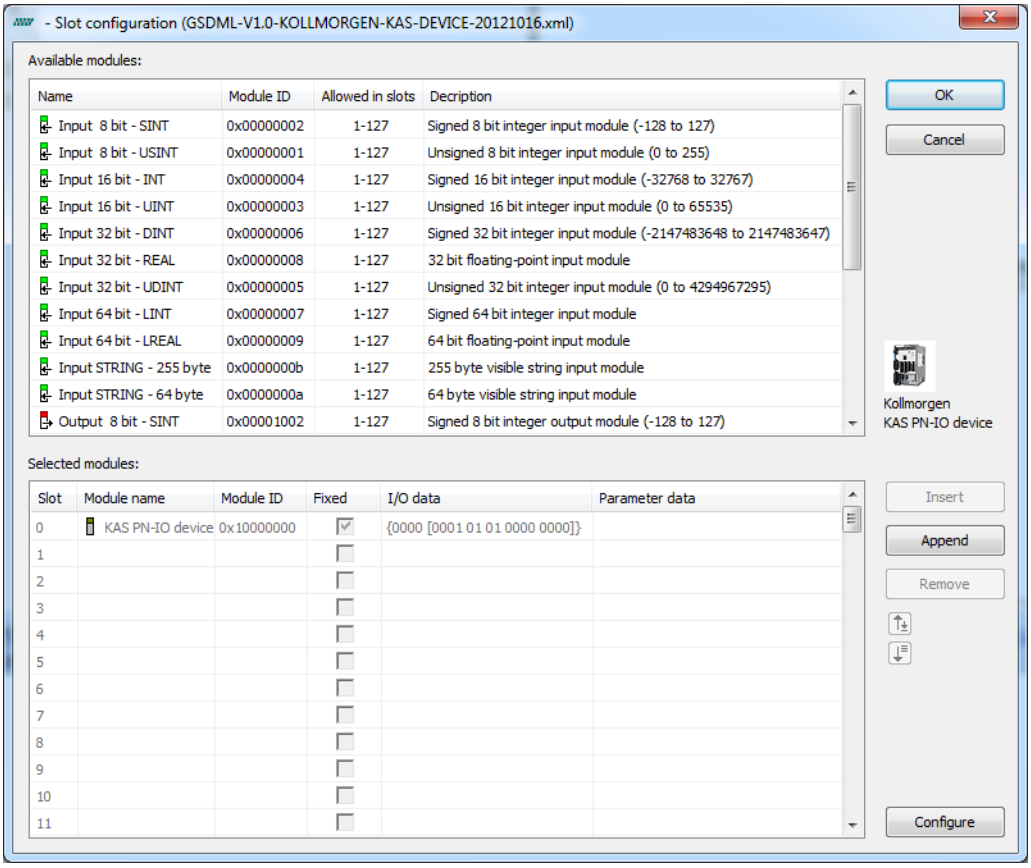
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

These parameters can be changed:

Parameter	Meaning
Configuration Name	• A device name consists of labels. • The labels must follow these conventions: • 1 or more labels, separated by [.]. • Total length is 1 to 240. • Label length is 1 to 63. • Labels consist of [a...z and 0...9]. • Labels do **not** start or end with a hyphen [-]. • The first label does **not** start with **port-xyz** or **port-xyz-abcde** with a,b,c,d,e, x, y, z = 0...9. • Device names do not have the form n.n.n.n, n = 0...999. • Labels only start with **xn-** if RFC 3490 is applied.
Adapter identification	• The PROFINET adapter identification for the controllers are: • PCMM2G: **eth1**. When the application is compiled, based on the controller type, the PROFINET adapter identification is changed to match the required name.
Maximum number of ARs	Maximum number of alarm retries (default 3).
Initialize IOxS	• Click the **Initialize IOxS** checkbox to initialize IOxS with good status. • Clear the **Initialize IOxS** checkbox to stop initialization of IOxS.
I/O communication only from within cycle	• Click the **I/O communication only from within cycle** checkbox to run IO communication from within the VM-cycle. • Clear the **I/O communication only from within cycle** checkbox to run IO communication outside the VM-cycle.
Default station name	Name of the station.

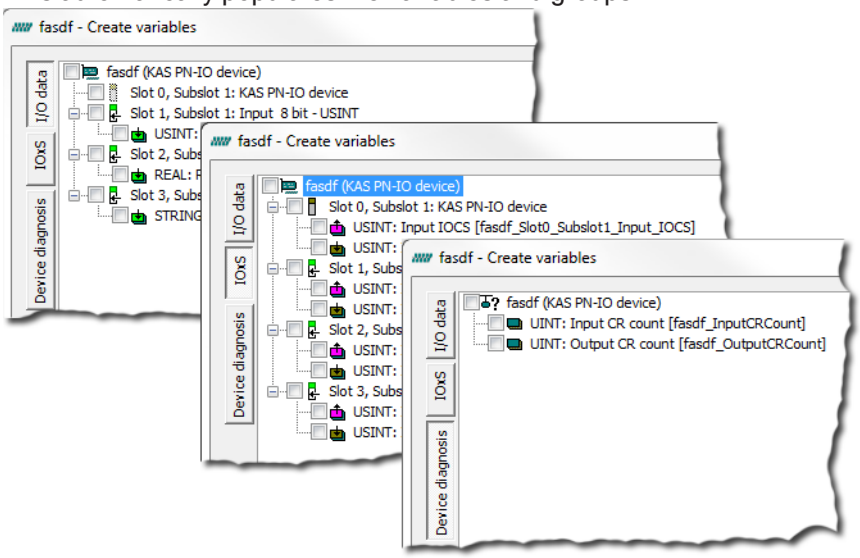
5. Click **OK** to save the changes or selections and close the dialog box.
6. Click the **Configure Slots** button.
   The **Slot Configuration** dialog opens.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



7. Select the modules in the upper list.
   With the buttons **Insert** and **Append** the modules are copied to the lower list.

   You can not configure each module.
   Only modules with some sub modules respectively with a sub module with parameter data can be configured.
   Mark the according module in the lower list and click the **Configure** button.
8. Click **OK** to close the **Slot Configuration** dialog.
9. Click **OK** to close the device window.
10. Click the Insert Variables button.
11. Right click on the master and select Create Variables.
    This automatically populates the variables and groups.



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

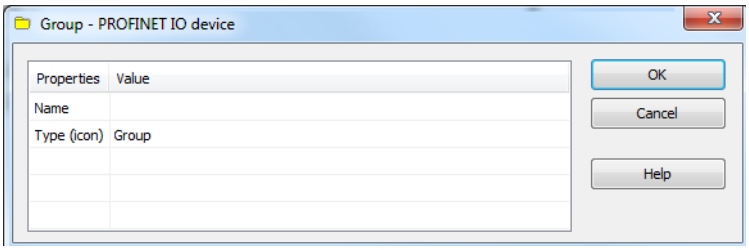
IMPORTANT

Every PROFINET variable is expanded to a set of Boolean variables in PLC by default.
Example: A SINT slot is mapped to eight PLC BOOL variables.
If you have many configured slots, many PLC variables are produced.
The controller is slowed by a large amount of PLC variables.
To avoid this, you can right click on a slot in the **Create variables** dialog and select **Pack bits**.
Example: With a SINT slot, this creates one SINT variable in KAS instead of eight BOOL variables.
This helps reduce the number of PLC variables and reduces the load on the controller.

TIP

The **Pack bits** action may be applied to all slots by right clicking on the root node in the **Create variables** dialog.

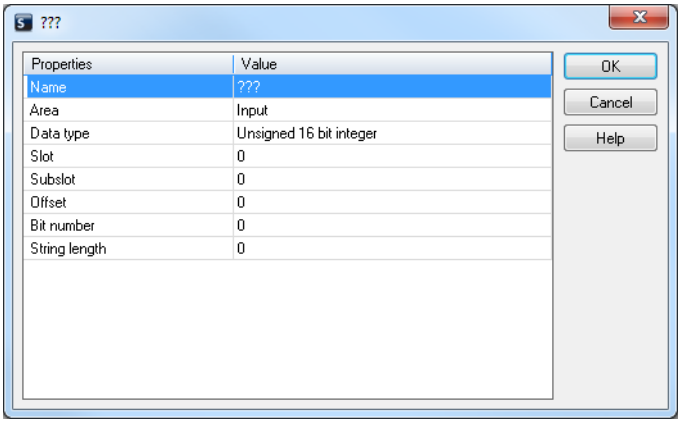
12. Right-click the device and select **Insert Slave** to add a group.



These parameters can be changed:

Parameter	Meaning
Name	Name of the group
Type (icon)	Icon used for the group

13. Now you can connect the variables with the I/Os.
Right-click the group and choose **Insert Symbol** to append a variable to a device.



These parameters can be changed:

Parameter	Description
Variable name	Variable name following the IEC 61131-3 syntax.
Area	• Output, Output IOCS, Output IOPS. • Input, Input IOCS, Input IOPS. • Device status or PNIO status.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Parameter	Description
Format	• 32-bit float • Signed: • 8-bit integer • 16-bit integer • 32-bit integer • Single bit • Unsigned: • 8-bit integer • 16-bit integer • 32-bit integer
Slot	Slot Number.
Subslot	Subslot Number.
Offset	Offset
Bit	Bit

The offset of a variable is relative to a sub module.
Thus also depending from a slot and subslot.
The offset of the first variable of a sub module is always 0 (zero).

- All settings can be changed in the table.
- The information refers to the items below of the selected item in the configuration tree.

Name	Area	Data type	Slot	Subslot	Offset	Bit number
Broadcom_Slot1_Subslot1_Input0_Bit0	Input	Single bit	1	1	0	0
Broadcom_Slot1_Subslot1_Input0_Bit1	Input	Single bit	1	1	0	1
Broadcom_Slot1_Subslot1_Input0_Bit2	Input	Single bit	1	1	0	2
Broadcom_Slot1_Subslot1_Input0_Bit3	Input	Single bit	1	1	0	3
Broadcom_Slot1_Subslot1_Input0_Bit4	Input	Single bit	1	1	0	4
Broadcom_Slot1_Subslot1_Input0_Bit5	Input	Single bit	1	1	0	5
Broadcom_Slot1_Subslot1_Input0_Bit6	Input	Single bit	1	1	0	6
Broadcom_Slot1_Subslot1_Input0_Bit7	Input	Single bit	1	1	0	7
Broadcom_Slot0_Subslot1_Input_IOCS	Input IOCS	Unsigned 8 bit int...	0	1	0	0
Broadcom_Slot0_Subslot1_Input_IOPS	Input IOPS	Unsigned 8 bit int...	0	1	0	0
Broadcom_Slot1_Subslot1_Input_IOCS	Input IOCS	Unsigned 8 bit int...	1	1	0	0
Broadcom_Slot1_Subslot1_Input_IOPS	Input IOPS	Unsigned 8 bit int...	1	1	0	0

- If KAS is connected to a target system and the system is running, the table shows the real-time data of the variables.

15.8.10.9.3 PROFINET - Additional Features

- "All PROFINET Equipment" (➔ p. 920)
  - "Create IOxS for Slave Modules" (➔ p. 920)
  - "Create Variables" (➔ p. 921)
  - "Device Diagnosis" (➔ p. 921)
- "PROFINET Controllers Only" (➔ p. 921)
  - "Browse Network for Slaves" (➔ p. 921)
  - "Device Configuration" (➔ p. 922)
  - "Read Module Configuration" (➔ p. 922)
  - "Set Slave Station Name" (➔ p. 922)
- "PROFINET - Additional Features" (➔ p. 920)
  - "Input Modules or Modules without IO Data" (➔ p. 921)
  - "Output Modules" (➔ p. 921)

15.8.10.9.4.1 All PROFINET Equipment

15.8.11 Create IOxS for Slave Modules

Based on the defined device modules, the referring IOPS and IOCS variables are generated.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.8.11.1 Input Modules or Modules without IO Data

- The IOPS state is managed by the device.
  - This is the status sent by the module.
  - If the data is invalid, the external controller that the PxMM is connected to has to ignore it.
- The IOCS state is managed by the external controller.
  - The external controller can indicate if it can't handle the data.

15.8.11.2 Output Modules

- The IOPS state is managed by the external controller, it reflects the status of the data sent by the external controller.
  - If the data is invalid, the device has to ignore it.
- The IOCS state is managed by the device.
  - The device can indicate to the external controller that the data can't be handled.

15.8.12 Create Variables

Find this command in the context menu of the device.

- Based on the defined device modules the referring variables are generated.
- In the same dialog, it is possible to define the variables for:
  - Device diagnosis
  - IOxS

15.8.13 Device Diagnosis

Based on the PROFINET standard, these referring variables can be generated:

- CycleCounter [UINT].
- DataValid [BOOL].
- ProviderState [BOOL].
- StationsProblemIndikator [BOOL].
- Status [BOOL].

15.8.13.0.1.1 PROFINET Controllers Only

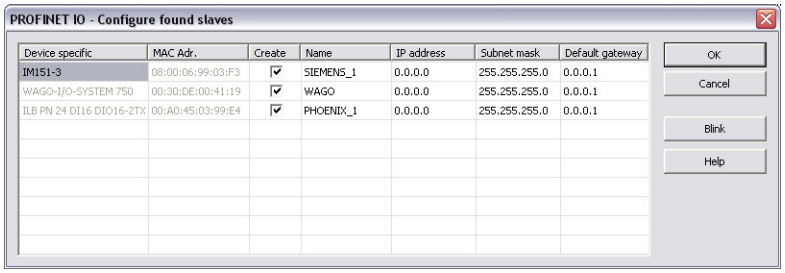
TIP

Install the **CD Prot** driver to use additional features of the PROFINET controller fieldbus editor (e.g., Browse Network for Slaves).
The installer is in `<C:\Program Files (x86)\Kollmorgen\Kollmorgen Automation Suite [version number]\Bin\CDProtDriver>`.
The computer must be rebooted after installing the driver.

15.8.14 Browse Network for Slaves

This command is available in the context menu of the network adapter.

All connected PROFINET IO devices are listed.



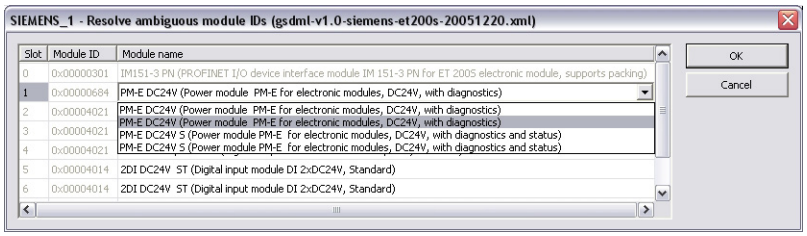
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. In the **Create** column, click the applicable check box to select the devices to add in the tree.
   The Network Link (or another) LED of the marked device blinks for three seconds when the **Blink** button is clicked.
2. In the **Name** column, set the device name.
   - A device name consists of labels.
   - The labels must follow these conventions:
     - 1 or more labels, separated by [.].
     - Total length is 1 to 240.
     - Label length is 1 to 63.
     - Labels consist of [a...z and 0...9].
     - Labels do **not** start or end with a hyphen [-].
     - The first label does **not** start with **port-xyz** or **port-xyz-abcde** with a,b,c,d,e, x, y, z = 0...9.
     - Device names do not have the form n.n.n.n, n = 0...999.
     - Labels only start with **xn-** if RFC 3490 is applied.
3. In the IP Address column, define the address of the device.
4. In the **Default gateway** table cell, click the cell to set the gateway automatically.
5. Click **OK** after setting the IP-parameters (address, subnet mask, default gateway).

15.8.15 Device Configuration

After adding the PROFINET devices to the controller, they need to be configured.

1. Double-click the referring device.
2. Load the GSDML file.
   If ambiguous module IDs are found, they appear in a window.
3. Choose the right module in the combo-box.



NOTE

Automatic creation of variables can be done primal after this step.

15.8.16 Read Module Configuration

Use this menu command to read out the module configuration.

15.8.17 Set Slave Station Name

Use this menu command to rename the slave names.

15.8.17.0.1 PROFINET - Data Types

Connect variables of any data type to the PROFINET I/Os.

- The Runtime converts the values of the I/Os to the type of the variable.
- STRING variables are not supported.

15.8.17.0.2 PROFINET - Error and Status Codes

The PNIO status is a 32-bit unsigned value.

- It is composed of 4 byte values which define the meaning of the status.
- For positive responses, the PNIO status has a value of 0 (zero).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.8.17.0.3.1 Error Codes and Meanings

- "PNIO Status Codes - Negative Responses" (→ p. 923)
- "Error Code - Negative Responses" (→ p. 923)
- "Error Decode - Negative Responses" (→ p. 923)
- "Error Code 1 for Error Decode = 80" (→ p. 923)
- "Error Code 1 for Error Decode = 81" (→ p. 924)
- "Error Code 2 for Error Code 1 = 40" (→ p. 925)
- "Error Code 2 for Error Code 1 <> 40" (→ p. 925)
- "Error Code 2 for Error Code 1 = 01" (→ p. 926)
- "Error Code 2 for Error Code 1 = 02" (→ p. 926)
- "Error Code 2 for Error Code 1 = 03" (→ p. 927)
- "Error Code 2 for Error Code 1 = 04" (→ p. 927)
- "Error Code 2 for Error Code 1 = 8" (→ p. 927)
- "Error Code 2 for Error Code 1 = 16" (→ p. 928)

15.8.17.0.4.2 PNIO Status Codes - Negative Responses

Bit	Meaning
0-7	Error Code 2
8-15	Error Code 1
16-23	Error Decode
24-31	Error Code

15.8.17.0.5.3 Error Code - Negative Responses

Error Code	Meaning
DB	Error in connect response
DC	Error in release response
DD	Error in control response
DF	Error in write response

15.8.17.0.6.4 Error Decode - Negative Responses

Error Decode	Meaning
80	Read/Write service
81	Connect, Control, Release service

15.8.17.0.7.5 Error Code 1 for Error Decode = 80

Error Code 1	Meaning
A1	Write error
A2	Module failure
A3-A6	Reserved
A7	Busy
A8	Version conflict
A9	Feature not supported

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Error Code 1	Meaning
AA-AF	Device specific
B0	Invalid index
B1	Write length error
B2	Invalid slot/subslot
B3	Type conflict
B4	Invalid area/API
B5	State conflict
B6	Access denied
B7	Invalid range
B8	Invalid parameter
B9	Invalid type
BA	Backup
BB-BF	Device specific
C0	Read constrain conflict
C1	Write constrain conflict
C2	Resource busy
C3	Resource unavailable
C4-C7	Reserved
C8-CF	Device specific

15.8.17.0.8.6 Error Code 1 for Error Decode = 81

Error Code 1	Meaning
01	Connect Parameter Error, Faulty ARBlockReq
02	Connect Parameter Error, Faulty IOCRBlockReq
03	Connect Parameter Error, Faulty ExpectedSubmoduleBlockReq
04	Connect Parameter Error, Faulty AlarmCRBlockReq
05	Connect Parameter Error, Faulty PrmServerBlockReq
06	Connect Parameter Error, Faulty MCRBlockReq
07	Connect Parameter Error, Faulty ARRPCBlockReq

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Error Code 1	Meaning
08	Read Write Record Parameter, Error Faulty Record
14	IODControl Parameter Error, Faulty ControlBlockConnect
15	IODControl Parameter Error, Faulty ControlBlockPlug
16	IOXControl Parameter Error , Faulty ControlBlock after a connection establishment
17	IOXControl Parameter Error, Faulty ControlBlock after a plug alarm
28	Release Parameter Error, Faulty ReleaseBlock
40	RMPM (Device state machines, device resources)

15.8.17.0.9.7 Error Code 2 for Error Code 1 = 40

Error Code 2	Meaning
00	ArgsLength invalid
01	Unknown Blocks
02	IOCR Missing
03	Wrong AlarmCRBlock count
04	Out of AR Resources
05	AR UUID unknown
06	State conflict
07	Out of Provider, Consumer, or Alarm Resources
08	Out of Memory
09-FF	Reserved

15.8.17.0.10.8 Error Code 2 for Error Code 1 <> 40

For Error Code 1 <> 40, Error Code 2 refers to a field in the block specified by Error Code 1.

Error Code 2	Meaning
00	Block type
01	Block length
02	Block version (high byte)
03	Block version (low byte)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.8.17.0.11.9 Error Code 2 for Error Code 1 = 01

Error Code 2	Meaning
04	AR Type
05	AR UUID
06	Session key
07	Initiator MAC address
08	Initiator Object UUID
09	AR Properties
0A	Activity timeout factor
0B	UDP RT port
0C	Station name length
0D	Station name

15.8.17.0.12.10 Error Code 2 for Error Code 1 = 02

Error Code 2	Meaning
04	IOCR Type
05	Reference
06	LT
07	IOCR properties
08	IO data length
09	Frame ID
0A	Send clock factor
0B	Reduction ratio
0C	Phase
0D	Sequence
0E	Frame send offset
0F	Watchdog factor
10	Data hold factor
11	Tag header
12	IOCR multicast MAC address
13	Number of APIs
14	API
15	Number of IO data objects
16	Slot
17	Subslot
18	IO data object offset
19	Number of IOCS

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Error Code 2	Meaning
20	Slot
21	Subslot
22	IOCS offset

15.8.17.0.13.11 Error Code 2 for Error Code 1 = 03

Error Code 2	Meaning
04	Number of APIs
05	API
06	Slot
07	Module identification number
08	Module properties
09	Number of sub-modules
0A	Subslot
0B	Sub-module identification number
0C	Sub-module properties
0D	Data description
0E	Data length
0F	IOPS length
10	IOCS length

15.8.17.0.14.12 Error Code 2 for Error Code 1 = 04

Error Code 2	Meaning
04	Type
05	LT
06	AlarmCR Properties
07	RTA timeout factor
08	RTA retries
09	Local alarm reference
0A	Maximum alarm data length
0B	Alarm CRT Tag high
0C	Alarm CRT Tag low

15.8.17.0.15.13 Error Code 2 for Error Code 1 = 8

Error Code 2	Meaning
04	Sequence number
05	AR UUID
06	API

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Error Code 2	Meaning
07	Slot number
08	Subslot number
09	Padding
0A	Index
0B	Data length
0C	Target AR UUID

15.8.17.0.16.14 Error Code 2 for Error Code 1 = 16

Error Code 2	Meaning
05	Padding
06	Session key
07	Padding
08	Control block command
09	Control block properties

15.8.17.0.17 PROFINET - Resolve Errors

NOTE

The errors listed in this topic apply to both controllers and devices except where identified.

- "Additional Analyses - Controllers Only" (➔ p. 928)
- "CL-RPC Lookup - Devices Only" (➔ p. 928)
- "Connect Response Error" (➔ p. 929)
- "Device is not Found" (➔ p. 929)
- "Error Setting the IP Configuration" (➔ p. 929)
- "Module Configuration is Different" (➔ p. 929)
- "Other Errors" (➔ p. 929)
- "Writing Parameterization Error (with status 0xDF80*)" (➔ p. 930)

15.8.17.0.18.1 Additional Analyses - Controllers Only

For further analyses, record the PNIO communication using Wireshark (http://www.wireshark.org/).

1. Start Wireshark.
2. In the **Capture** menu, click **Options**.
3. Choose the network interface used for the PNIO communication and enter this **Capture** filter text:

```
ether proto 0x8892 or (ether proto 0x8100 and ether[16:2]=0x8892) or udp
port 0x8894
```

4. Click **Start**.
5. Start the runtime with your application several times.
6. In the **Capture** menu, click **Stop** to stop the Wireshark capturing.
7. In the **File** menu, click **Save** to save the recorded traffic.

15.8.17.0.19.2 CL-RPC Lookup - Devices Only

15.8.18 Timeout Error

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Verify the IP configuration is valid and appropriate for the network.
- Increase the IP configuration delay.

15.8.18.0.1.1 Connect Response Error

15.8.19 Timeout Error

This Timeout Error sub-section was copied from the Device Resolve Errors topic - is this true for Controllers?

- Verify the IP configuration is valid and appropriate for the network.
- Increase the IP configuration delay.

Depending on the PNIO status error, check if the settings listed in this table are appropriate for your device.

PNIO Status	Check the Specified Setting
1C010003	Slave\ObjectUUID (correct GDML file)
DB8103,*	Slave\Module IDs, IOs
DB81010A	Master\Activity timeout factor
DB81010B	Master\Controller name
DB81010C	Master\Controller name
DB81020A	Master\Send clock factor
DB81020B	Slave\Reduction ratio
DB81020C	Slave\Phase
DB81020F	Master\Watchdog factor
DB810207	Master\RT-Class
DB810210	Master\Watchdog factor
DB810407	Master\Alarm timeout
DB810408	Master\Alarm retries

15.8.19.0.1.1 Device is not Found

- Verify:
  - If the device is switched on.
  - The network connection.
  - The correct name was set on the device.

15.8.19.0.2.2 Error Setting the IP Configuration

- Verify:
  - The IP configuration is valid and appropriate for the network.
  - The VLAN ID is setup correctly in the device settings.

15.8.19.0.3.3 Module Configuration is Different

- Verify:
  - The modules on the device are configured.
    - If the module configuration is read from the device, verify you have selected the correct modules in the **Resolve Ambiguous Modules** dialog that opens after selecting the GSDML file.
  - The correct GSDML file is chosen.
  - The GSDML file version is appropriate for the firmware version of the device.

15.8.19.0.4.4 Other Errors

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Verify:
  - The correct device name is used.
  - The correct GSDML file is chosen.
  - If the GSDML file version is appropriate for the firmware version of the device.

15.8.19.0.5.5 Writing Parameterization Error (with status 0xDF80*)

- Verify:
  - The modules on the device are configured.
    - If the module configuration is read from the device, verify you have selected the correct modules in the **Resolve Ambiguous Modules** dialog that opens after selecting the GSDML file.
  - The correct GSDML file is chosen.
  - The GSDML file version is appropriate for the firmware version of the device.
- If the error could not be resolved using these options, update the firmware of the device and the GSDML file to the latest version and try again.

15.8.20 Use Wireshark

For further analyses, record the PNIO communication using Wireshark (http://www.wireshark.org/).

1. Start Wireshark.
2. In the **Capture** menu, click **Options**.
3. Choose the network interface used for the PNIO communication and enter this **Capture** filter text:

```
ether proto 0x8892 or (ether proto 0x8100 and ether[16:2]=0x8892) or udp
port 0x8894
```

4. Click **Start**.
5. Start the runtime with your application several times.
6. In the **Capture** menu, click **Stop** to stop the Wireshark capturing.
7. In the **File** menu, click **Save** to save the recorded traffic.

15.8.20.1 OPC UA Server

This document describes how to create and configure an Open Platform Communications Unified Architecture (OPC UA) Server on a PCMM2G controller and how to verify communication using various OPC UA client applications.

- This is the successor of the former OPC specifications.
- It is a standardized machine to machine communication protocol for industrial automation.
  - It is developed and maintained by the OPC Foundation.
- When configured, the OPC UA driver enables secure communication using digital certificates and data encryption.

15.8.20.1.1 OPC UA Server Overview

The OPC UA Server is a platform-independent, service-oriented architecture designed for secure and reliable data exchange in industrial automation environments.

The OPC UA Server acts as a central communication hub, enabling seamless interoperability between devices, systems, and applications.

NOTE

Videos of these features are here: PCMM2G OPC UA Server Setup in KAS-IDE.

15.8.20.1.2.1 Key Features

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- **Cross-Platform**: Operates across Windows, Linux, and embedded systems.
- **Data Modeling**: Supports complex data structures and semantic information.
- **Scalability**: Suitable for small devices to enterprise-level systems.
- **Security**: Built-in encryption, authentication, and user access control.
- **Standardized Communication**: Facilitates consistent data exchange across heterogeneous systems.

Typical Use Cases

- Industrial IoT applications
- Machine-to-machine communication
- Remote monitoring and diagnostics
- SCADA and MES integration

15.8.20.1.3.2 OPC UA Server Configuration

See "OPC UA - Fieldbus Configuration" (→ p. 931).

15.8.20.1.4.3 OPC UA Communications using External Software

These topics describe how to establish OPC UA communications using external software tools:

- "OPC UA - Establish a Connection using UaExpert" (→ p. 965).
- "OPC UA - Establish a Connection using Exor HMI" (→ p. 992).

15.8.20.1.5 OPC UA - Fieldbus Configuration

- "OPC UA - Create New Driver Configuration" (→ p. 931)
- "OPC UA - Insert an Endpoint" (→ p. 934)
- "OPC UA - Insert a Group" (→ p. 936)
- "OPC UA - Insert a Variable" (→ p. 938)
- "OPC UA - Add TAG(s)* dialog box" (→ p. 948)

See Also

- "OPC UA - Add Multiple Groups and Variables to the Driver Configuration" (→ p. 941)
- "OPC UA - Move Variables" (→ p. 945)
- "OPC UA - Remove Variables" (→ p. 946)
- "OPC UA - Endpoint dialog box" (→ p. 953)

15.8.20.1.6.1 OPC UA - Create New Driver Configuration

1. Start the KAS-IDE.
2. Create a new project or open an existing KAS project.
3. In the Project Explorer, double-click the Fieldbus node to open the Fieldbus Editor.
4. On the vertical toolbar, click the **Insert Network** (Configuration) button (). (Figure 19-43)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

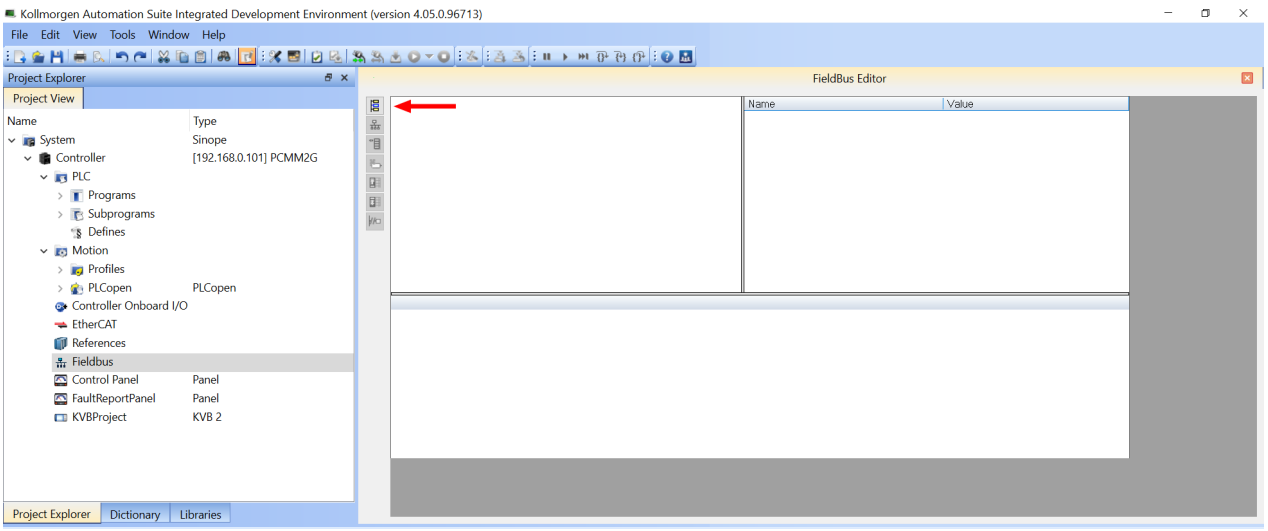


Figure 19-43: Insert Network (Configuration) button

The **Add Configuration** dialog box opens.

5. In the **Select a configuration** tree, select **OPC UA Server**. (Figure 19-44)

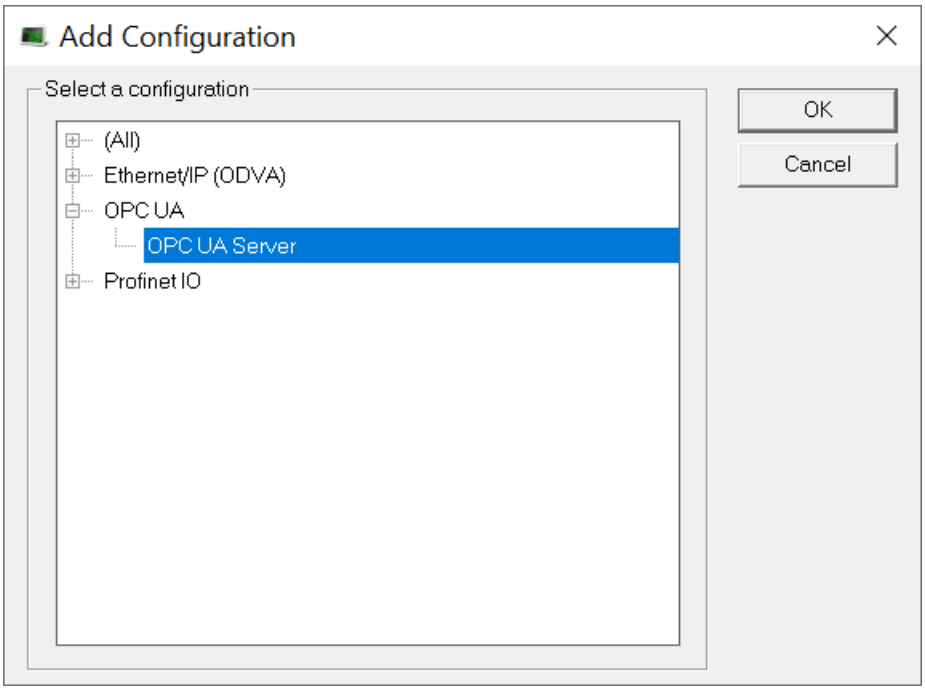


Figure 19-44: Add Configuration dialog box - OPC UA Server selected

6. Click **OK** to save the changes or selections and close the dialog box.
The "OPC UA - Server dialog box" (➔ p. 960) opens. (Figure 19-45)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

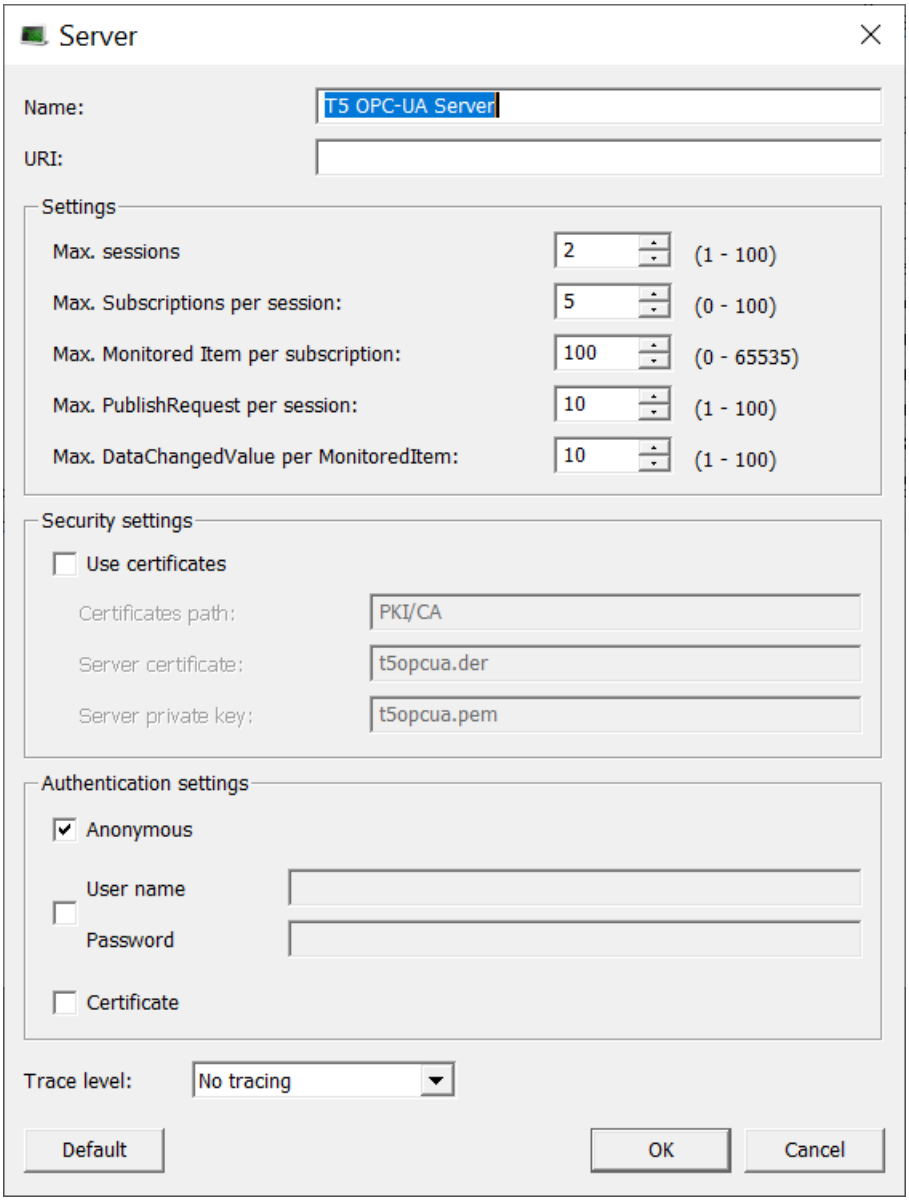


Figure 19-45: Server dialog box

7. In the **Name** text box, enter a name for this specific server configuration.
8. In the **URI** text box, enter the URL used for the certificate authentication.

NOTE

Leave this text box blank to use the PCMM2G assigned default URL.

9. Optional: In the **Settings** section, accept the defaults or:
   - In the **Max. Sessions** spin box, enter the maximum number of simultaneously connected OPC UA client sessions.
   - In the **Max. Subscriptions per session** spin box, enter the maximum number of subscription groups a client can create per session.
   - In the **Max. Monitored Item per subscription** spin box, enter the maximum number of items that can be monitored within a single subscription.
   - In the **Max. Publish Request per session** spin box, enter the maximum number of active publish requests per session.
   - In the **Max. Data Changed Value per Monitored Item** spin box, enter the maximum number of value changes stored in memory for each Monitored Item.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

10. Optional: In the **Security settings** section, click the **Use certificates** check box to use digital certificates for data encryption between the server and client.
When selected, the text boxes of this section are activated.

TIP

Kollmorgen recommends accepting the defaults in the **Security settings** section to use self-signed certificates.

- In the **Certificates path** text box, enter the relative directory path where the certificates are stored.
- In the **Server certificate** text box, enter the filename of the server certificate used for encryption.
- In the **Server private key** text box, enter the filename of the server private key used for encryption.

IMPORTANT

- When the **Use Certificates** option is enabled, an SSL server certificate must be installed on the PCMM2G controller.
  - The PCMM2G supports self-signed certificates only for secure communication with external OPC UA client applications.
  - For instructions on generating a self-signed server certificate, see:
    - Certificates tab - Create SSL Server Certificate (Self-Signed).
- To establish a successful OPC UA connection, the SSL client certificates of each OPC UA client application must also be uploaded to the PCMM2G controller.
  - See Certificates tab - Upload SSL Client Certificate.

11. In the **Authentication settings** section:
    - Click the **Anonymous** check box to use non-encrypted communications.
    - Click the **User name/Password** check box to activate the **User name** and **Password** text boxes.
    - Click the **Certificate** check box so the server allows the client to authenticate with a certificate.
      - The **Certificate** check box is used in conjunction with the "OPC UA - Server dialog box" (➔ p. 960), **Security settings** section - **Use certificates** check box.
12. Click the **Trace Level** list box arrow and select the log level for OPC UA Server Runtime Service messages written to the controller log output.
See Log Messages tabs.
13. Click **OK** to save the changes or selections and close the dialog box.
The Fieldbus Editor returns with the added OPC UA Server node. (Figure 19-46)

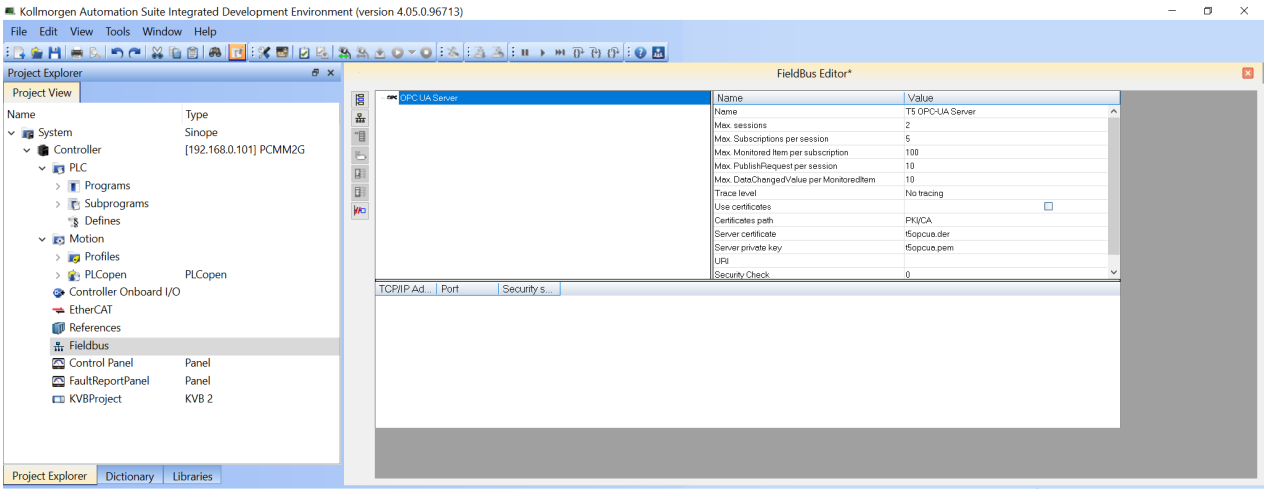


Figure 19-46: Fieldbus Editor with the added OPC UA Server node

14. Continue with "OPC UA - Insert an Endpoint" (➔ p. 934).

15.8.20.1.7.2 OPC UA - Insert an Endpoint

1. In the Fieldbus Editor tree, select **OPC UA Server**.
The **Insert Master/Port** button in the vertical toolbar is activated. (Figure 19-47)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

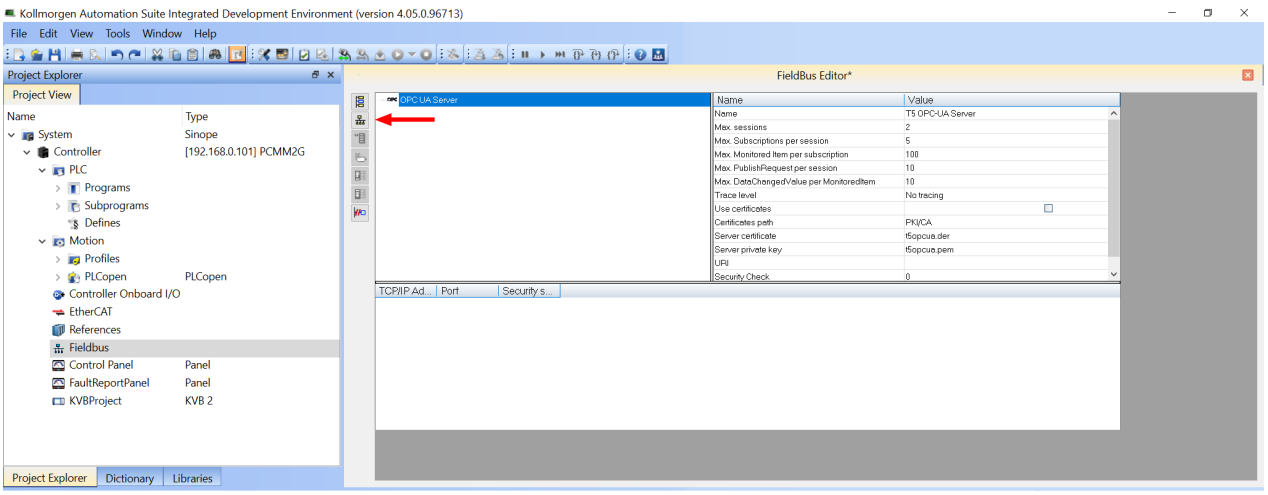


Figure 19-47: Insert Master/Port button

2. On the vertical toolbar, click the **Insert Master/Port** button ( ).
The "OPC UA - Endpoint dialog box" (→ p. 953) opens. (Figure 19-48)

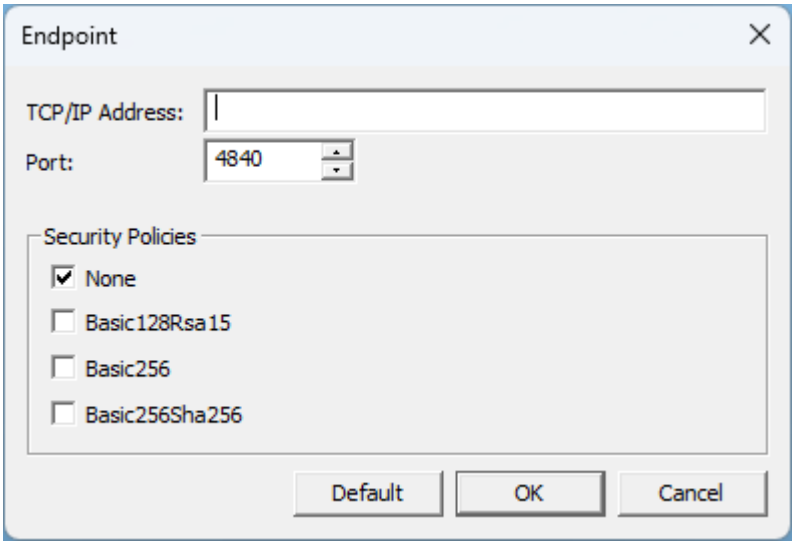


Figure 19-48: Endpoint dialog box

3. Optional: In the **TCP/IP Address** text box, enter the TCP/IP address of the KAS controller (e.g., PCMM2G).

TIP

Leave this text box blank to use the KAS controller assigned IP address.

4. In the **Port** spin box, enter the Listening Port opened by the OPC UA Server.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

5. In the Security Policies section, select the applicable policy for the application.

IMPORTANT

- If using certificates in the "OPC UA - Server dialog box" (➔ p. 960), **Security settings** section, at least one of the Basic128Rsa15, Basic256, or Basic256Sha256 check boxes must be selected.
- If the **Anonymous** check box is selected in the "OPC UA - Server dialog box" (➔ p. 960), **Authentication settings** section, the **None** check box must be selected.

- Click the **None** check box so that no security settings are used.
- Click the **Basic128Rsa15** check box to use both the Basic128 and RSA15 security policies.
- Click the **Basic256** or **Basic256Sha256** check box to use the 256-bit Basic as the message encryption algorithm.
- Click the **Basic256Sha256** check box to use both the Basic256 and SHA256 security policies.

6. Click **OK** to save the changes or selections and close the dialog box.
The endpoint is now available in the OPC UA Server Fieldbus Editor. (Figure 19-49)

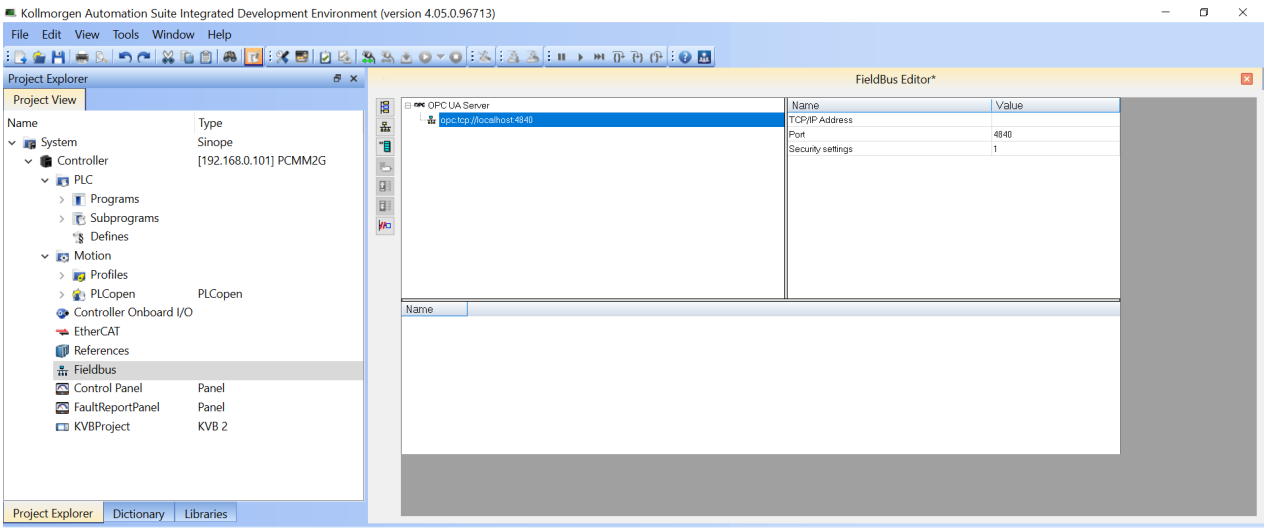


Figure 19-49: Fieldbus Editor with the added endpoint

7. Continue with "OPC UA - Insert a Group" (➔ p. 936).

15.8.20.1.8.3 OPC UA - Insert a Group

1. In the Fieldbus Editor tree, select an **OPC UA Server** endpoint.
The **Insert Slave/Data Block** button in the vertical toolbar is activated. (Figure 19-50)

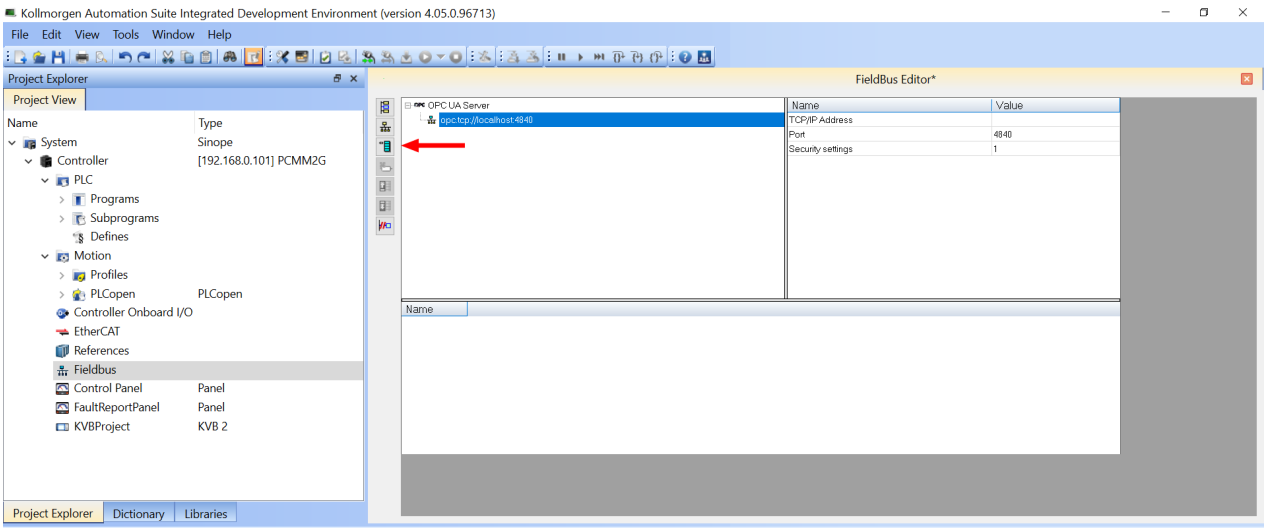


Figure 19-50: Insert Slave/Data Block button

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

2. On the vertical toolbar, click the **Insert Slave/Data Block** button ( ).
The "OPC UA - Group: dialog box" (→ p. 955) opens. (Figure 19-51)

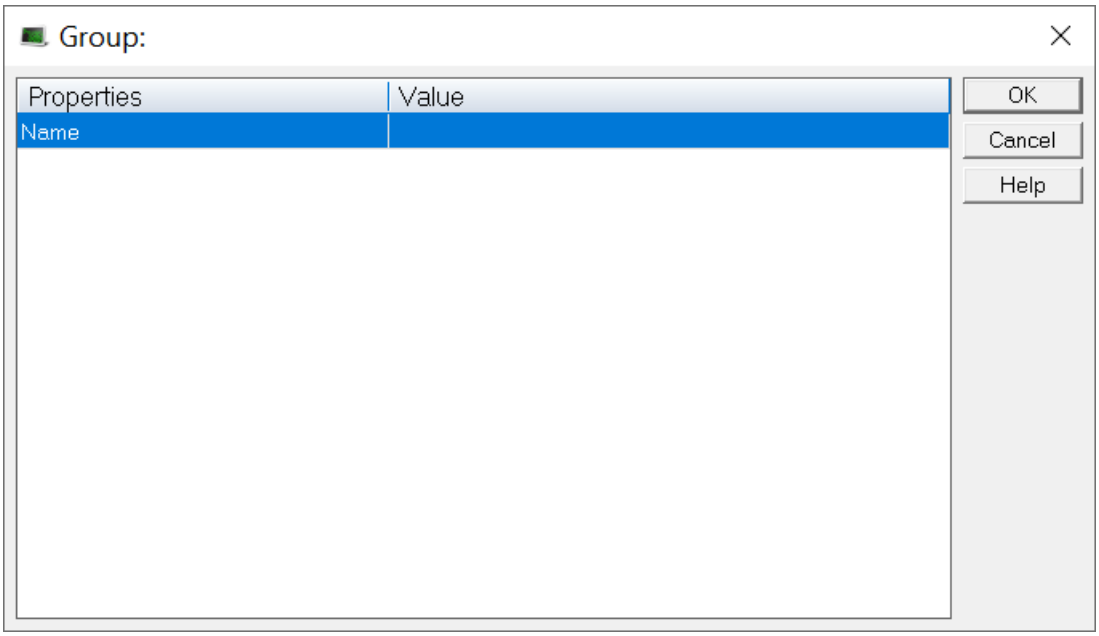


Figure 19-51: Group: dialog box

The **Group:** dialog box is used to create an individual group.

NOTE

- The Group: dialog box can only create a single group.
- See "OPC UA - Add Multiple Groups and Variables to the Driver Configuration" (→ p. 941).

3. In the **Name** row, double click the **Value** column cell.
The **Name - Value** text box appears.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4. Enter the name for the new group. (Figure 19-52)

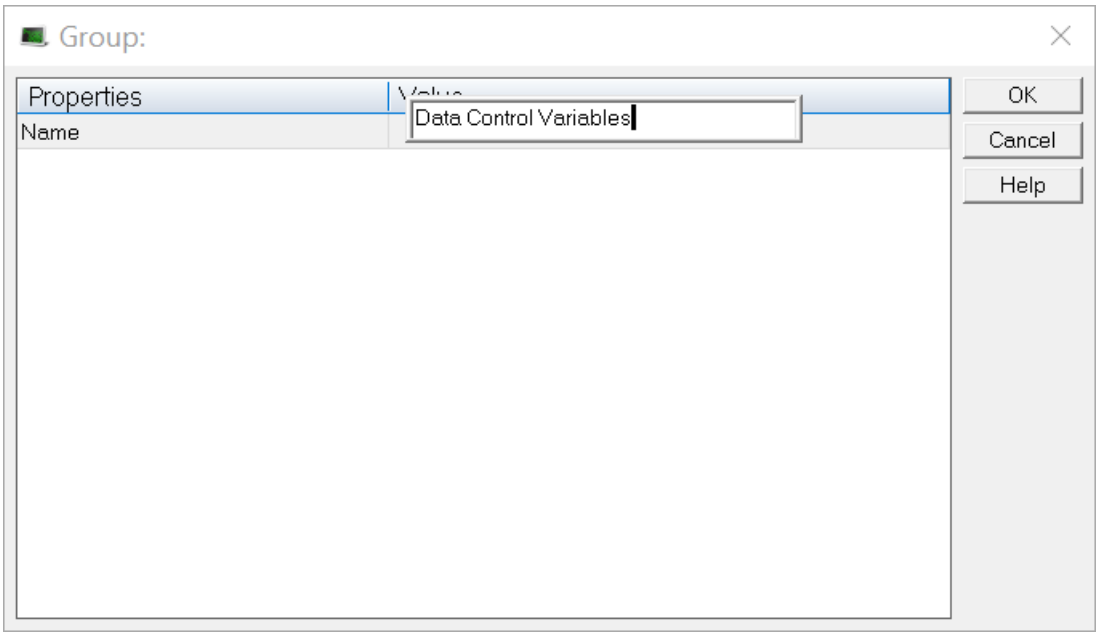


Figure 19-52: Name - Value text box

5. Press <Enter> to save the information and close the text box.
6. Click **OK** to save the changes or selections and close the dialog box.
   The new data group is added under the OPC UA Server endpoint in the Fieldbus Editor. (Figure 19-53)

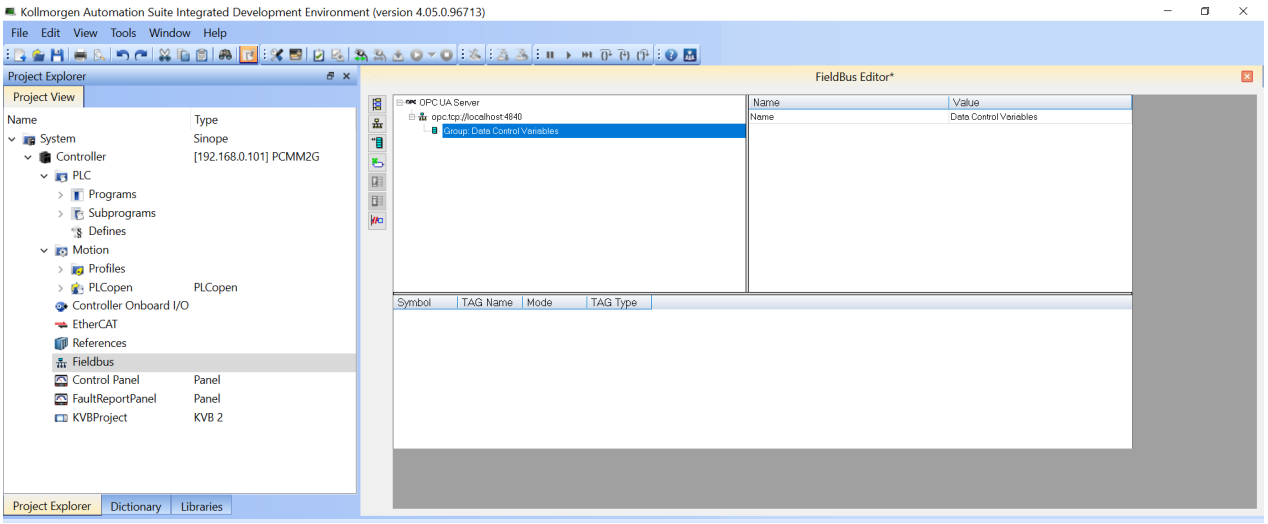


Figure 19-53: Fieldbus Editor with the added group

7. Continue with "OPC UA - Insert a Variable" (➔ p. 938).

15.8.20.1.9.4 OPC UA - Insert a Variable

1. In the Fieldbus Editor tree, select the **OPC UA Server** data group to assign a new variable to.
   The **Insert Variable** button in the vertical toolbar is activated. (Figure 19-54)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

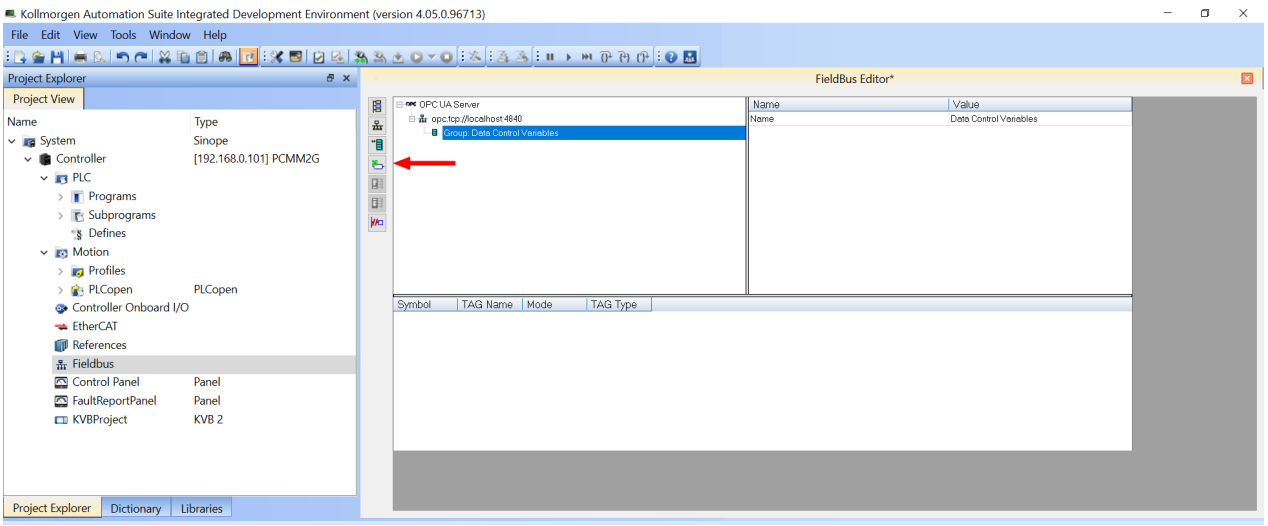


Figure 19-54: Insert Variable button

2. On the vertical toolbar, click the **Insert Variable** button ( ).
The "OPC UA - Insert Variable dialog box" (➔ p. 957) opens. (Figure 19-55)

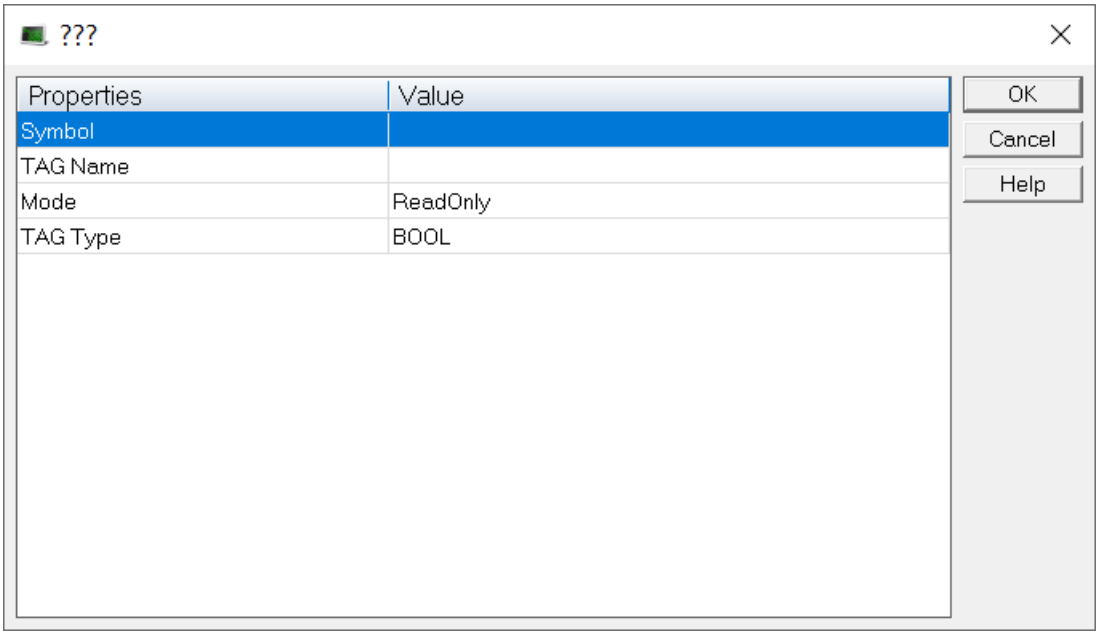


Figure 19-55: Insert Variable (???) dialog box

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

3. In the Symbol row, double-click the **Value** column cell to open the **Variables** combo-box. (Figure 19-56)

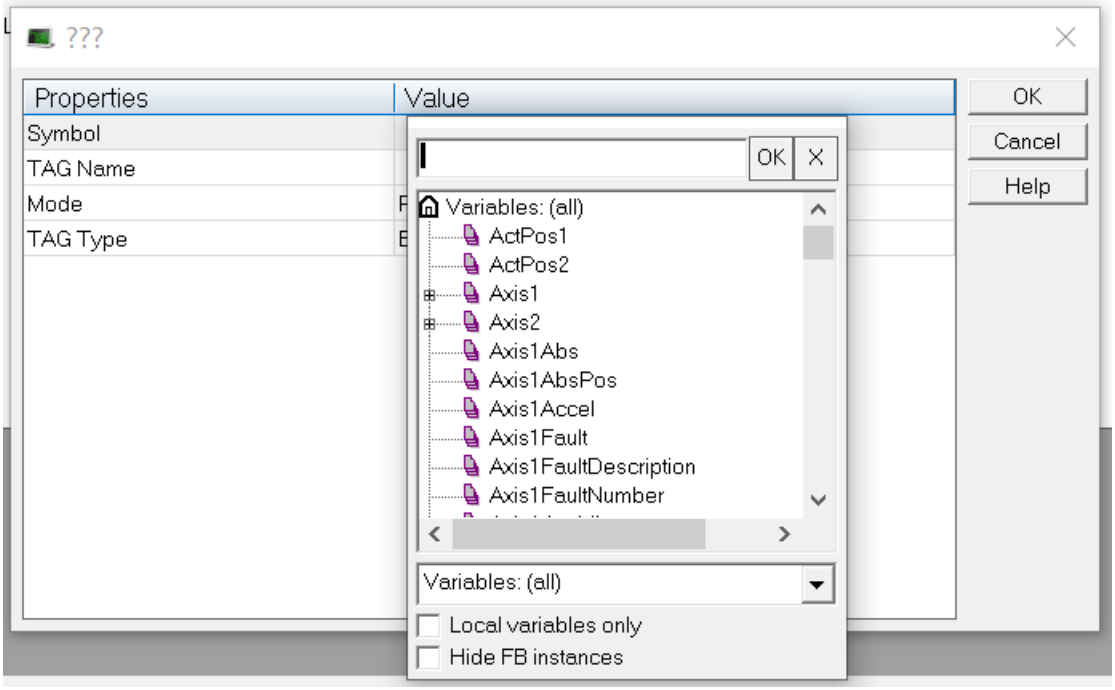


Figure 19-56: Variables combo-box

4. In the **Variables** combo-box, select the applicable variable.

NOTE

Only a single variable can be selected in the combo-box.

5. Click **OK** to save the selection and close the combo-box.
   The variable is added to the **Value** column cell.
6. In the **TAG Name Value** column cell, double-click the cell to open the **TAG Name - Value** text box.
7. In the **TAG Name - Value** text box, enter the name used by the OPC UA client to address the variable.
8. Press <Enter> to save the information and close the text box.
9. In the **Mode - Value** column cell, double-click the cell to open the **Mode - Value** list box.
10. Double-click an option in the **Mode - Value** list box.
    The **Insert Variables** dialog box returns showing the selection.
11. In the **TAG Type Value** column cell, double-click the cell to open the **TAG Type - Value** list box.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

12. Double-click an option in the **TAG Type - Value** list box.
The **Insert Variables** dialog box returns showing the selection. (Figure 19-57)

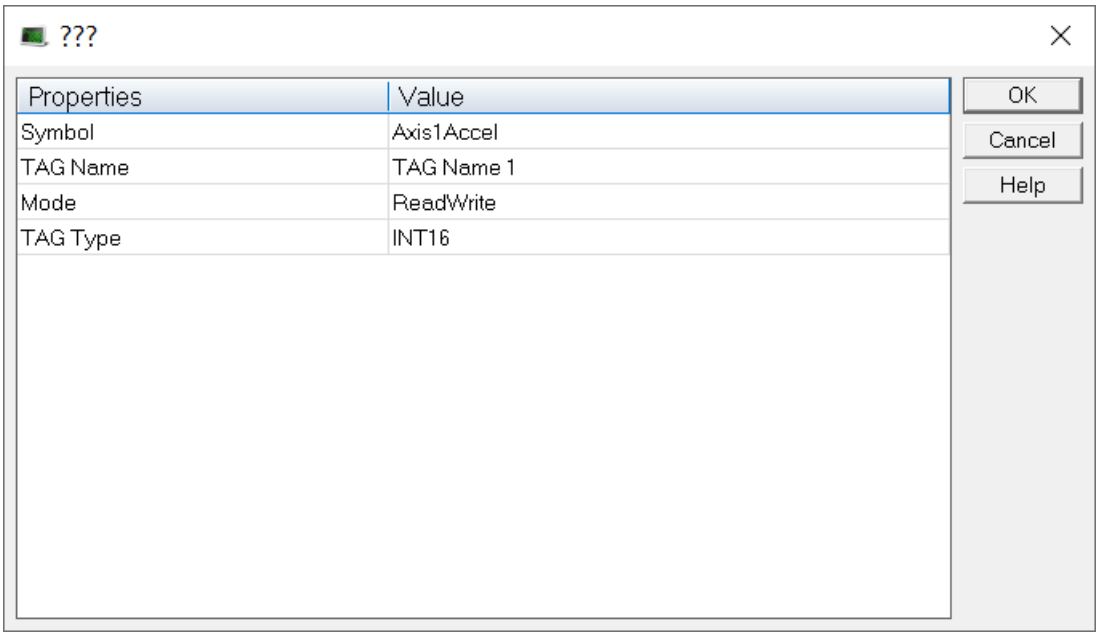


Figure 19-57: Example: Completed Insert Variables dialog box

13. Click **OK** to save the changes or selections and close the dialog box.
The new variable is added to the data group. (Figure 19-58)

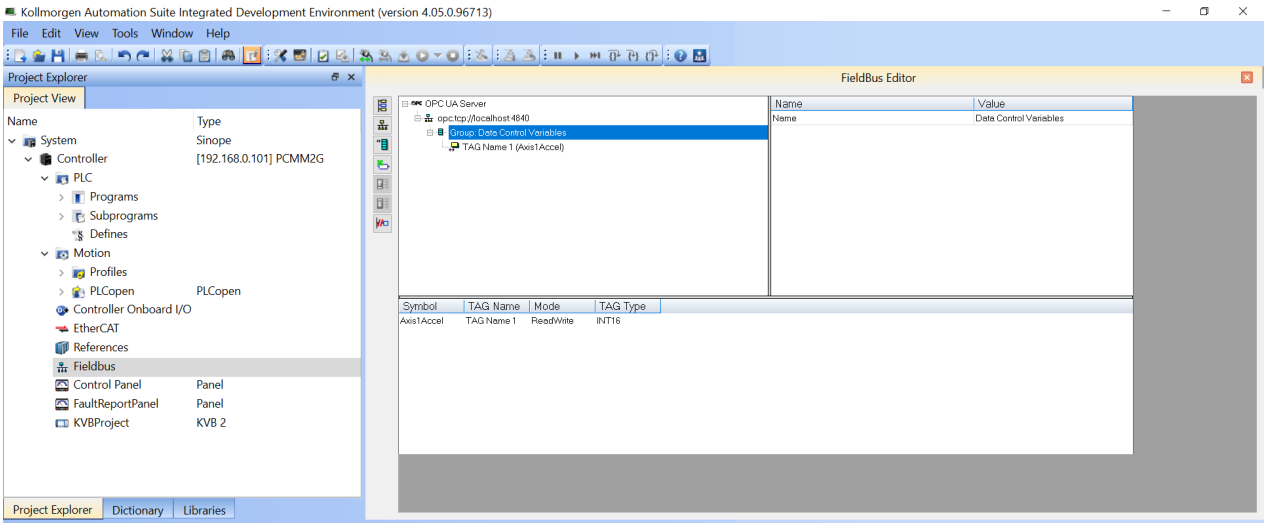


Figure 19-58: Fieldbus Editor with added variable

14. Optional:
    - Repeat Steps 2 to 13 to add more variables.
    - Continue with "OPC UA - Add Multiple Groups and Variables to the Driver Configuration" (➔ p. 941).

⚠WARNING

- **Do not** manually add variables to be exchanged (e.g., Mode = RO, WO, RW)!
- Use the "OPC UA - Add Multiple Groups and Variables to the Driver Configuration" (➔ p. 941) procedure instead.

See Also

"OPC UA - Add TAG(s)* dialog box" (➔ p. 948)

15.8.20.1.10.5 OPC UA - Add Multiple Groups and Variables to the Driver Configuration

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. In the **OPC UA Server** tree, right-click a Master node and select the **Add TAG(s)** option. (Figure 19-59)

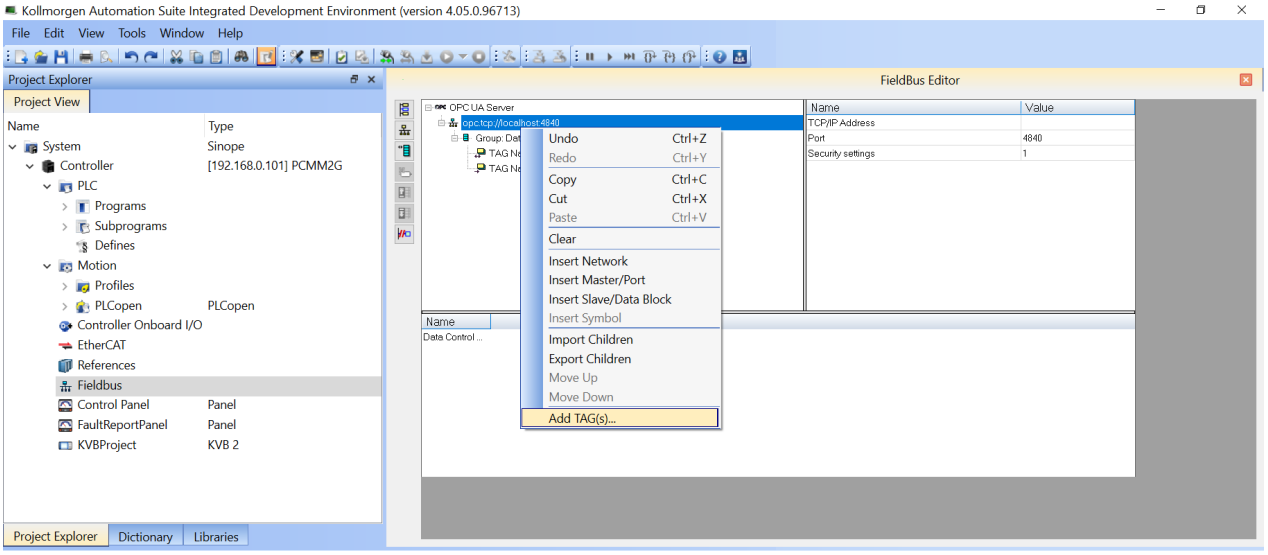


Figure 19-59: Add TAG(s) option

The "OPC UA - Add TAG(s)* dialog box" (→ p. 948) opens. (Figure 19-60)

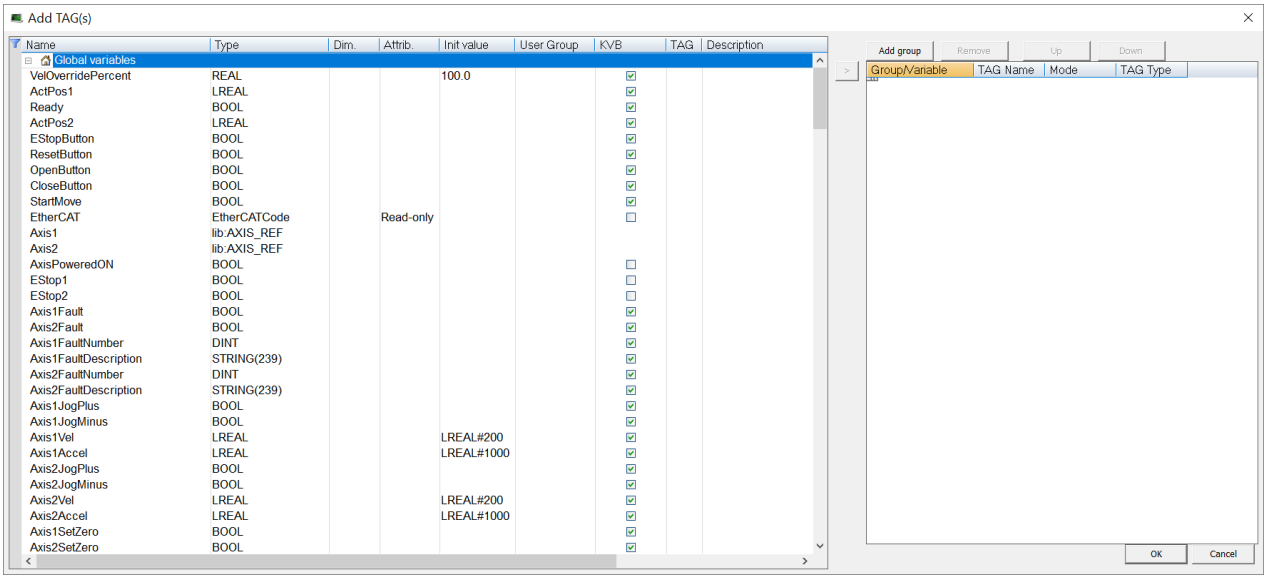


Figure 19-60: Add TAG(s) dialog box

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

2. Add a new group:
   a. Click the **Add** group button to add a new group row in the **Group/Variable** table.
   b. In the **Group/Variable** column, double-click the column cell.
   The **Group/Variable** text box appears. (Figure 19-61)

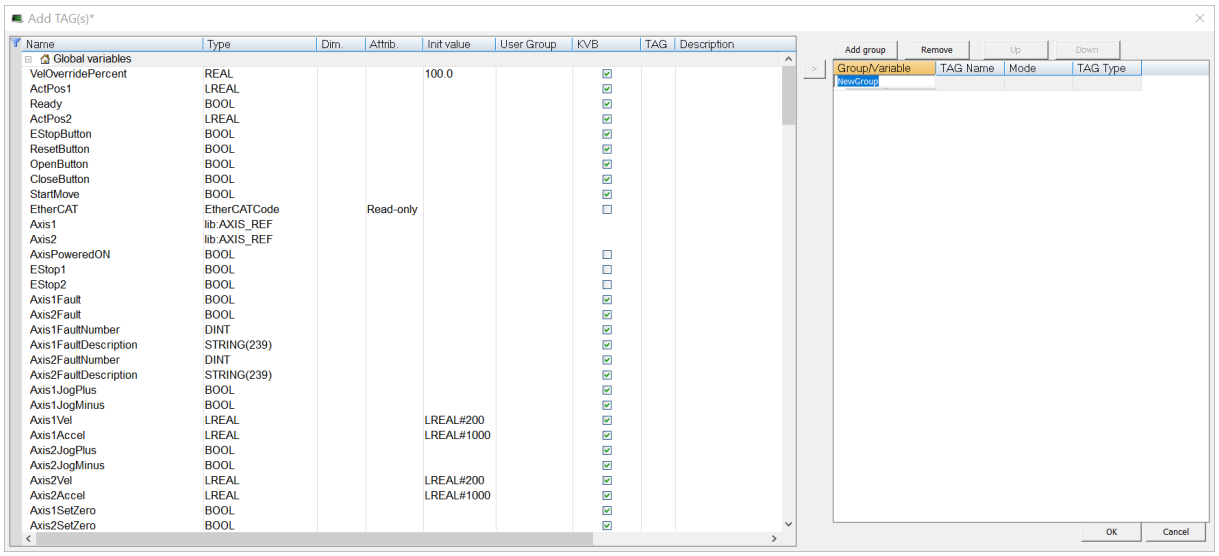


Figure 19-61: Group/Variable text box

   c. In the **Group/Variable** text box, enter a name for the new group.
   d. Press <Enter> to save the information and close the text box.
3. Add variables to a group:
   a. In the **Group/Variable** table, select a group to add a variable to.
   b. In the **Available Variables** table, select a variable and click the > button. (Figure 19-62)

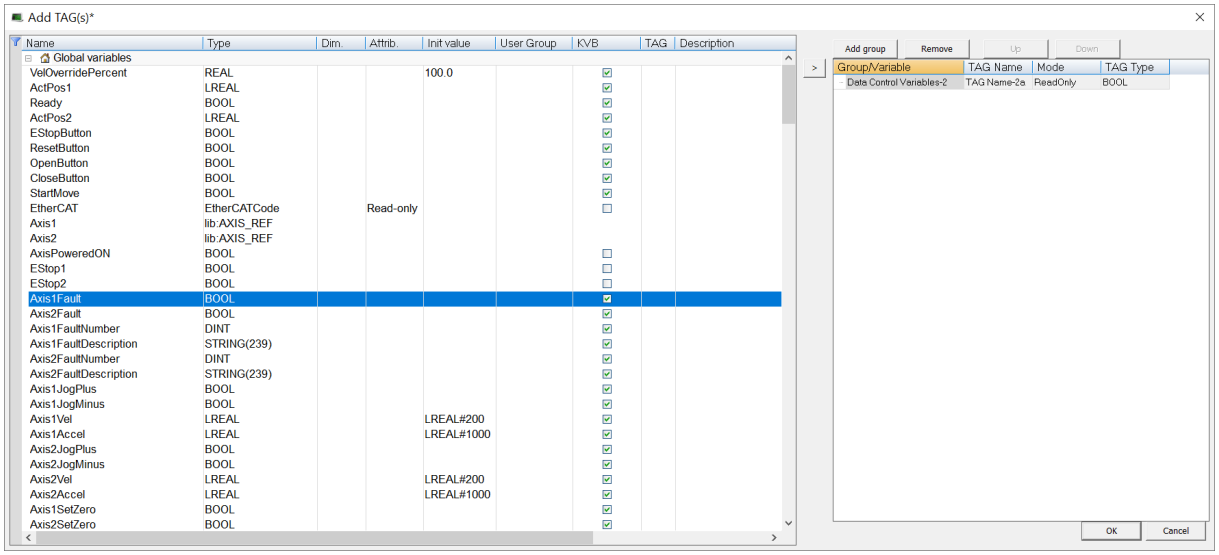


Figure 19-62: Selected variable

The variable is copied to the **Group/Variable** table. (Figure 19-63)

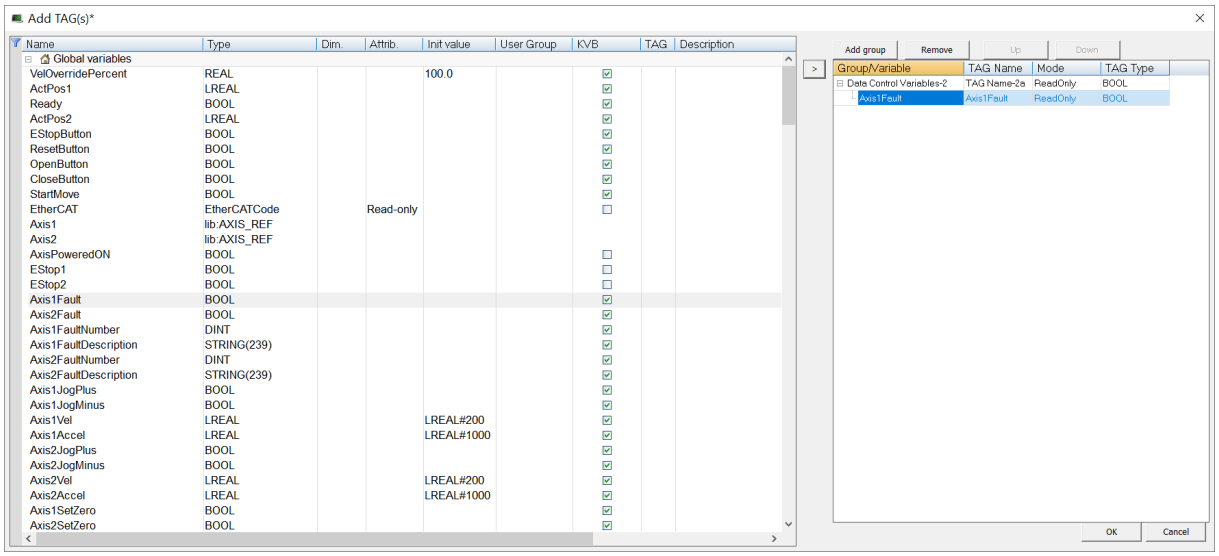


Figure 19-63: Variable added to group

c. Repeat Steps 3a and 3b to add more variables to the group.

TIP

In the Variable Dictionary table, select a variable and drag it under the designated group.

4. Repeat Steps 2 and 3 to add more groups and their variables to the application.
5. Click **OK** to save the changes or selections and close the dialog box.
   A confirmation message appears. (Figure 19-64)

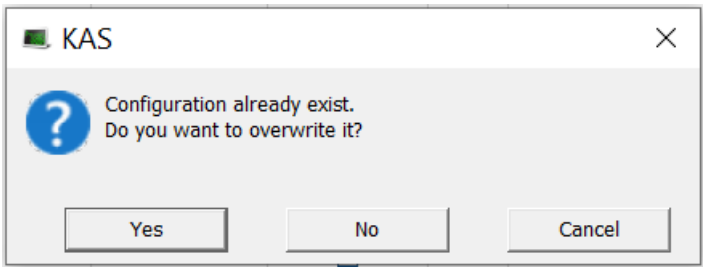


Figure 19-64: Confirmation message

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

6. Click **Yes** to continue.
   The new groups are now available in the Fieldbus Editor. (Figure 19-65)
   The TAG types of the variables are automatically set to the correct type.

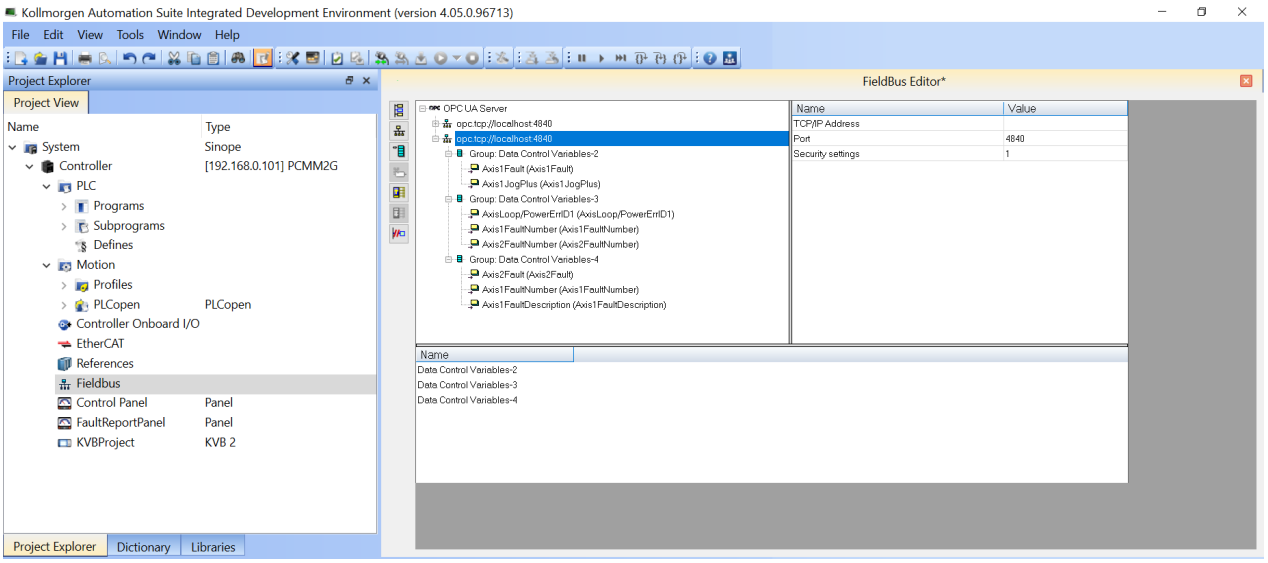


Figure 19-65: Multiple groups in the Fieldbus Editor

⚠WARNING

- Using Drag and Drop functionality to add variables from the Variable Dictionary table to the driver configuration may result in wrong TAG type assignments.
- The KAS-IDE generates errors during project compilation if mismatches between variable Data Type and OPC UA TAG type for single data points are detected.
- This ensures only valid configurations are applied to the target system.

See Also

- "OPC UA - Move Variables" (➔ p. 945)
- "OPC UA - Remove Variables" (➔ p. 946)

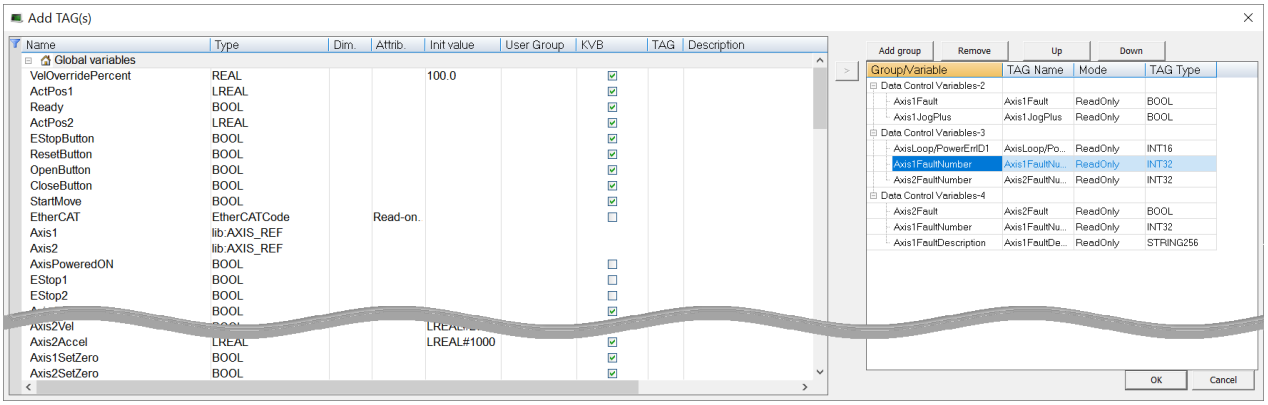
15.8.20.1.11.6 OPC UA - Move Variables

Use this procedure to move:

- Selected groups up or down.
- Selected variables in the current group up or down.

Procedure

1. In the "OPC UA - Add TAG(s)* dialog box" (➔ p. 948), in the **Group/Variable** table, select a variable. (Figure 19-66)



See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Figure 19-66: Selected variable

2. Either:
   - Click the **Up** button to move the selected item in the **Group/Variable** table up.
   - Click the **Down** button to move the selected item in the **Group/Variable** table down.

   The variable moves in the designated direction. (Figure 19-67)

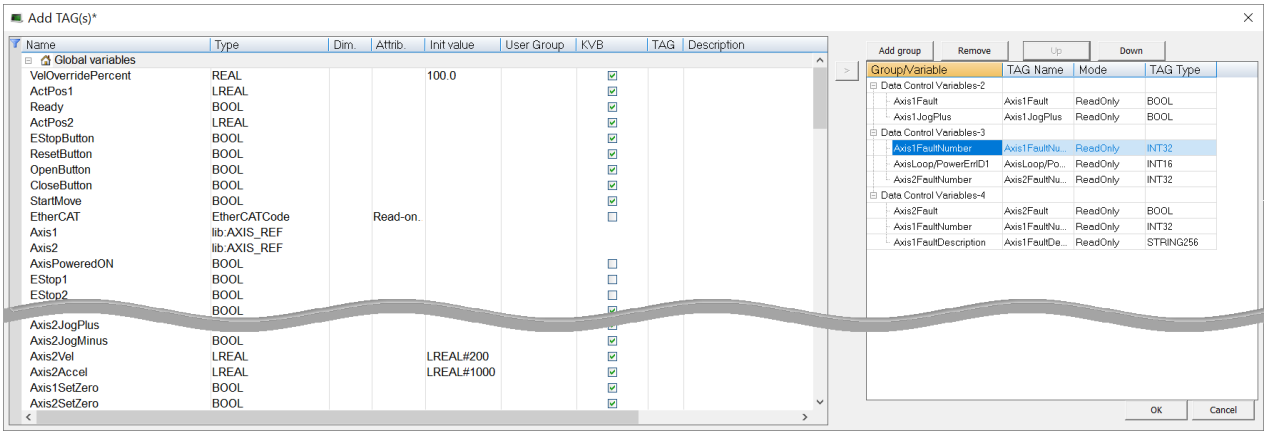


Figure 19-67: Variable moved up

3. Click **OK** to save the changes or selections and close the dialog box.
   A confirmation message appears.
4. Click **OK** to continue.
   The edited groups or variables are updated in the Fieldbus Editor.

15.8.20.1.12.7 OPC UA - Remove Variables

1. In the "OPC UA - Add TAG(s)* dialog box" (➔ p. 948), in the **Group/Variable** table, select a variable. (Figure 19-68)

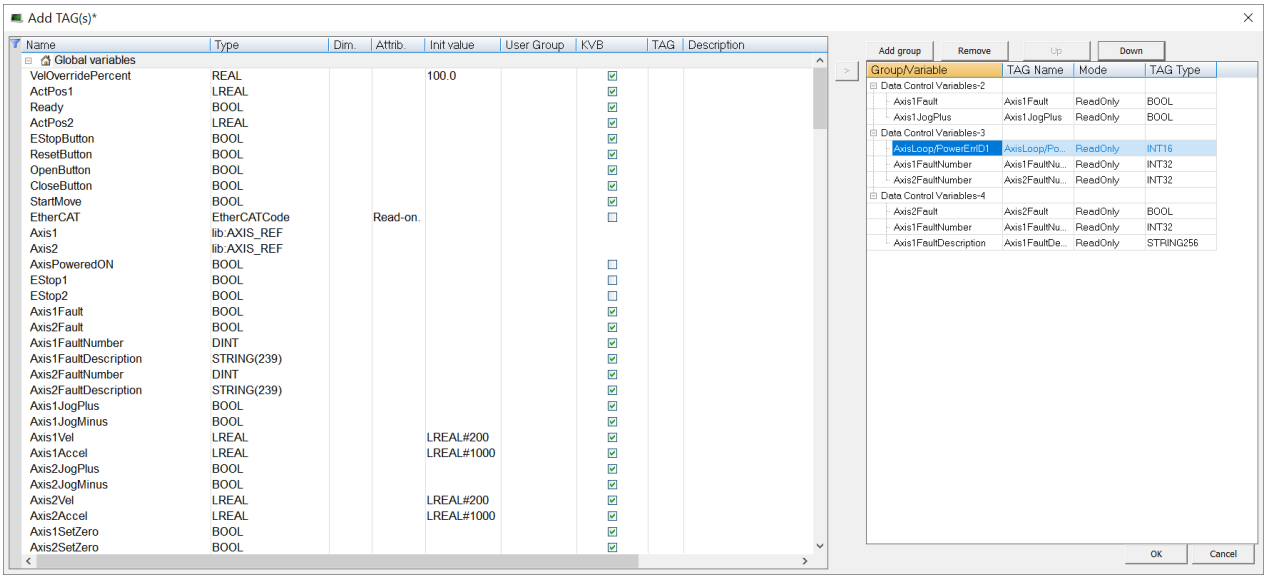


Figure 19-68: Selected variable to remove

2. Click the **Remove** button to immediately delete the selected variable from the **Group/Variable** table. (Figure 19-69)
   There is no undo option to restore a deleted variable.
   Click **Cancel** to discard all changes or selections and close the dialog box.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

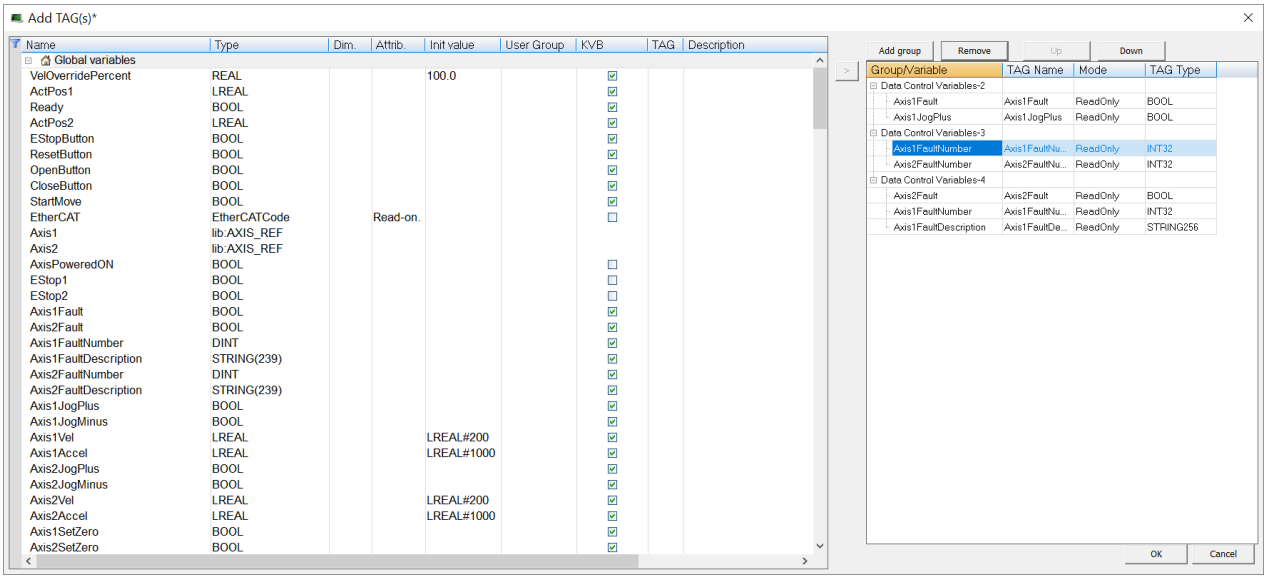


Figure 19-69: Removed variable

3. Click **OK** to save the changes or selections and close the dialog box.
   A confirmation message appears.
4. Click **OK** to continue.
   The edited groups or variables are updated in the Fieldbus Editor.

15.8.20.1.13 OPC UA - Additional Features

According to the OPC UA Standard, Part 7, the OPC UA Server implementation supports these features:

Service Set	Service	Supported
Attribute	• Read • History Read • Write • History Update	• YES • NO • YES • NO
Discovery	• Find Servers • Find Servers On Network • GetEndpoints • Register Server • Register Server 2	• YES • NO • YES • NO • NO
Method	Call	NO
Monitored Item	• Create • Modify • Set Monitoring Mode • Set Triggering • Delete Monitored Items	• YES • YES • YES • NO • YES
Node Management	• Add nodes • Add references • Delete nodes • Delete references	• NO • NO • NO • NO
Query	• Query First • Query Next	• NO • NO

Service Set	Service	Supported
Session	• Create Session • Activate Session • Close Session • Cancel	• YES • YES • YES • NO
Subscription	• Create • Modify • Set Publishing Mode • Publish • Republish • Transfer Subscriptions • Delete Subscriptions	• YES • YES • YES • YES • YES • NO • YES
View	• Browse • Browse next • TranslateBrowsePathsToNodeIds • Register Nodes • Unregister Nodes	• YES • YES • YES • YES • YES

15.8.20.1.14 OPC UA - Troubleshooting

If the connection to the server cannot be established, verify if the:

- Correct server IP and port are configured within the client configuration.
- Runtime Service can open the configured OPC UA listening port successfully.
- Machine running the Runtime Service can be reached within the network (e.g., ping command).
- Passwords for the server and client configuration match.
- Correct certificates are used.
- Correct security policy is configured for server and client.

15.8.20.1.15 OPC UA - Add TAG(s)* dialog box

Usage

The Add TAG(s)* dialog box is used to:

- Add and configure global variables and variables from all programs to specific variable groups of the driver.
- Create new groups for organizing variables.
- Remove selected variables.
- Shift groups or selected variables within the current group up or down.

The sections are:

- "Add TAG(s)* dialog box buttons" (➔ p. 950)
- "Variables Dictionary table" (➔ p. 952)
- "Group/Variables table" (➔ p. 953)

Access

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

1. In the **OPC UA Server** tree, right-click a Master node and select the **Add TAG(s)** option. (Figure 19-70)

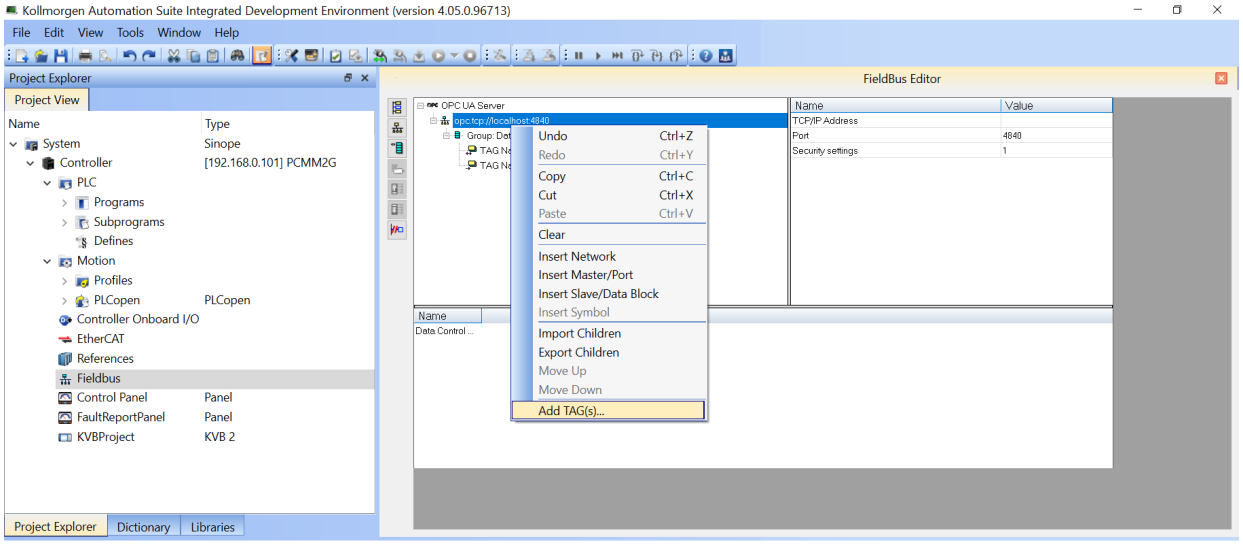


Figure 19-70: Add TAG(s) option

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The Add TAG(s)* dialog box opens. (Figure 19-71)

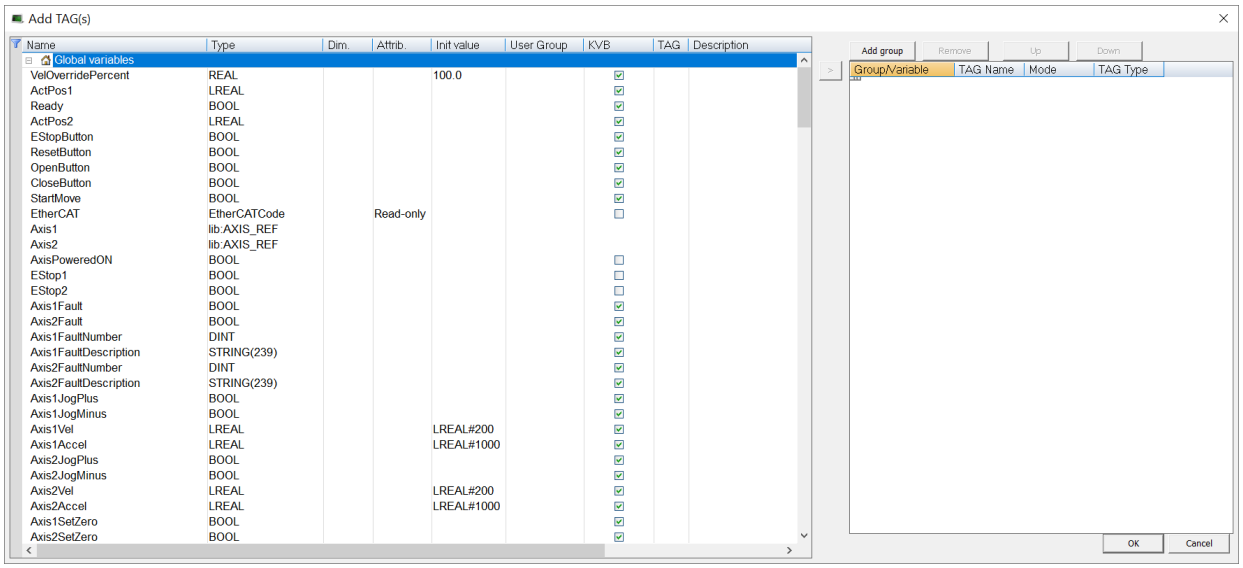


Figure 19-71: Example: Add TAG(s)* dialog box with content

15.8.20.1.16.1 Add TAG(s)* dialog box buttons

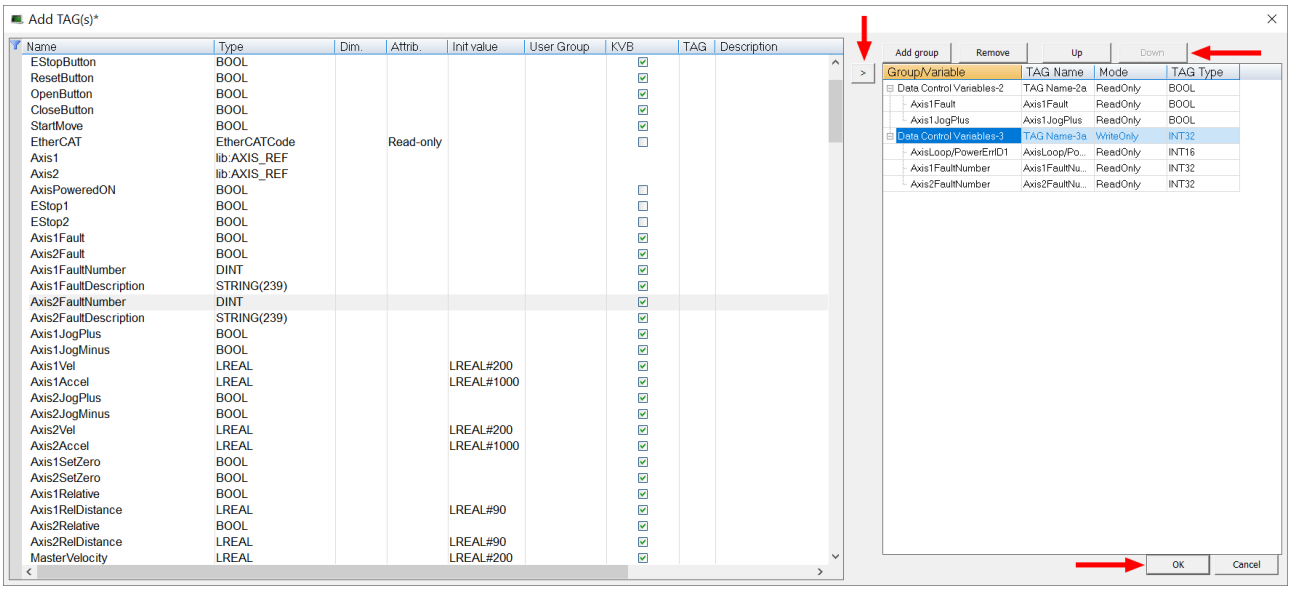


Figure 19-72: Add TAG(s)* dialog box buttons

Dialog box Description

Button	Description
> button	Click the > () button to move the selected variable to the **Group/Variable** table. **TIP** In the Variable Dictionary table, select a variable and drag it under the designated group.
Add group button	Click the **Add** group button to add a new group row in the **Group/Variable** table.
Remove button	Click the **Remove** button to immediately delete the selected variable from the **Group/Variable** table.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Button	Description
Up button	Click the **Up** button to move the selected item in the **Group/Variable** table up.
Down button	Click the **Down** button to move the selected item in the **Group/Variable** table down.
OK button	Click **OK** to save the changes or selections and close the dialog box.
Cancel button	Click **Cancel** to discard all changes or selections and close the dialog box.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Variables Dictionary table

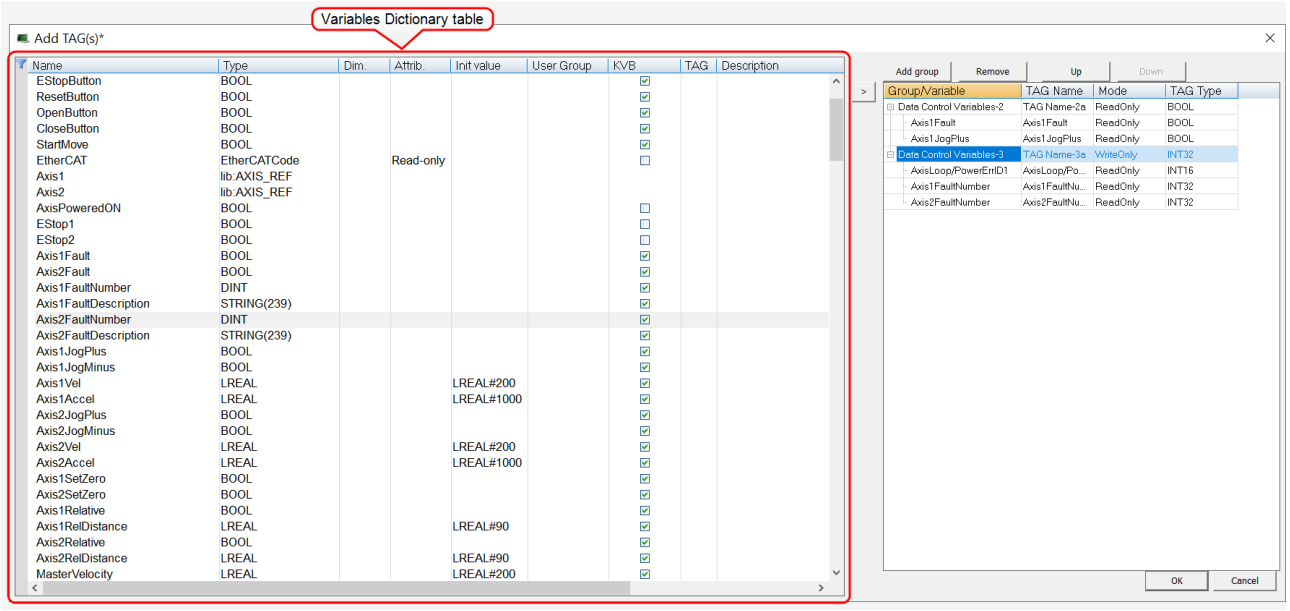


Figure 19-73: Add TAG(s)* dialog box - Available Dictionary table

Dialog box Description

Column	Description
Name column	This column is the name of the variable.
Type column	This column is the data type of the variable.
Dim. column	• This column shows the dimensions of the variable's array. • See Arrays.
Attrib. column	• This column is the attribute of the variable. • Example: Read-only
Init value column	This column is the initial value of the variable.
User Group column	• This column is the part of the description (comment). • Can be used as a TAG in the variable editor to filter or sort variables.
KVB column	N/A
TAG column	N/A
Description column	N/A

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Group/Variables table

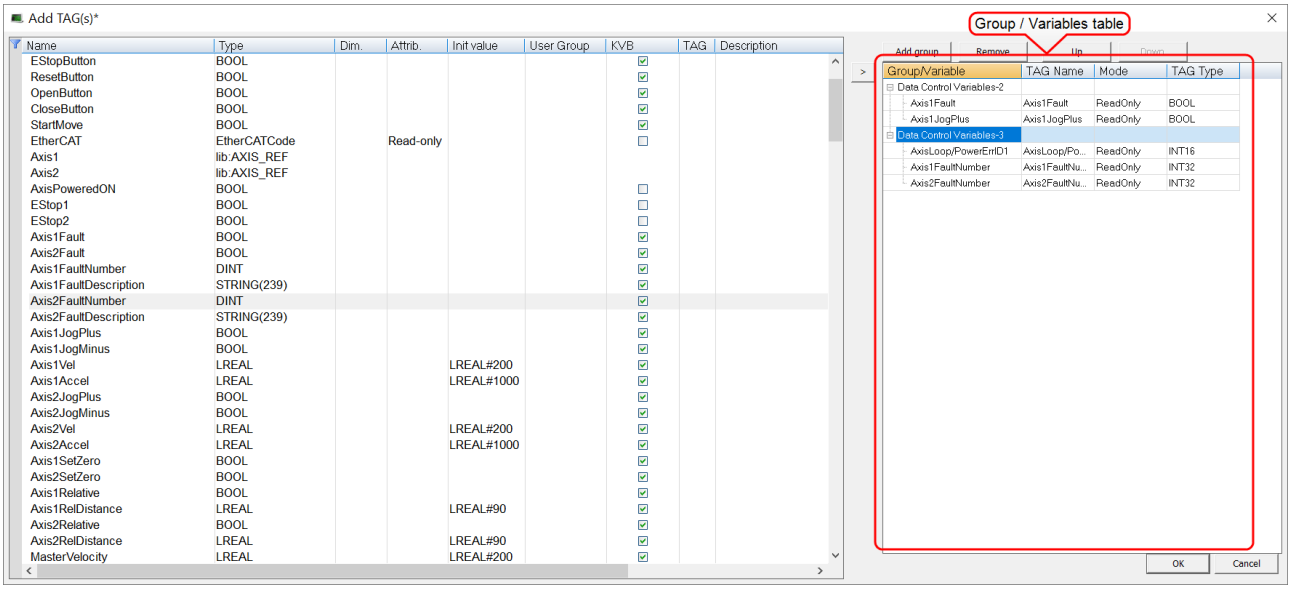


Figure 19-74: Add TAG(s)* dialog box - **Group/Variable** table

Dialog box Description

Column	Description
Group/Variable column	• This column is the user-defined name of the OPC UA group or variable. • In the **Group/Variable** column, double-click the column cell. The **Group/Variable** text box appears. • In the **Group/Variable** text box, enter a name for the new group.
TAG name column	• This column is the user-defined name of the TAG. • In the **TAG Name** column, double-click the column cell. The **TAG Name** text box appears. • In the **TAG Name** text box, enter the name used by OPC UA clients to identify this variable.
Mode column	• This column is the user-defined data access mode of the variable. • In the **Mode** column, double-click the column cell. The **Mode** list box appears. • In the **Mode** list box, double-click a selected option.
TAG Type column	• This column is the user-defined data type of the TAG. • In the **TAG Type** column, double-click the column cell. The **TAG Type** list box appears. • In the **TAG Type** list box, double-click a selected option.

15.8.20.1.17 OPC UA - Endpoint dialog box

Usage

The **Endpoint** dialog box is used to add an individual endpoint to an OPC UA Server.

See "OPC UA - Insert an Endpoint" (➔ p. 934).

Access

1. In the Fieldbus Editor tree, select **OPC UA Server**.
   The **Insert Master/Port** button in the vertical toolbar is activated. (Figure 19-75)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

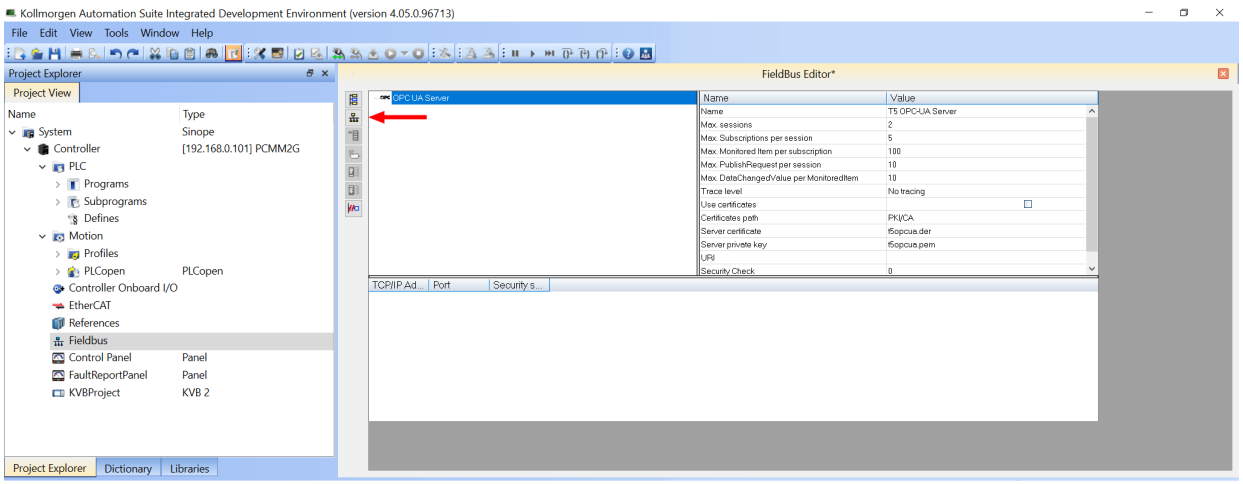


Figure 19-75: Insert Master/Port button

2. On the vertical toolbar, click the **Insert Master/Port** button ( ). The Endpoint dialog box opens. (Figure 19-76)

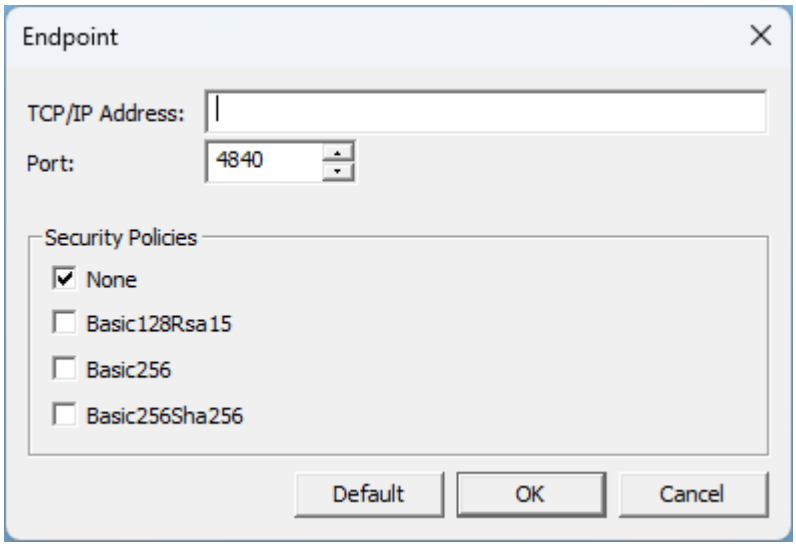


Figure 19-76: Endpoint dialog box

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Dialog box Description

Section	Element	Description
	TCP/IP Address text box	• In the **TCP/IP Address** text box, enter the TCP/IP address of the KAS controller (e.g., PCMM2G). • The default is a blank text box. **TIP** • Leave this text box blank to use the KAS controller assigned IP address.
	Port spin box	• In the **Port** spin box, enter the Listening Port opened by the OPC UA Server. • The range is 1025 to 65535. • The default is 4840.
Security Policies section	**None** check box	**IMPORTANT** • If using certificates in the "OPC UA - Server dialog box" (→ p. 960), **Security settings** section, at least one of the Basic128Rsa15, Basic256, or Basic256Sha256 check boxes must be selected. • If the **Anonymous** check box is selected in the "OPC UA - Server dialog box" (→ p. 960), **Authentication settings** section, the **None** check box must be selected. • Click the **None** check box so that no security settings are used. • This check box is the default selection.
	Basic128Rsa15 check box	• Click the **Basic128Rsa15** check box to use both the Basic128 and RSA15 security policies. • RSA15 is the key wrap algorithm. • 128-bit Basic is the message encryption algorithm.
	Basic256 check box	Click the **Basic256** or **Basic256Sha256** check box to use the 256-bit Basic as the message encryption algorithm.
	Basic256Sha256 check box	Click the **Basic256Sha256** check box to use both the Basic256 and SHA256 security policies. • SHA256 is the signature digest. • 256-bit Basic is the message encryption algorithm.
	Default button	Click **Default** to reset all changes or selections back to the default settings.
	OK button	Click **OK** to save the changes or selections and close the dialog box.
	Cancel button	Click **Cancel** to discard all changes or selections and close the dialog box.

15.8.20.1.18 OPC UA - Group: dialog box

Usage

The **Group:** dialog box is used to create an individual group.

NOTE

- The Group: dialog box can only create a single group.
- See "OPC UA - Add Multiple Groups and Variables to the Driver Configuration" (→ p. 941).

Access

1. In the Fieldbus Editor tree, select an **OPC UA Server** endpoint.
   The **Insert Slave/Data Block** button in the vertical toolbar is activated. (Figure 19-77)

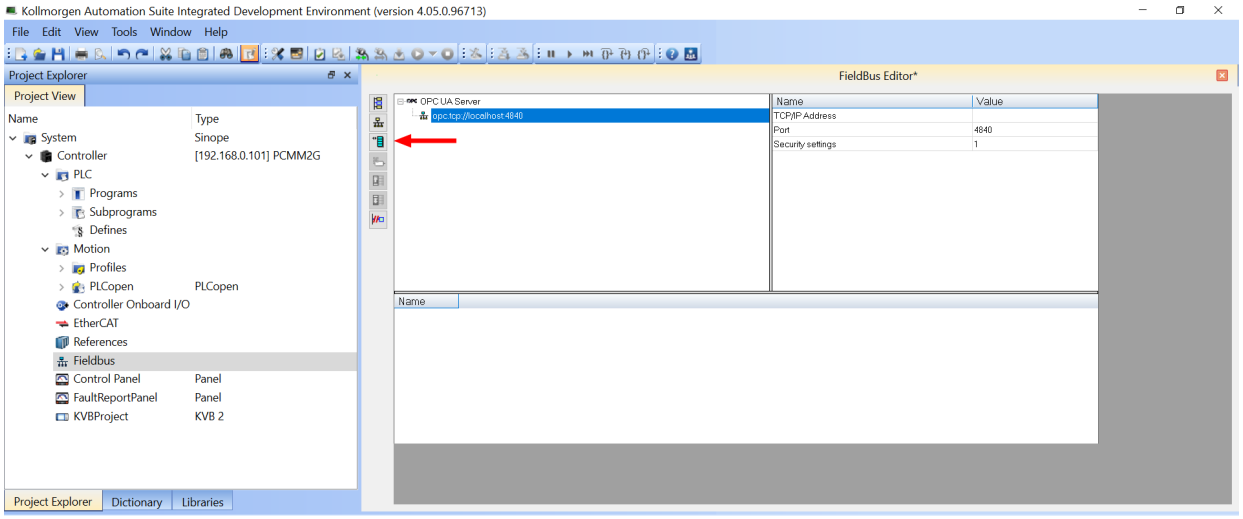


Figure 19-77: Insert Slave/Data Block button

2. On the vertical toolbar, click the **Insert Slave/Data Block** button ( ).
   The Group: dialog box opens. (Figure 19-78)

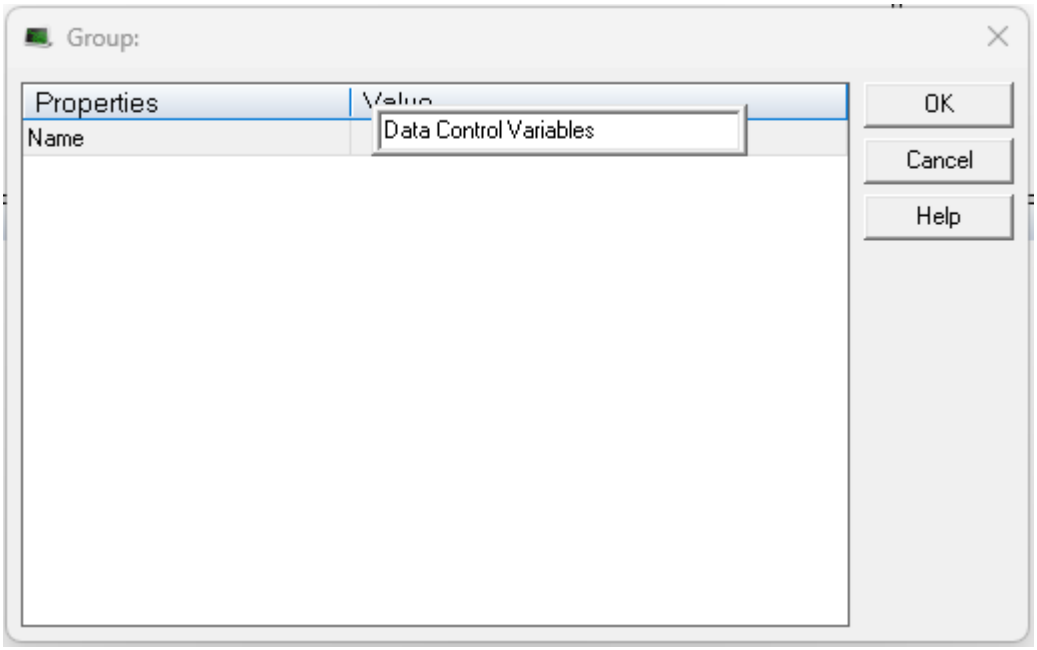


Figure 19-78: Group: dialog box with Name - Value text box

Dialog box Description

Section	Element	Description
Properties column		The **Properties** column lists the properties to define for this variable.
Value column		The **Value** column shows the entered **Name** of the data group.
	Name - Value text box	In the **Name - Value** text box, enter the name of the new data group.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Section	Element	Description
	Default button	Click **Default** to reset all changes or selections back to the default settings.
	OK button	Click **OK** to save the changes or selections and close the dialog box.
	Cancel button	Click **Cancel** to discard all changes or selections and close the dialog box.

15.8.20.1.19 OPC UA - Insert Variable dialog box

Usage

The **Insert Variable (???)** dialog box is used to define a variable for a data group.

Access

1. In the Fieldbus Editor tree, select the **OPC UA Server** data group to assign a new variable to.
   The **Insert Variable** button in the vertical toolbar is activated. (Figure 19-79)

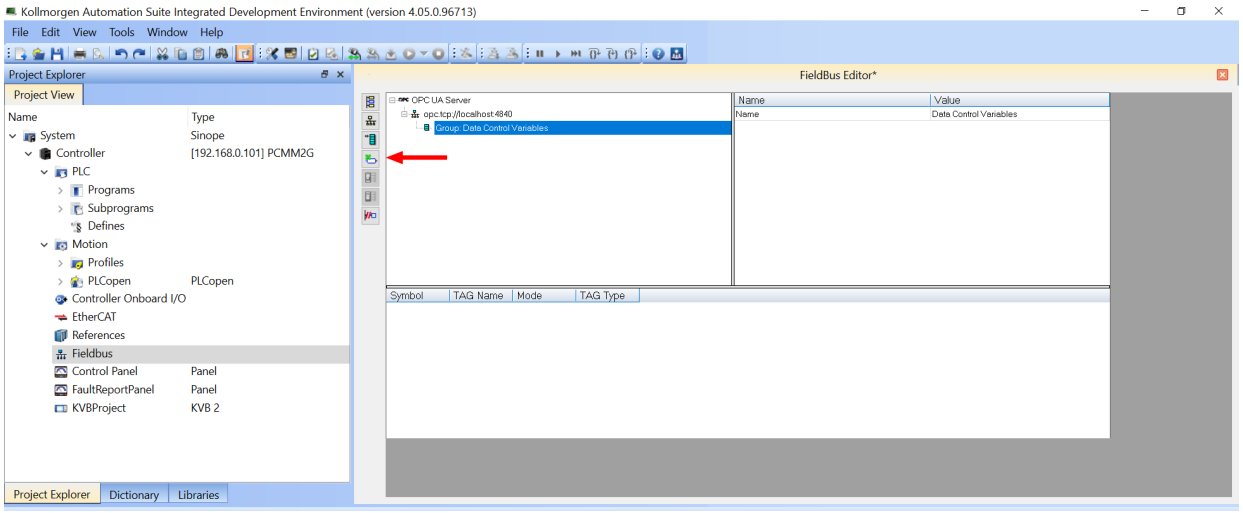


Figure 19-79: Insert Variable button

2. On the vertical toolbar, click the **Insert Variable** button ( ).
   The Insert Variable (???) dialog box opens. (Figure 19-80)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

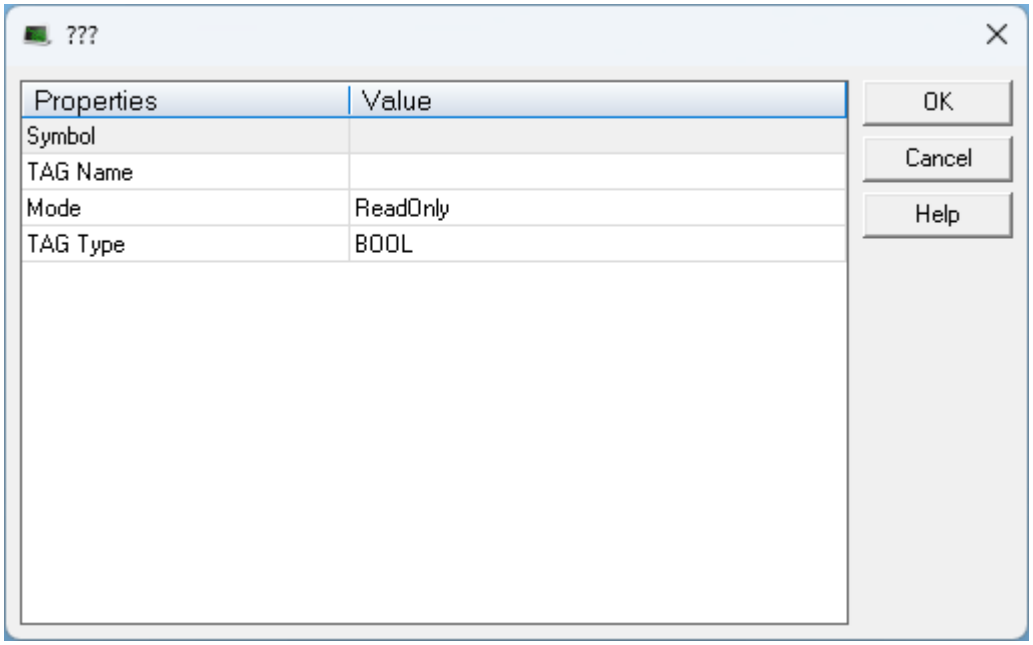


Figure 19-80: Insert Variable (???) dialog box

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Dialog box Description

Section	Element	Description
Properties column		The **Properties** column lists the properties to define for this variable.
Value column		The **Value** column shows the entered **Name** of the variable.
Symbol row	**Variables** combo-box	• In the Symbol row, double-click the **Value** column cell to open the **Variables** combo-box. • In the **Variables** combo-box, select the applicable variable. **NOTE** Only a single variable can be selected in the combo-box.
TAG Name row	**TAG Name - Value** text box	In the **TAG Name - Value** text box, enter the name used by the OPC UA client to address the variable.
Mode row	**Mode - Value** list box	• In the **Mode - Value** column cell, double-click the cell to open the **Mode - Value** list box. • In the **Mode - Value** list box, select the Read/Write permissions for this variable. The options are: • Read-only • Write-only • Read / Write (default) • Nb. Use Sessions • Displays the number of clients connected (sessions) to the server. • Nb. Used Subscriptions • Displays the number of subscriptions to the server. • Nb. Used Monitored Items • Displays the number of monitored items to the server. • Nb. Used Publish Request • Displays the current managed requests in the server. • Nb. Used Notification Msg • Displays the current number of notification messages send by the server.
TAG Type row	**TAG Type - Value** list box	• In the **TAG Type Value** column cell, double-click the cell to open the **Mode - Value** list box. • In the **TAG Type - Value** list box, select the OPC UA data type for this variable. The options are: • BOOL • FLOAT32 • FLOAT64 • INT8 • INT16 • INT32 • INT64 • STRING256 • UINT8 • UINT16 • UINT32 • UINT64
	Default button	Click **Default** to reset all changes or selections back to the default settings.
	OK button	Click **OK** to save the changes or selections and close the dialog box.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Section	Element	Description
	Cancel button	Click **Cancel** to discard all changes or selections and close the dialog box.

15.8.20.1.20 OPC UA - Server dialog box

Usage

The Server dialog box is used to define the application settings for the OPC UA Server.

Access

1. In the Project Explorer, double-click the Fieldbus node to open the Fieldbus Editor.
2. On the vertical toolbar, click the **Insert Network** (Configuration) button ( ).
   The **Add Configuration** dialog box opens.
3. In the **Select a configuration** tree, select **OPC UA Server**. (Figure 19-81)

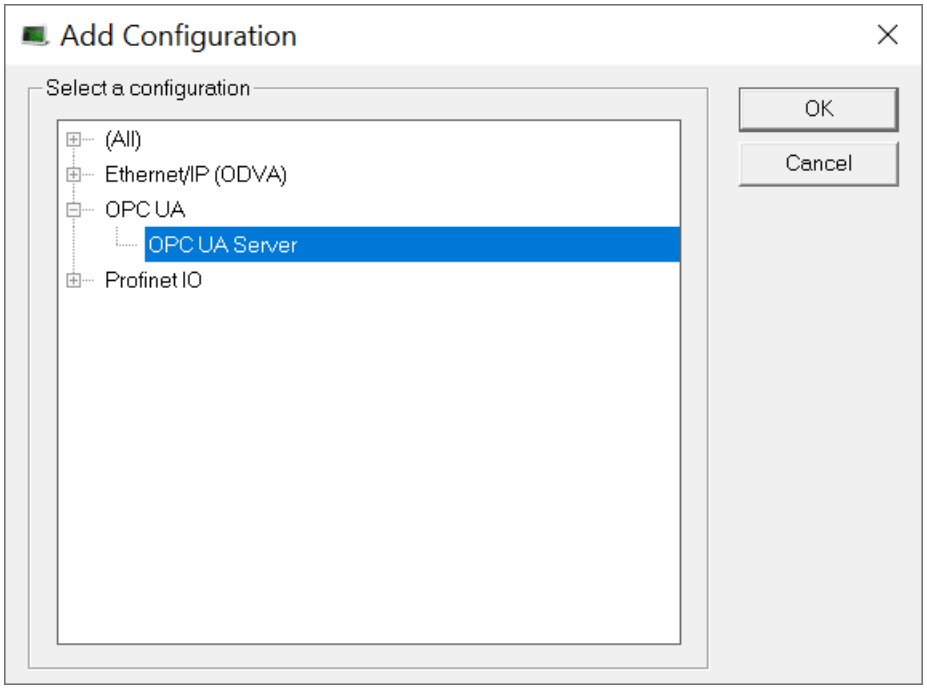


Figure 19-81: Add Configuration dialog box - OPC UA Server selected

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4. Click **OK** to save the changes or selections and close the dialog box.
   The Server dialog box opens. (Figure 19-82)

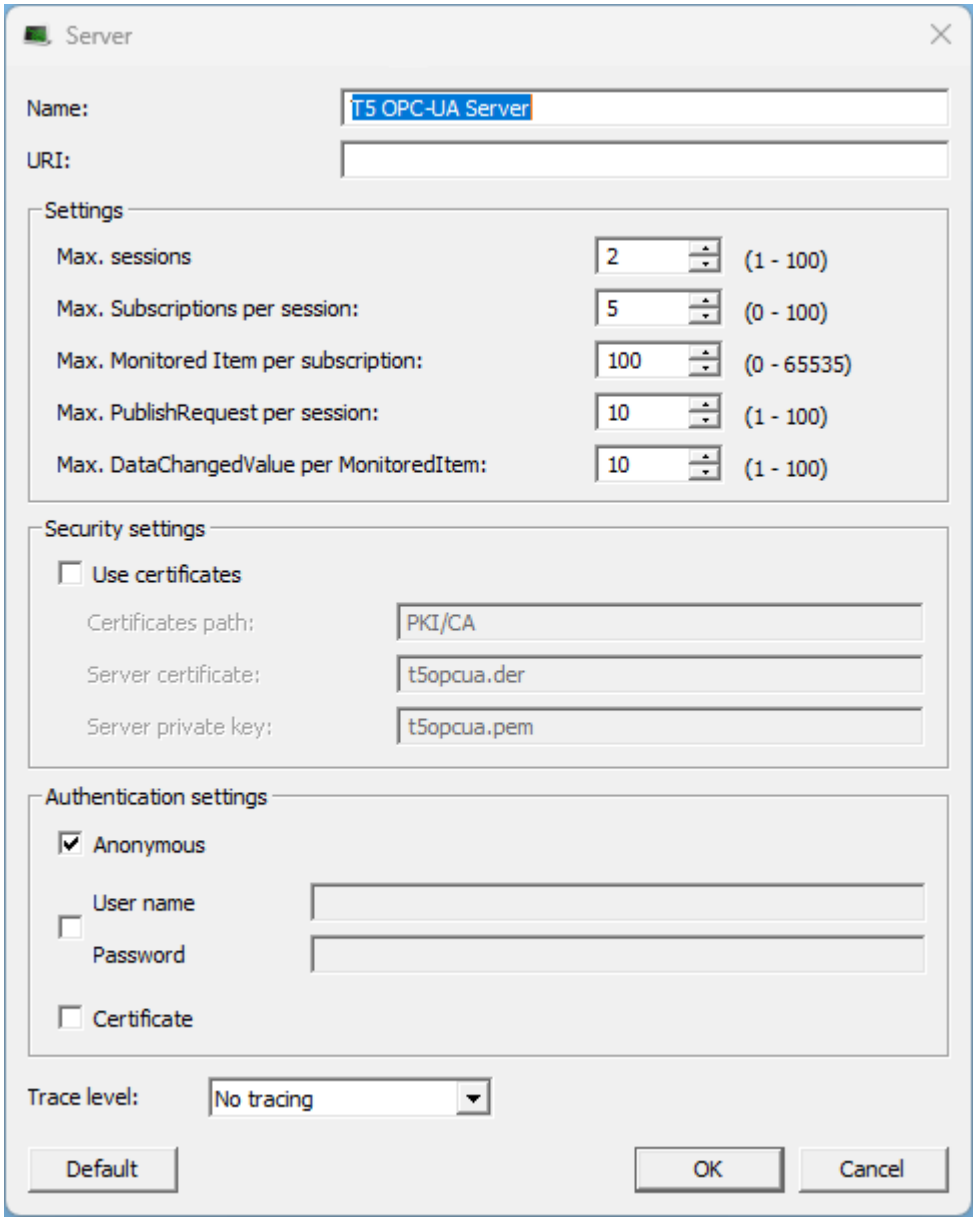


Figure 19-82: Server dialog box

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Dialog box Description

Section	Element	Description
	Name text box	• In the **Name** text box, enter a name for this specific server configuration. • This Name is used to distinguish the server instance. • The server is identified via this name. • The default is T5 OPC UA Server.
	URI text box	• In the **URI** text box, enter the URL used for the certificate authentication. • It must be the same URL as the URL used in the Subject Alternative Name of the server certificate.
Settings section	**Max. Sessions** spin box	• In the **Max. Sessions** spin box, enter the maximum number of simultaneously connected OPC UA client sessions. • The range is 1 to 100. • The default is 2.
	Max. Subscriptions per session spin box	• In the **Max. Subscriptions per session** spin box, enter the maximum number of subscription groups a client can create per session. • The range is 1 to 100. • The default is 5.
	Max. Monitored Item per subscription spin box	• In the **Max. Monitored Item per subscription** spin box, enter the maximum number of items that can be monitored within a single subscription. • The range is 0 to 65535. • The default is 100.
	Max. Publish Request per session spin box	• In the **Max. Publish Request per session** spin box, enter the maximum number of active publish requests per session. • Requests are initiated immediately after a subscription is created. • The range is 1 to 100. • The default is 10.
	Max. Data Changed Value per Monitored Item spin box	• In the **Max. Data Changed Value per Monitored Item** spin box, enter the maximum number of value changes stored in memory for each Monitored Item. • The range is 1 to 100. • The default is 10.
Security settings section	**Use certificates** check box	***TIP*** Kollmorgen recommends accepting the defaults in the **Security settings** section to use self-signed certificates. • The settings selected in this section enables secure communication using digital certificates. • In the **Security settings** section, click the **Use certificates** check box to use digital certificates for data encryption between the server and client. When selected, the text boxes of this section are activated. • When selected, the text boxes of this section are activated. • By default, this check box is not selected.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Section	Element	Description
	Certificates path text box	• In the **Certificates path** text box, enter the relative directory path where the certificates are stored. • The default is PKI/CA.
	Server certificate text box	• In the **Server certificate** text box, enter the filename of the server certificate used for encryption. • The default is t5opcua.der. **IMPORTANT** The file **must be** stored in a subfolder called **certs** in the designated certificates folder.
	Server private key text box	• In the **Server private key** text box, enter the filename of the server private key used for encryption. • The default is t5opcua.pem. **IMPORTANT** The file **must be** stored in a subfolder called **private** in the designated certificates folder.
Authentication settings section	**Anonymous** check box	• The check boxes in the **Authentication settings** section can be used singly or in any combination. • Click the **Anonymous** check box to use non-encrypted communications.
	User name/Password check box	Click the **User name/Password** check box to activate the **User name** and **Password** text boxes.
	User name text box	In the **User name** text box, enter a user name restricting the access to the server.
	Password text box	In the **Password** text box, enter a password restricting the access to the server.
	Certificate check box	Click the **Certificate** check box so the server allows the client to authenticate with a certificate.
	Trace Level list box	• Click the **Trace Level** list box arrow and select the log level for OPC UA Server Runtime Service messages written to the controller log output. See Log Messages tabs. The options are: • **No tracing** - Select the **No tracing** option to log only critical and warning messages. • This is the default. • **Information** - Select the **Information** option to log information messages. • This selection includes critical and warning messages. • **Debug** - Select the **Debug** option to log debug messages. • This selection includes critical, warning, and information messages.
	Default button	Click **Default** to reset all changes or selections back to the default settings.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Section	Element	Description
	OK button	Click **OK** to save the changes or selections and close the dialog box.
	Cancel button	Click **Cancel** to discard all changes or selections and close the dialog box.

15.8.20.2 OPC UA - External Software Communications

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.8.20.2.1 OPC UA - Establish a Connection using UaExpert

These procedures are used to communicate with the PCMM2G OPC UA Server using the UaExpert tool.

- "OPC-UA - Establish a Server Connection with Certificates - UaExpert" (→ p. 965)
- "OPC-UA - Establish a Server Connection without Certificates - UaExpert" (→ p. 980)

See Also

- "OPC-UA - UaExpert Installation" (→ p. 965)
- "OPC UA - Troubleshooting" (→ p. 948)

15.8.20.2.2.1 OPC-UA - UaExpert Installation

IMPORTANT

Registration is required to download UaExpert.

- UaExpert is a free, platform-independent OPC UA client used for testing, diagnostics, and development.
  - It provides a graphical interface to connect to OPC UA servers, browse address spaces, monitor variables, and test communication.
- This tool is provided by the Unified Automation GmbH and can be downloaded from here: UaExpert.
  - Registration is required to download UaExpert.
  - See the UaExpert documentation for install requirements and initial start-up instructions.

15.8.20.2.3.2 OPC-UA - Establish a Server Connection with Certificates - UaExpert

These are the overall steps when using certificates for an OPC UA Server connection:

"1. PCMM2G Configuration (with Certificates)" (→ p. 965)

"2. UaExpert Configuration (with Certificates)" (→ p. 968)

"3. UaExpert Connection to the PCMM2G (with Certificates)" (→ p. 977)

NOTE

Videos of these features are here: PCMM2G OPC UA Server Setup in KAS-IDE.

15.9 1. PCMM2G Configuration (with Certificates)

The procedure in this section configures the PCMM2G for connection to UaExpert.

1. Verify the "OPC-UA - UaExpert Installation" (→ p. 965) procedure is completed.
2. Open a web browser and enter the PCMM2G IP address.
3. Login as administrator.
   See User Authentication.
4. Click the KAS Application tab.
5. Click the **Stop** button if the Status of KAS app is started.
6. Click the **Settings** tab.
7. Click the **Certificates** tab. (Figure 19-83)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

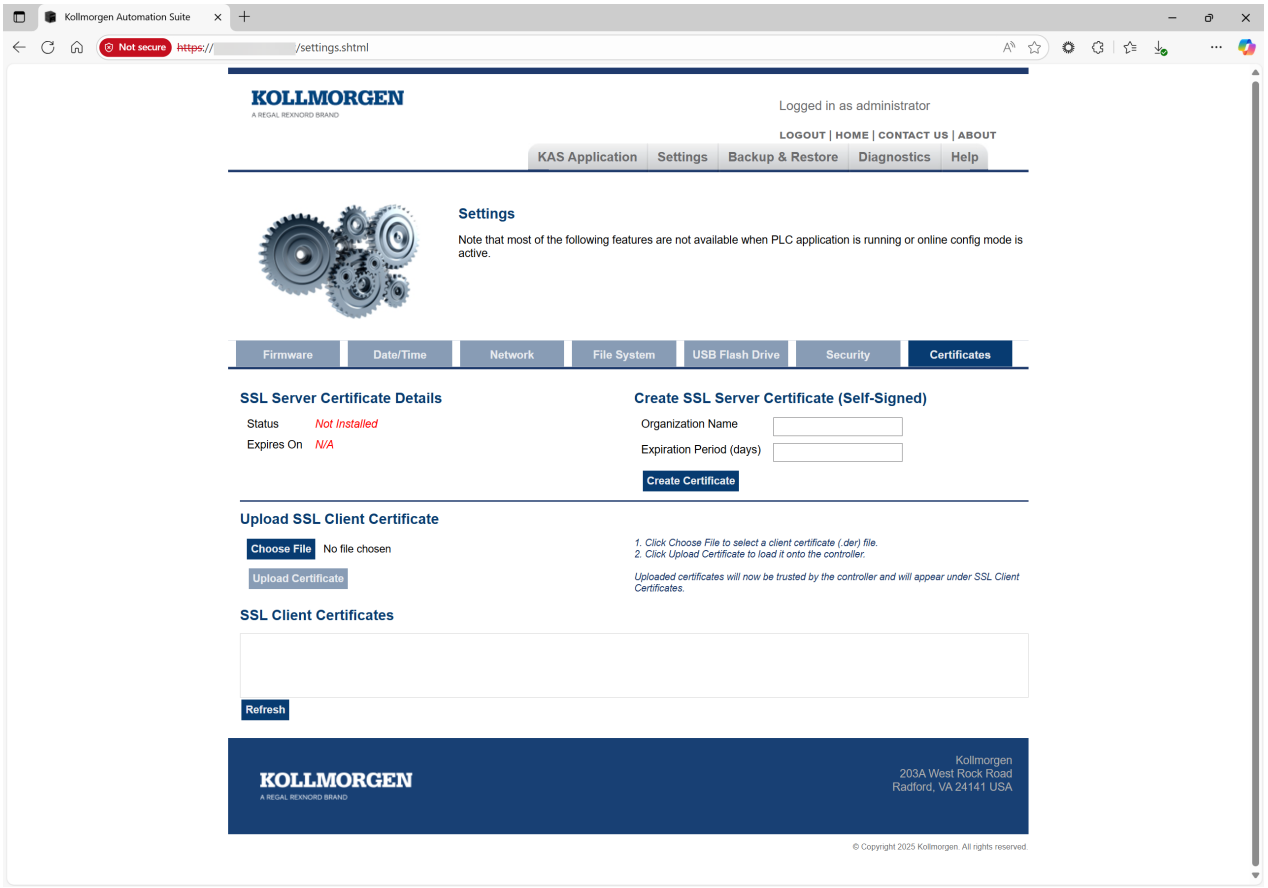


Figure 19-83: Certificates tab

8. If the **Status** in the SSL Server Certificate Details section is shown as **Not Installed**, create a new SSL Server Certificate.
   See Install the SSL Certificate.
9. Start the KAS-IDE.
10. Create a new project or open an existing KAS project.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

11. Double-click an OPC UA Server node in the Fieldbus Editor project tree.
The "OPC UA - Server dialog box" (→ p. 960) opens. (Figure 19-84)

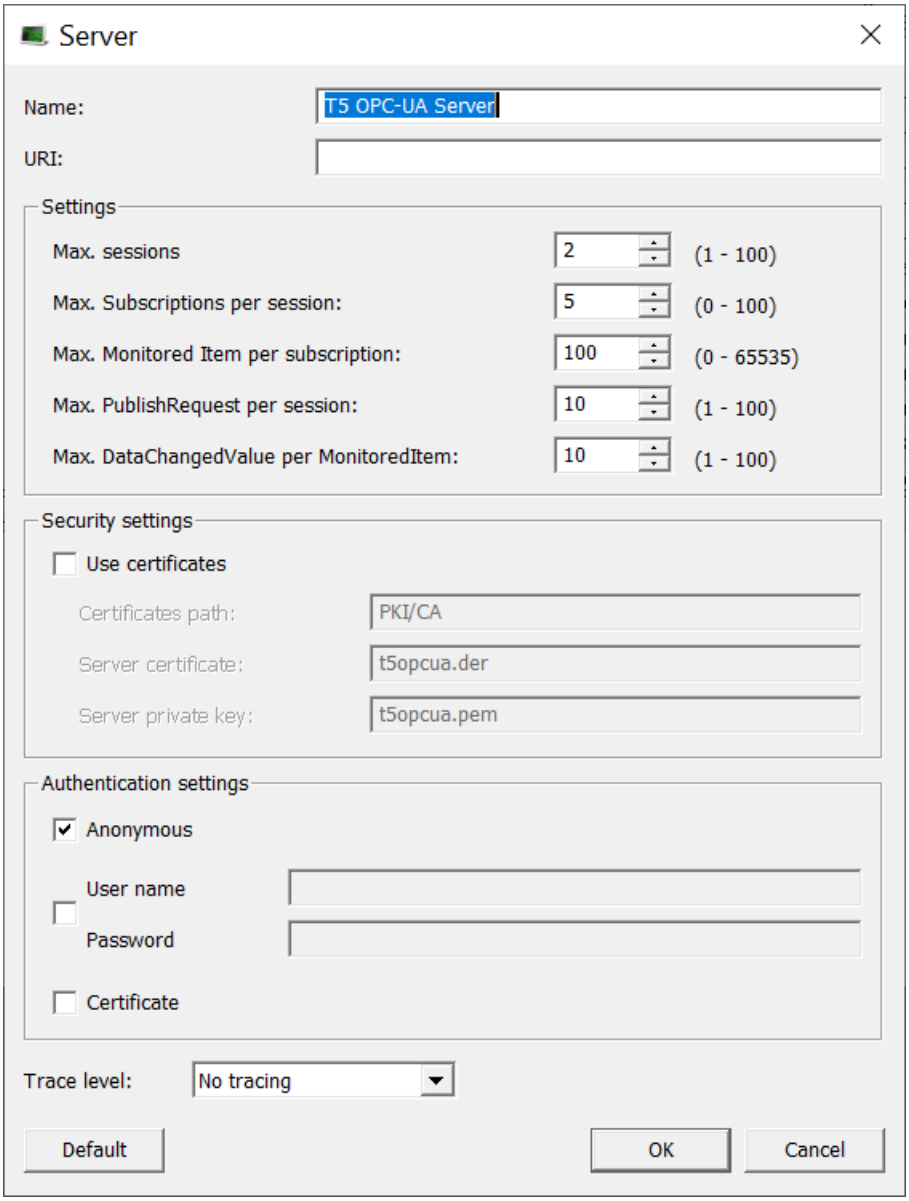


Figure 19-84: Server dialog box

12. In the **Security settings** section, click the **Use certificates** check box to use digital certificates for data encryption between the server and client.
When selected, the text boxes of this section are activated.

TIP

Kollmorgen recommends accepting the defaults in the **Security settings** section to use self-signed certificates.

13. In the **Authentication settings** section:
    a. Clear the **Anonymous** check box.
    b. Optional: Click the **User name/Password** check box to activate the **User name** and **Password** text boxes.
    c. If applicable, enter a **User name** and **Password** for the configuration.
    d. Click the **Certificate** check box so the server allows the client to authenticate with a certificate.
    e. Click **OK** to save the changes or selections and close the dialog box.
    The Fieldbus Editor returns showing the added OPC UA Server node.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

14. On the vertical toolbar, click the **Insert Master/Port** button ( ).
    The "OPC UA - Endpoint dialog box" (➔ p. 953) opens. (Figure 19-85)

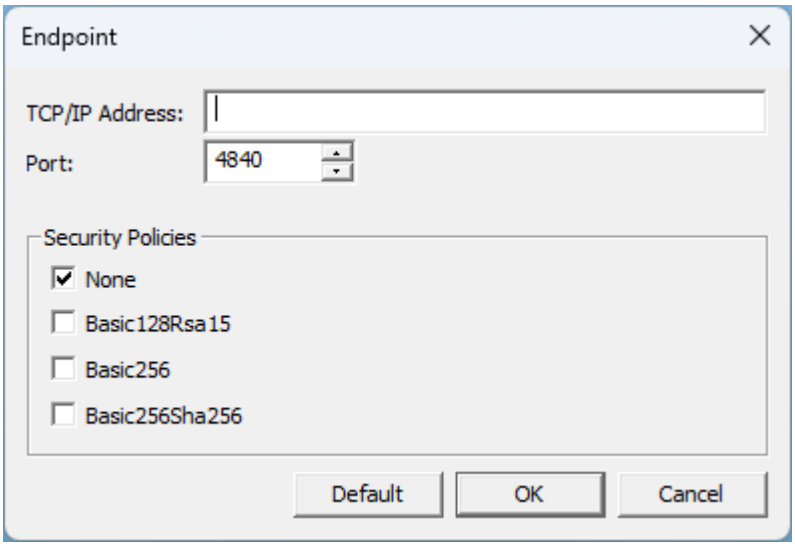


Figure 19-85: Endpoint dialog box

15. In the **Security Policies** section:
    a. Clear the **None** check box.
    b. Click the **Basic256** or **Basic256Sha256** check box to use the 256-bit Basic as the message encryption algorithm.
    c. Click **OK** to save the changes or selections and close the dialog box.
    The Fieldbus Editor returns.
16. Add applicable groups and variables using the "OPC UA - Add Multiple Groups and Variables to the Driver Configuration" (➔ p. 941) procedure.
17. Compile and download the project to the KAS controller (e.g., PCMM2G).
18. Continue with "2. UaExpert Configuration (with Certificates)" (➔ p. 968).

15.10 2. UaExpert Configuration (with Certificates)

The procedure in this section configures UaExpert for connection to PCMM2G.

1. Verify the procedure in the "1. PCMM2G Configuration (with Certificates)" (➔ p. 965) section is completed.
2. Start UaExpert.
   The UaExpert main window opens.
3. In the Project tree, right-click **Servers** and click **Add**. (Figure 19-86)

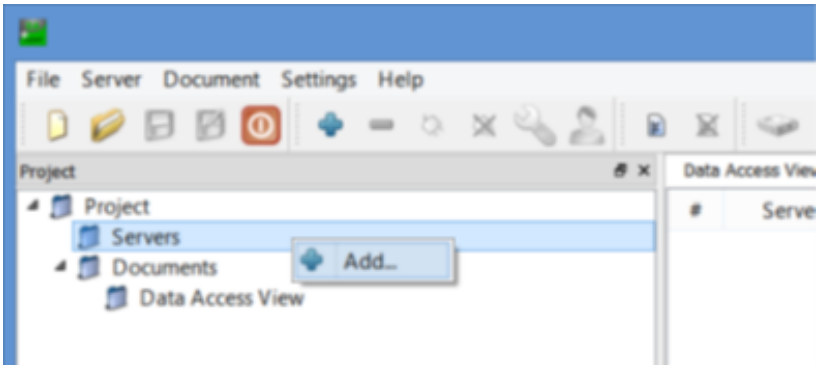


Figure 19-86: Add option

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The **Add Server** dialog box opens. (Figure 19-87)

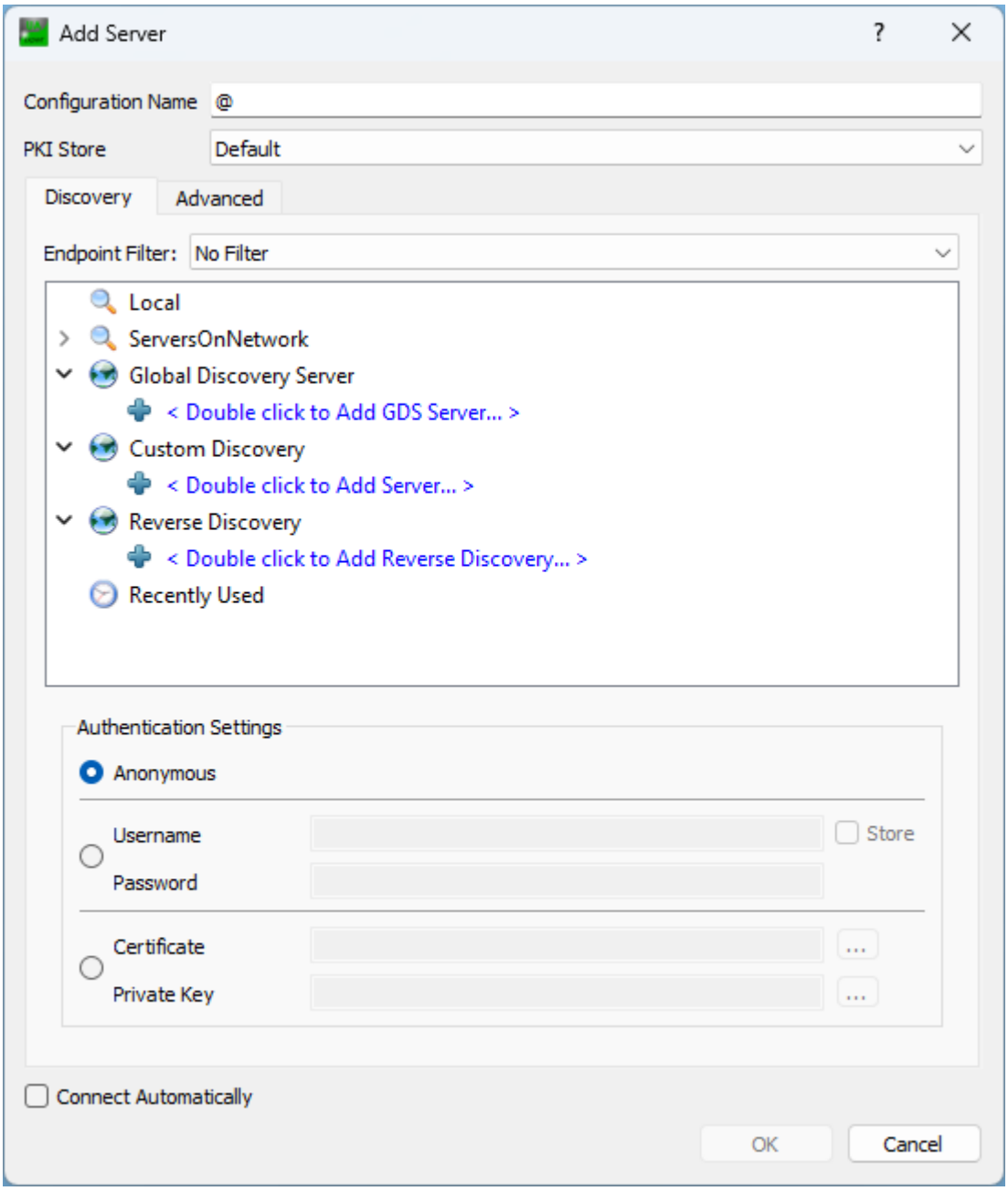


Figure 19-87: UaExpert Add Server dialog box

4. Under **Custom Discovery**, double-click the text.
The **Enter URL** dialog box opens.

5. Enter the URL as `opc.tcp://(PCMM2G_IP_Address)` to add the OPC UA Server. (Figure 19-88)

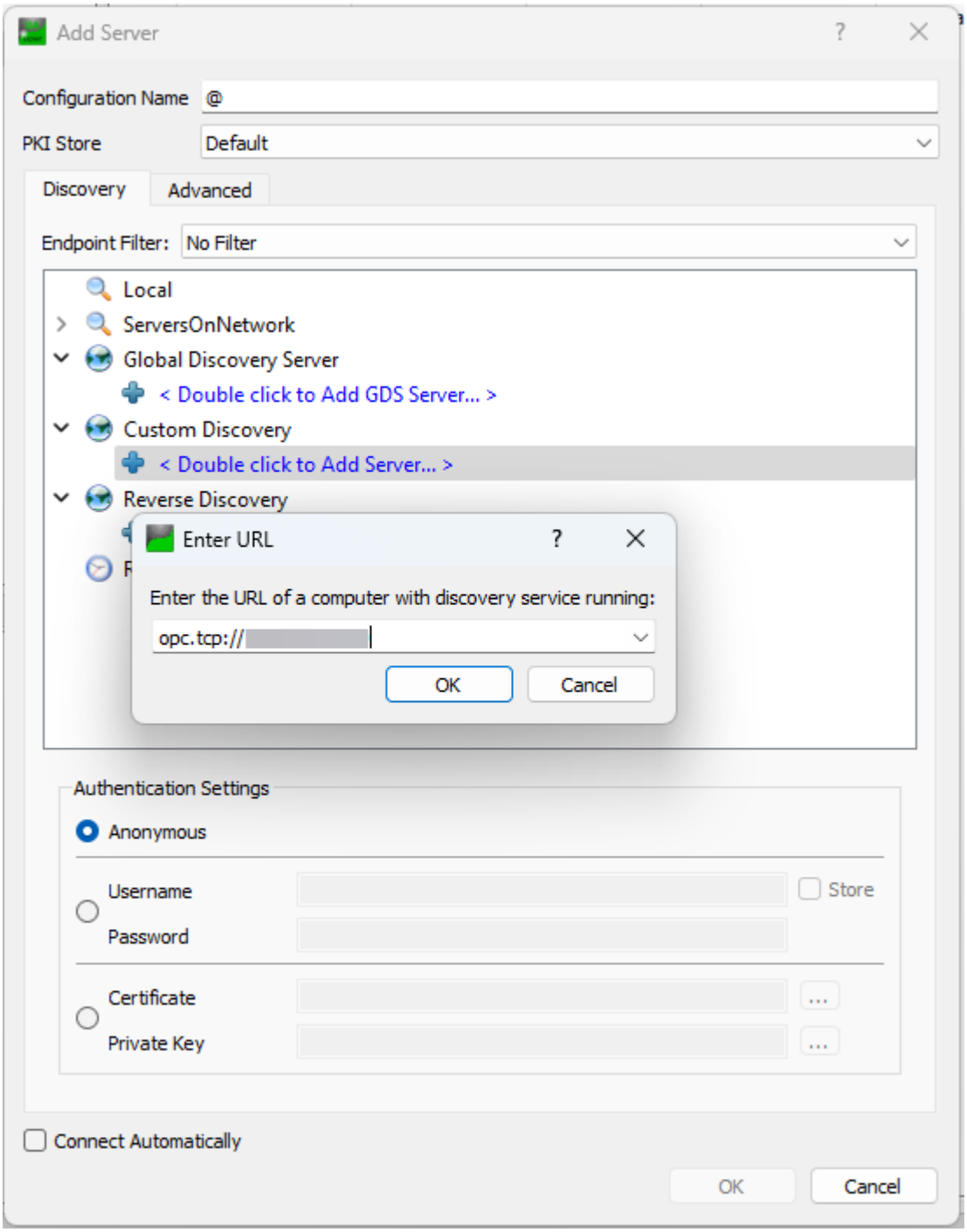


Figure 19-88: Enter URL dialog box

6. Click **OK** to save the changes or selections and close the dialog box.
The **Add Server** dialog box returns.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

A Replace Hostname message may appear. (Figure 19-89)

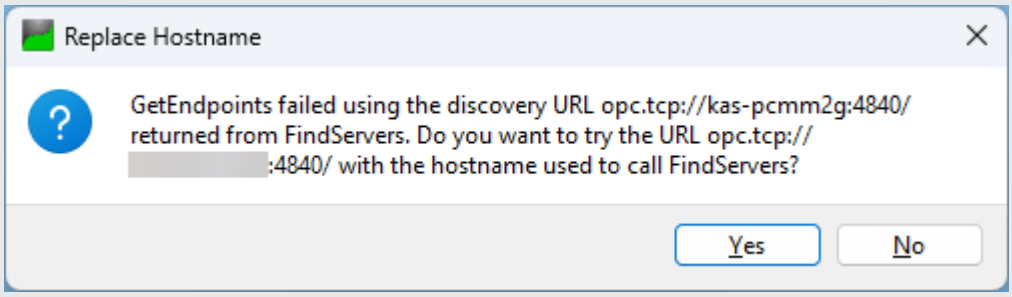


Figure 19-89: Replace Hostname message

This prompt appears because the PCMM2G supports OPC UA device discovery using its IP address, not by its hostname.

Click **Yes** to continue.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

7. Under **Custom Discovery**, expand the newly added node.
8. Under the opc.tcp://(PCMM2G_IP_Address), select **Basic256Sha256 - Sign & Encrypt**. (Figure 19-90)

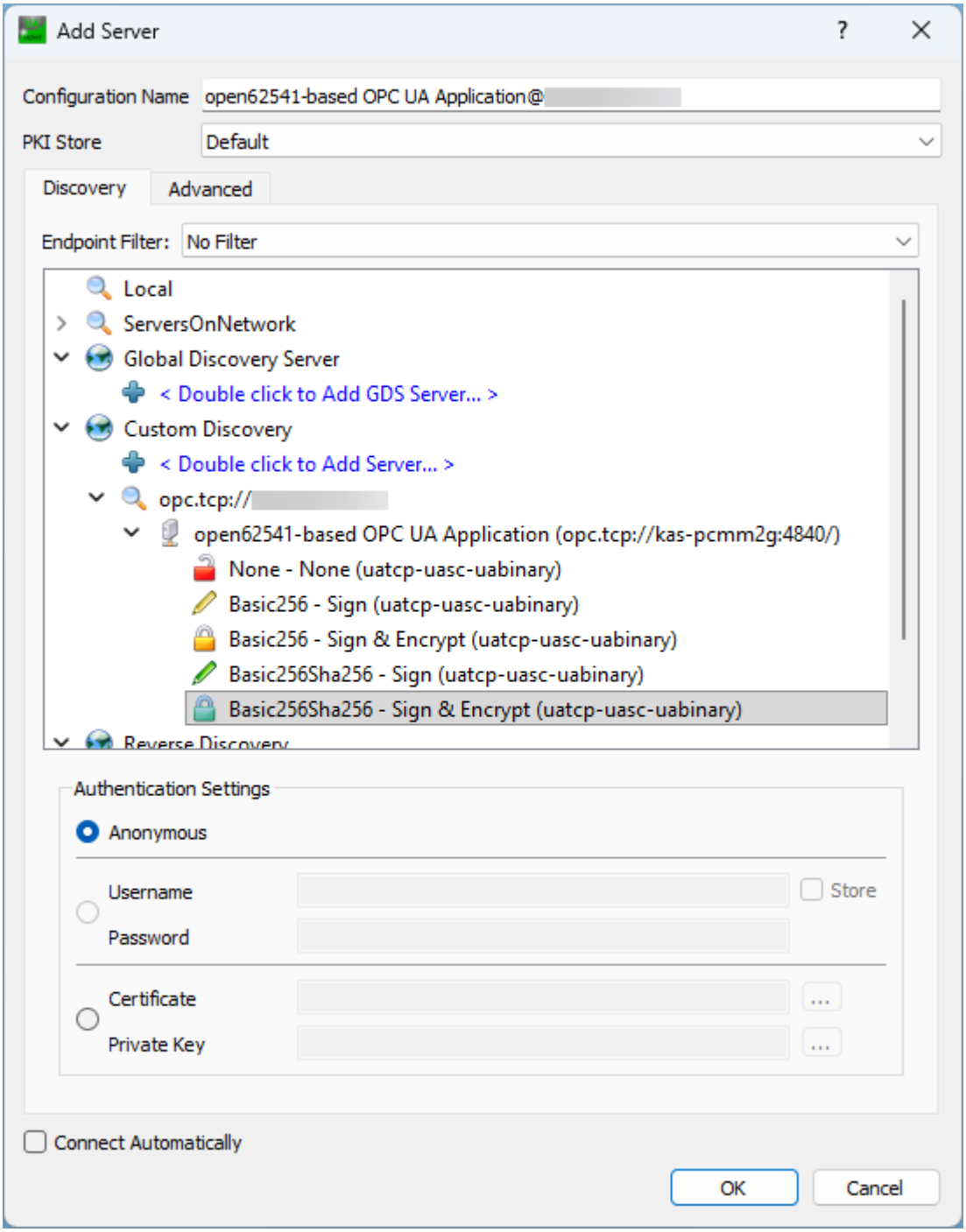


Figure 19-90: Add Server dialog box - Basic256Sha256 - Sign & Encrypt

9. In the **Authentication settings** section:
   a. Click the **Certificate / Private Key** option button.
   b. In the **Certificate** check box, enter the path to the UaExpert certificate.
   Example: C:/Users/<<user_
   name>>/AppData/Roaming/unifiedautomation/uaexpert/PKI/own/certs/uaexpert.der

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

c. In the **Private Key** text box, enter the path to the UaExpert private key. (Figure 19-91)
Example: C:\Users\<<user_
name>>\AppData\Roaming\unifiedautomation\uaexpert\PKI\own\private\uaexpert_key.pem

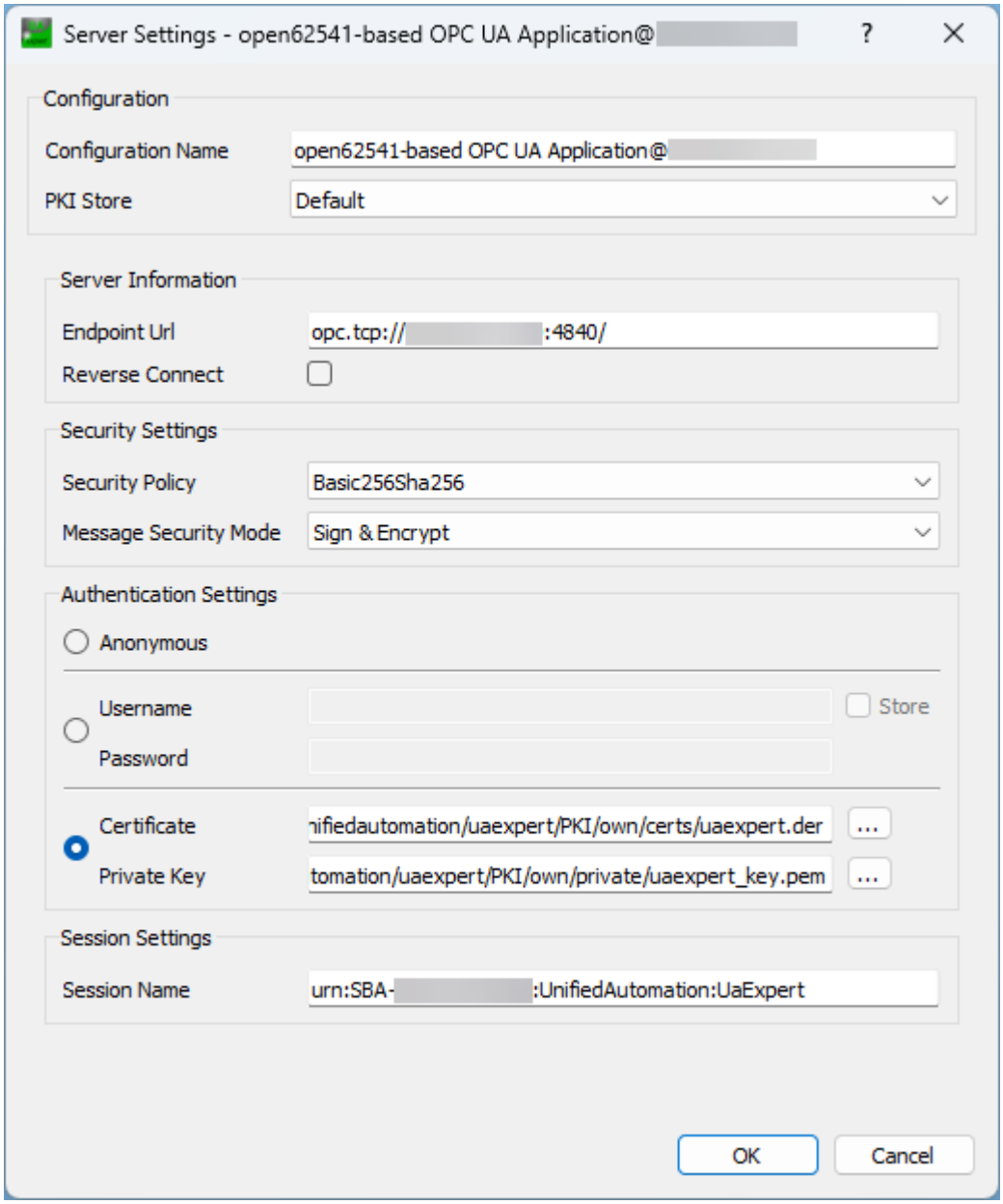


Figure 19-91: Server Settings dialog box

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- A Username and Password can be used if this Authentication Setting option is enabled in the KAS project. (Figure 19-92)
- The password is always encrypted if **Use Certificates** is enabled in the KAS project.

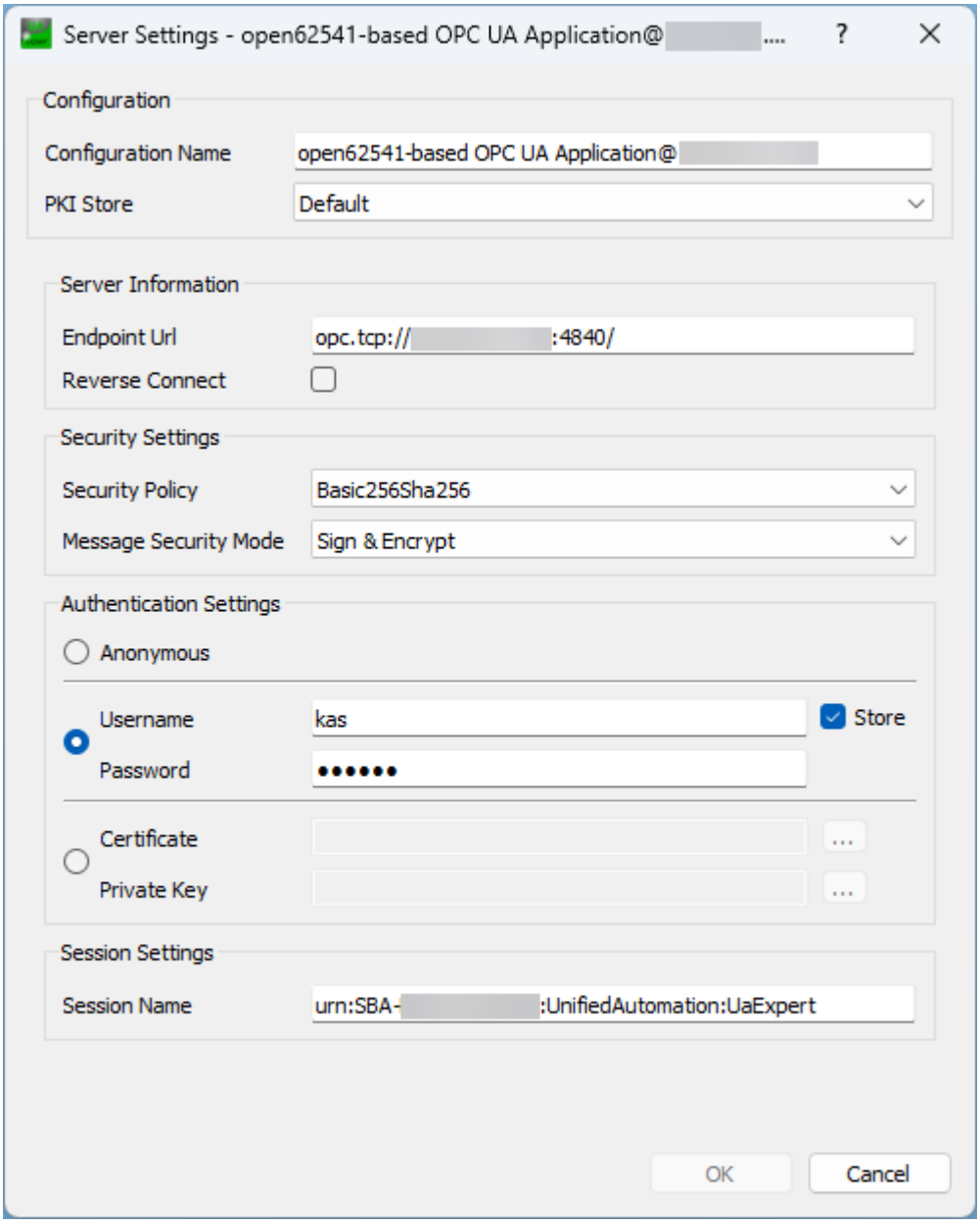


Figure 19-92: Server Settings dialog box - Username and Password

d. Click **OK** to save the changes or selections and close the dialog box.
The server is now available in the UaExpert project tree.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

10. On the UaExpert toolbar, click the **Settings** button and select **Manage Certificates > Properties**. The UaExpert **Manage Certificates** dialog box opens. (Figure 19-93)

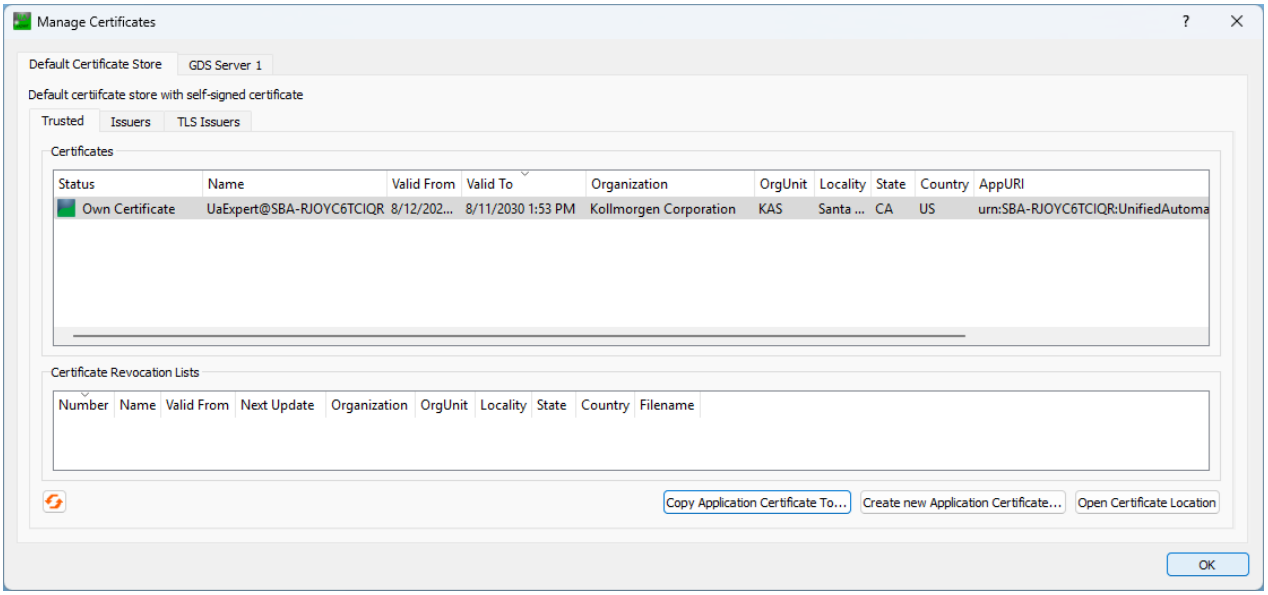


Figure 19-93: Manage Certificates dialog box

11. Click the **Copy Application Certificate To...** button to open the **Save As** dialog box. (Figure 19-94)

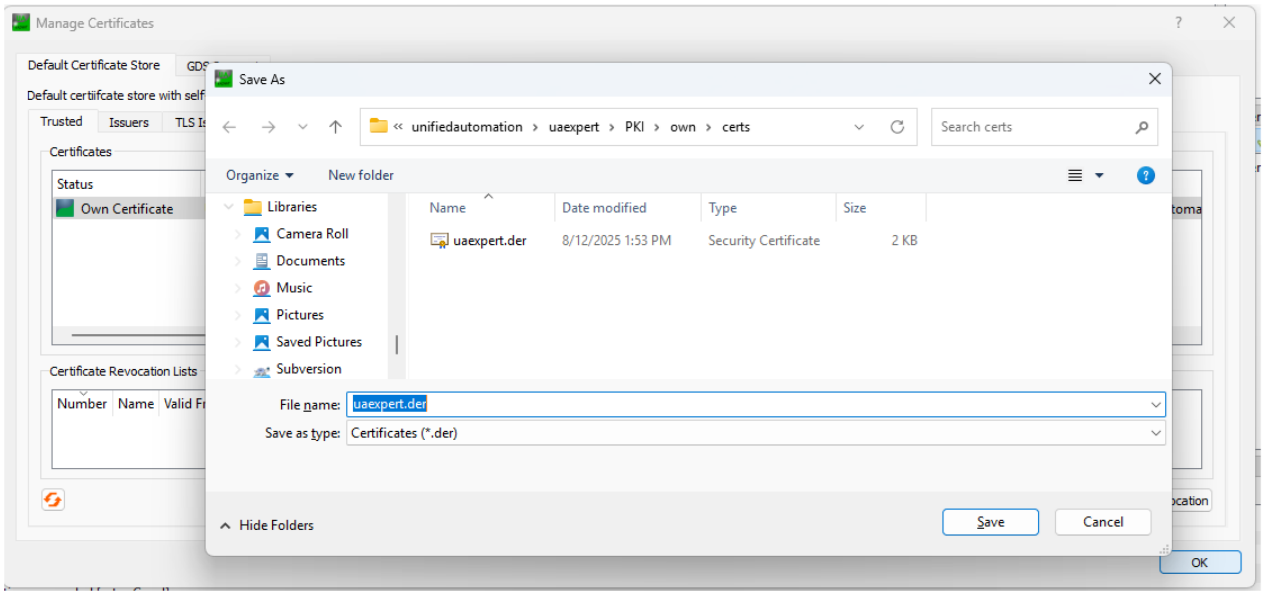


Figure 19-94: Save As dialog box

12. Save the file **uaexpert.der** to a temporary directory on the PC (e.g., C:\Temp).
13. Click **Save** to save the file in the designated location and close the dialog box.

14. Using the web browser opened in the "1. PCMM2G Configuration (with Certificates)" (➔ p. 965) section:
    a. Click the **Settings** tab.
    b. Click the **Certificates** tab. (Figure 19-95)

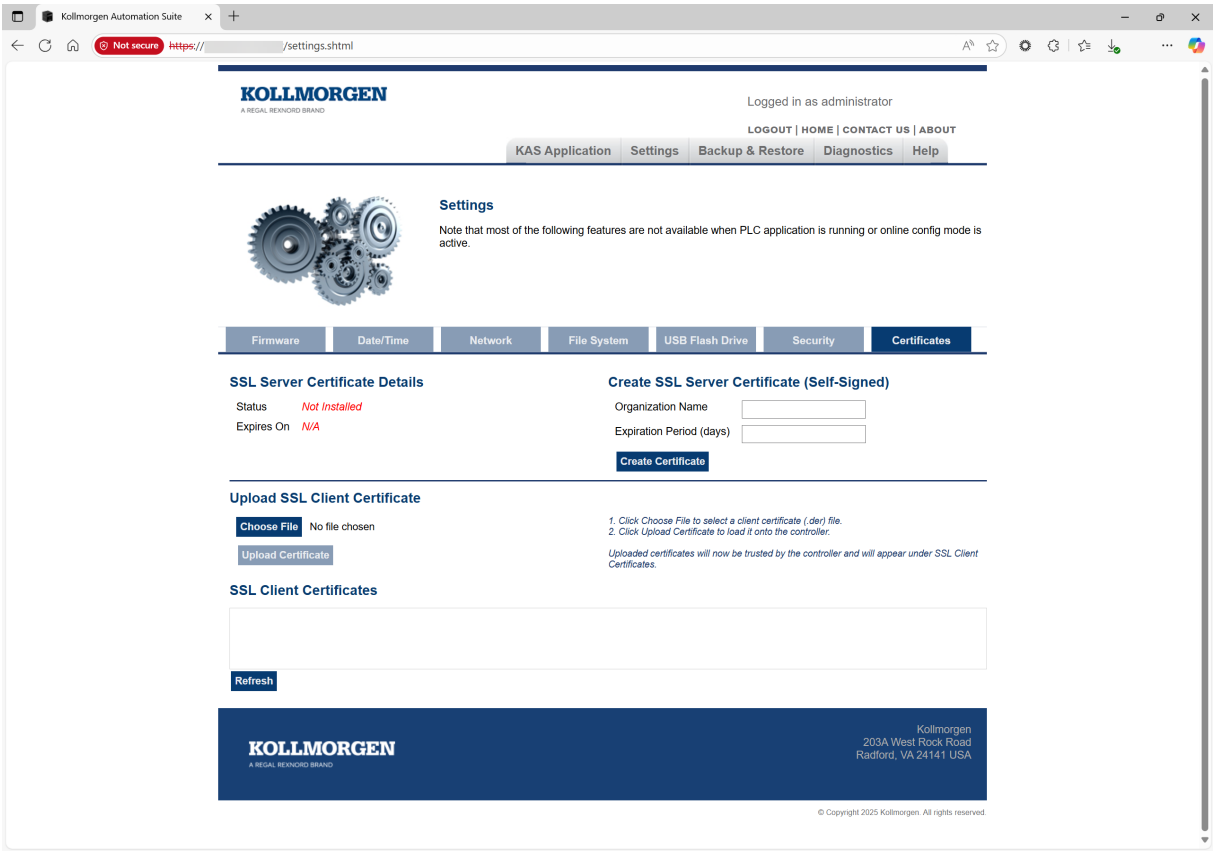


Figure 19-95: Certificates tab

    c. In the **Upload SSL Client Certificate** section, click the **Choose File** button.
    A File Explorer window opens.
    d. Locate and select the certificate file **uaexpert.der** that was saved in Step 12 in section "2. UaExpert Configuration (with Certificates)" (➔ p. 968).
    e. Click **Open** to confirm the selection and close the File Explorer window.
    f. Click the **Upload Certificate button** to upload the file to the PCMM2G.
        - The Certificated is added to the PCMM2G trusted store.
            - This allows OPC UA connections from UaExpert.
        - The file **uaexpert.der** appears under SSL Client Certificates explorer tree.
15. Continue with "3. UaExpert Connection to the PCMM2G (with Certificates)" (➔ p. 977).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.11 3. UaExpert Connection to the PCMM2G (with Certificates)

The procedure in this section connects UaExpert to the PCMM2G.

1. Verify the procedure in the "2. UaExpert Configuration (with Certificates)" (→ p. 968) section is completed.
2. On the toolbar, click the **Connect** button ( ) to connect to the server.
   A Connect Error message opens. (Figure 19-96)

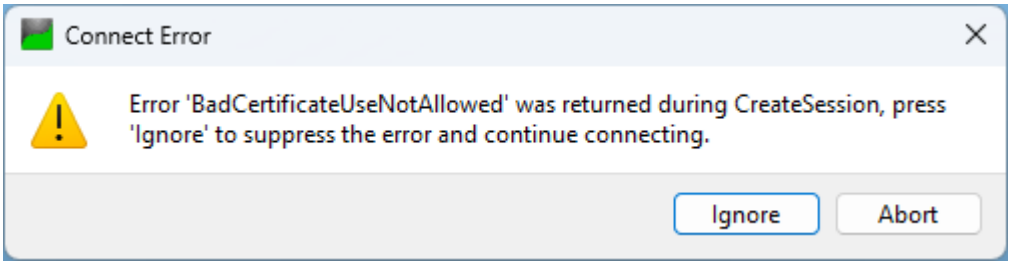


Figure 19-96: Connect Error message

3. Click **Ignore** to continue connecting.

   Data is now available from the KAS/PCMM2G OPC UA Server. (Figure 19-97)

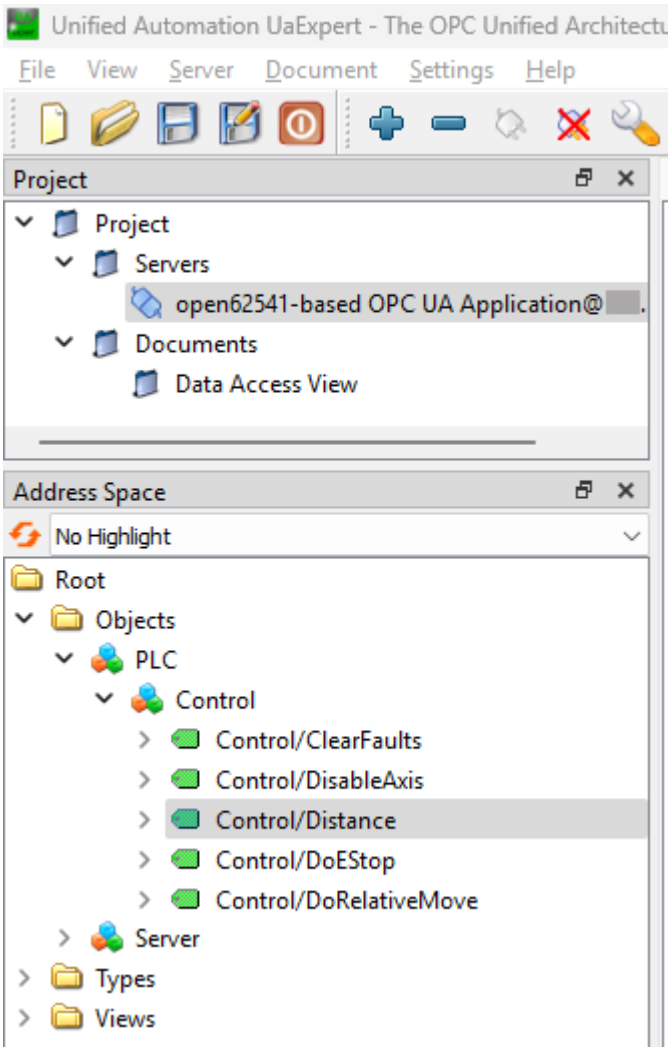


Figure 19-97: UaExpert - KAS OPC UA Data Objects

4. In the **Address Space** scroll box, select the applicable PLC object.
5. Drag-and-drop the selected objects to the **Data Access View** table. (Figure 19-98)

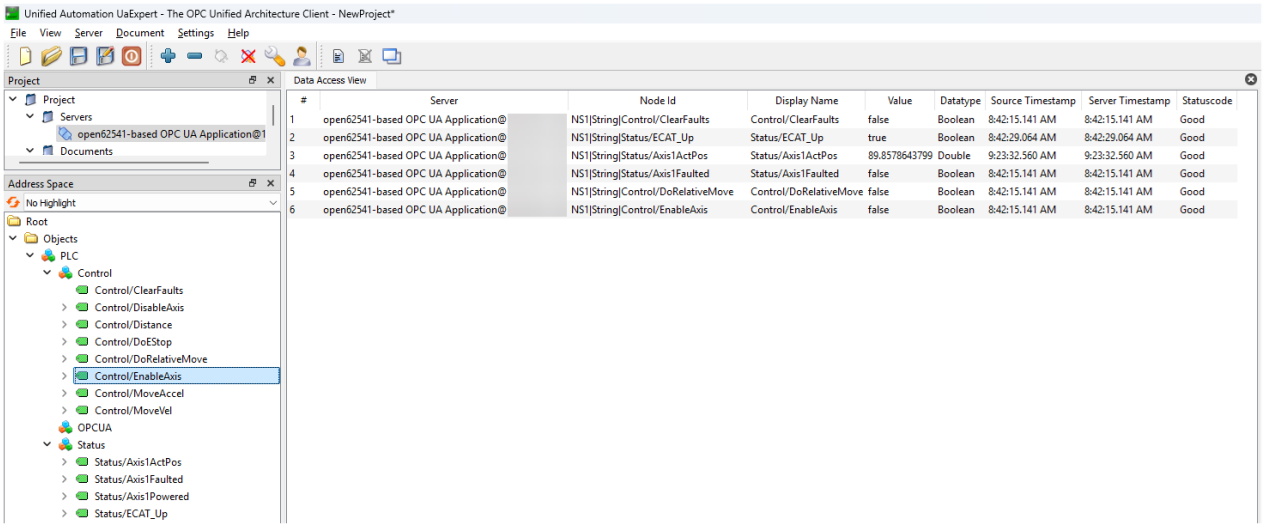


Figure 19-98: Data Access View table

6. Verify the values of PLC objects in the **Data Access View** table matches with the KAS project running on the PCMM2G.
7. In the **Data Access View** table, select a variable.
   The **Attributes** panel appears.
8. Scroll through the variables to locate a variable with an **AccessLevel** set to **CurrentWrite**. (Figure 19-99)

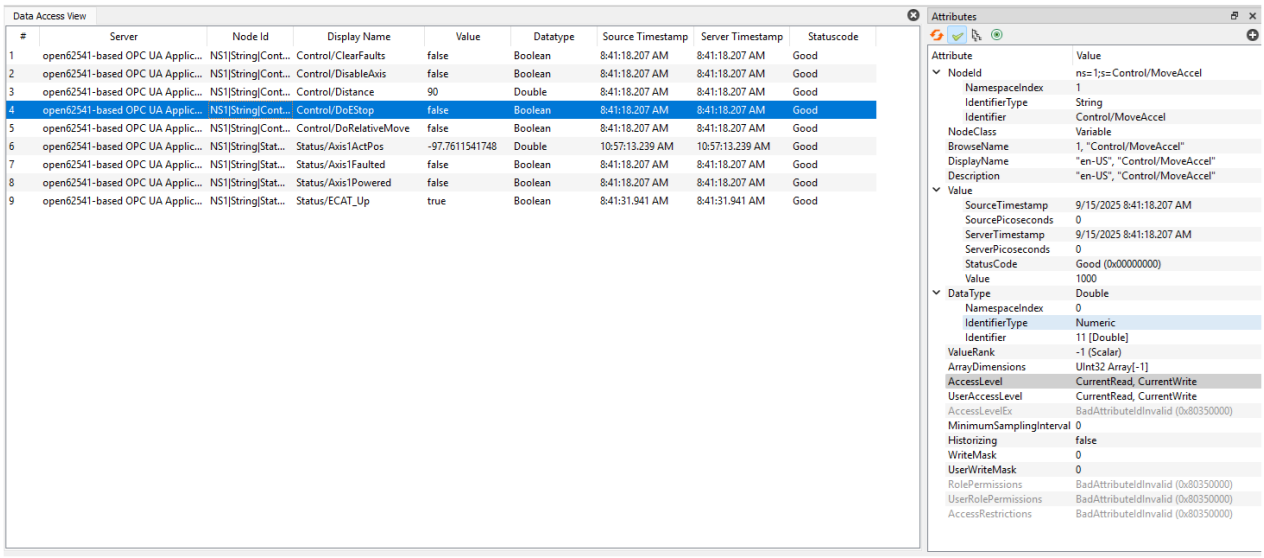


Figure 19-99: Attributes panel

9. Double-click the applicable **Value** column cell and change its value.
10. Switch to the KAS-IDE window.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

11. Navigate to the OPC UA Server Fieldbus Editor. (Figure 19-100)

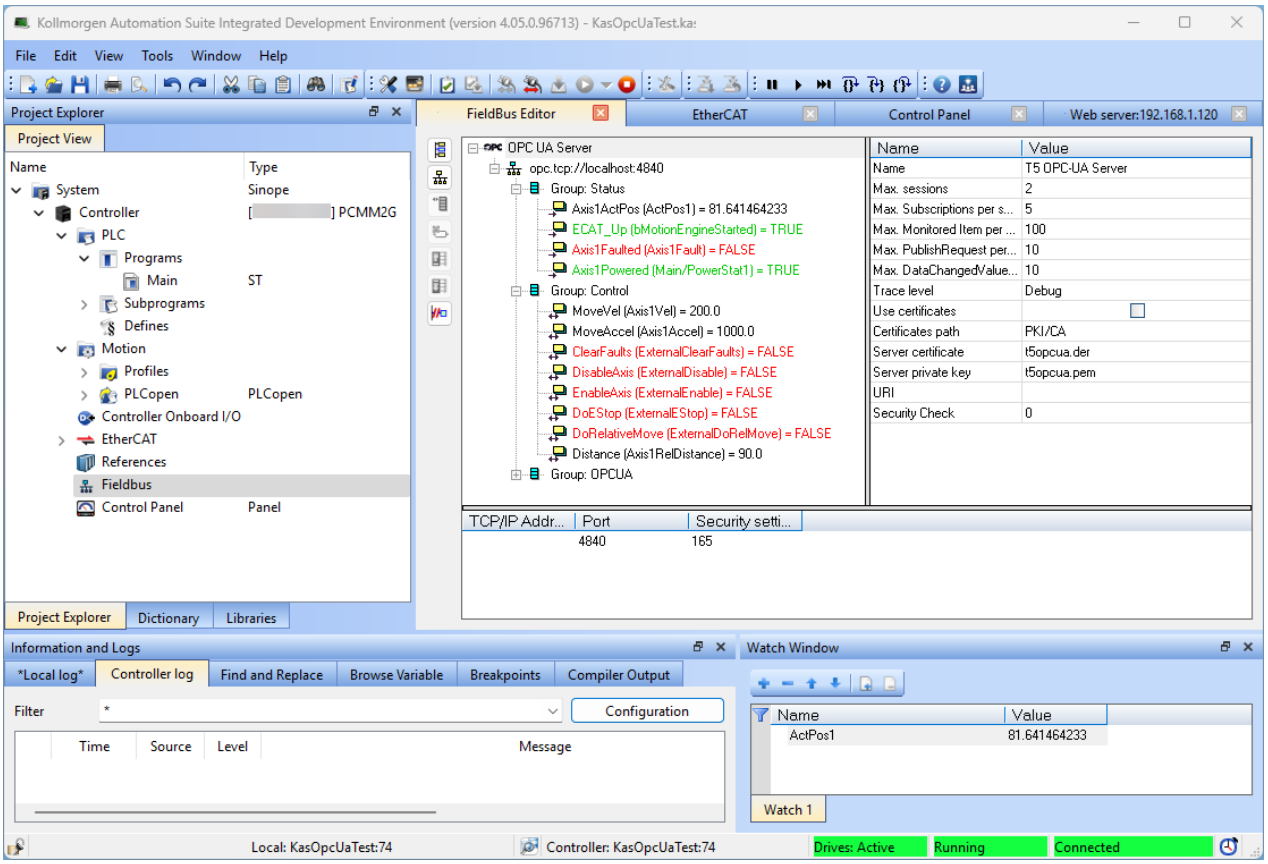


Figure 19-100: KAS Fieldbus Editor

12. Change a variable value in the KAS-IDE.
13. Verify the change in UaExpert.
    A re-browse of the variable value in UaExpert may be necessary to refresh the visible values.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

14. Select the PLC object and right-click to use **Rebrowse**. (Figure 19-101)

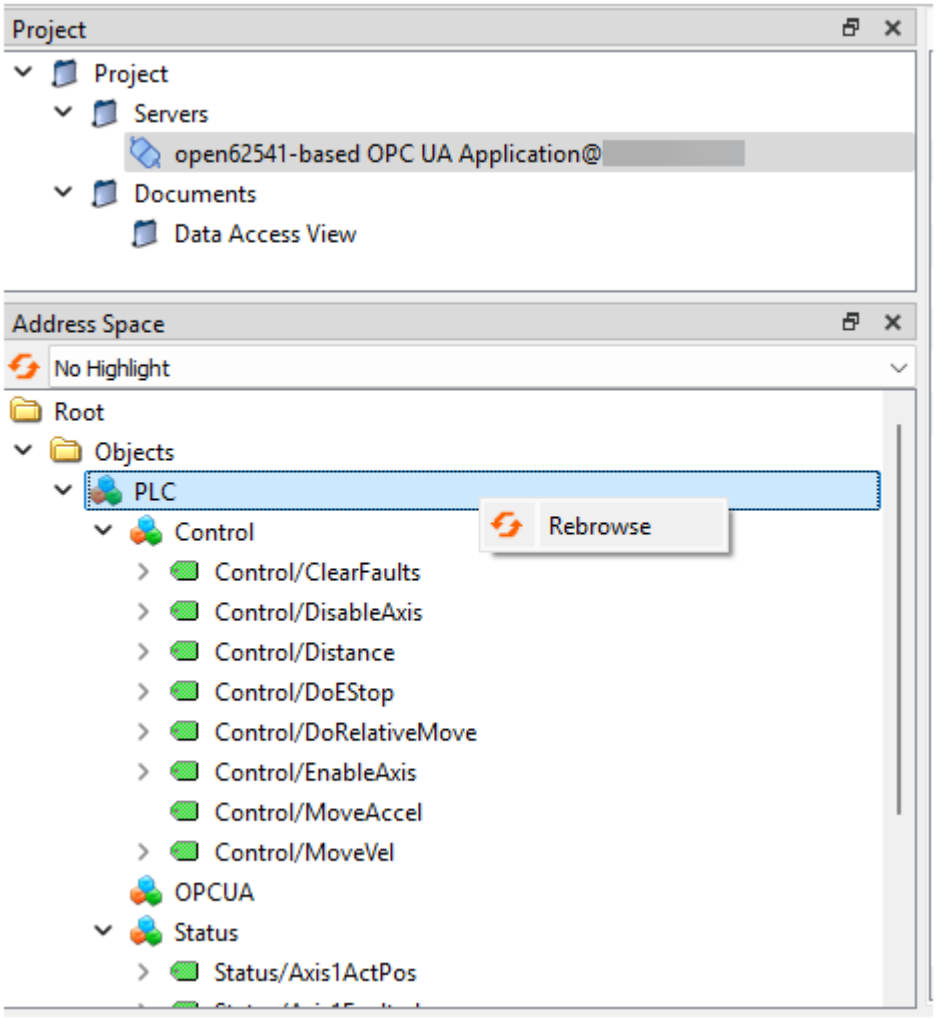


Figure 19-101: Rebrowse option

NOTE

- A maximum of 100 OPC UA client connections are accepted simultaneously.
- Established connections are disconnected if no data is exchanged within 30 seconds.

NOTE

- Uploaded certificates are automatically trusted by the OPC UA Server running on PCMM2G.
- The certificates enable secure communication when using Sign or SignAndEncrypt security modes.
- If the certificate is not uploaded, the PCMM2G OPC UA Server rejects the connection attempt from UaExpert.

See Also

- "OPC-UA - Establish a Server Connection without Certificates - UaExpert" (→ p. 980)
- "OPC UA - Establish Server Connection without Certificates - Exor HMI" (→ p. 993)
- "OPC UA - Troubleshooting" (→ p. 948)

15.11.0.0.1.1 OPC-UA - Establish a Server Connection without Certificates - UaExpert

These are the overall steps without using certificates for an OPC UA Server connection:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "1. PCMM2G Configuration (without Certificates)" (➔ p. 981)
- "2. UaExpert Configuration (without Certificates)" (➔ p. 983)
- "3. UaExpert Connection to the PCMM2G (without Certificates)" (➔ p. 989)

NOTE

Videos of these features are here: PCMM2G OPC UA Server Setup in KAS-IDE.

15.12 1. PCMM2G Configuration (without Certificates)

The procedure in this section configures the PCMM2G for connection to UaExpert.

1. Verify the "OPC-UA - UaExpert Installation" (➔ p. 965) procedure is completed.
2. Start the KAS-IDE.
3. Create a new project or open an existing KAS project.
4. Double-click an OPC UA Server node in the Fieldbus Editor project tree.
   The "OPC UA - Server dialog box" (➔ p. 960) opens. (Figure 19-102)

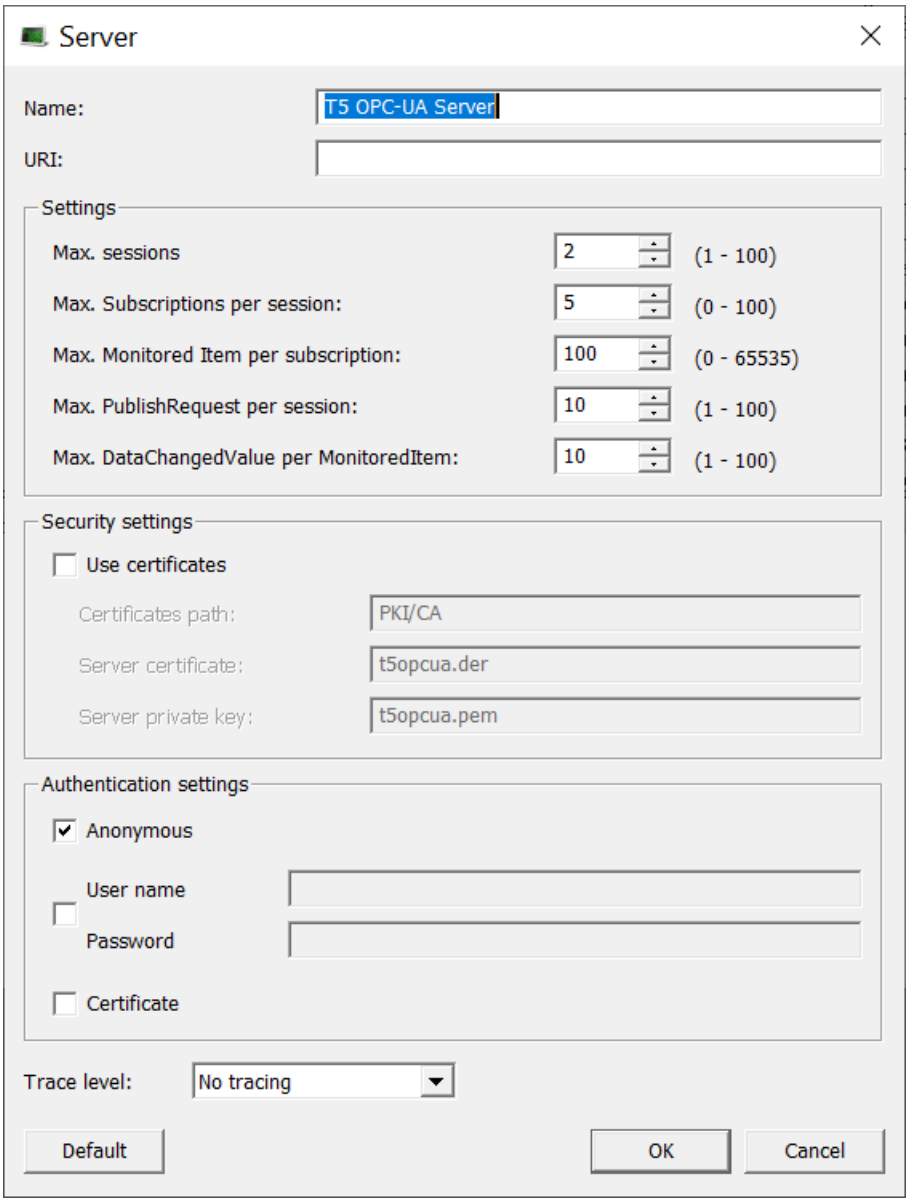


Figure 19-102: Server dialog box

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

5. In the **Authentication settings** section:
   a. Keep the **Anonymous** check box selection.
   b. Optional: Click the **User name/Password** check box to activate the **User name** and **Password** text boxes.
   c. If applicable, enter a **User name** and **Password** for the configuration.
   d. Click **OK** to save the changes or selections and close the dialog box.
   The Fieldbus Editor returns showing the added OPC UA Server node.
6. On the vertical toolbar, click the **Insert Master/Port** button ( ).
   The "OPC UA - Endpoint dialog box" (→ p. 953) opens. (Figure 19-103)

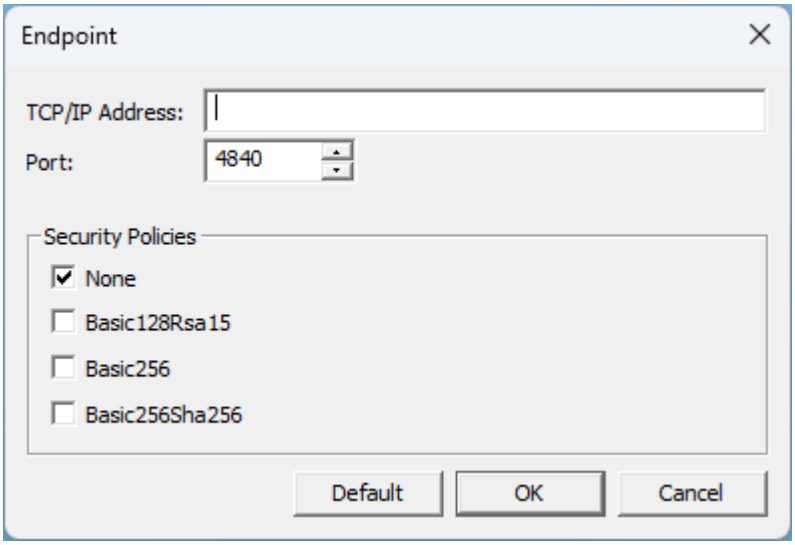


Figure 19-103: Endpoint dialog box

7. In the **Security Policies** section, accept the default selection of **None**.
8. Click **OK** to save the changes or selections and close the dialog box.
   The Fieldbus Editor returns.
9. Add applicable groups and variables using the "OPC UA - Add Multiple Groups and Variables to the Driver Configuration" (→ p. 941) procedure.
10. Compile and download the project to the KAS controller (e.g., PCMM2G).
11. Continue with "2. UaExpert Configuration (without Certificates)" (→ p. 983).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.13 2. UaExpert Configuration (without Certificates)

The procedure in this section configures UaExpert for connection to PCMM2G.

1. Verify the procedure in the "1. PCMM2G Configuration (without Certificates)" (→ p. 981) section is completed.
2. Start UaExpert.
   The UaExpert main window opens.
3. In the Project tree, right-click **Servers** and click **Add**. (Figure 19-104)

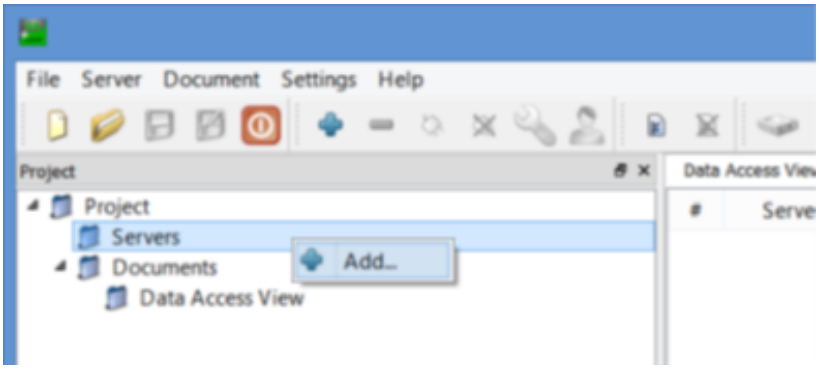


Figure 19-104: Add option

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The **Add Server** dialog box opens. (Figure 19-105)

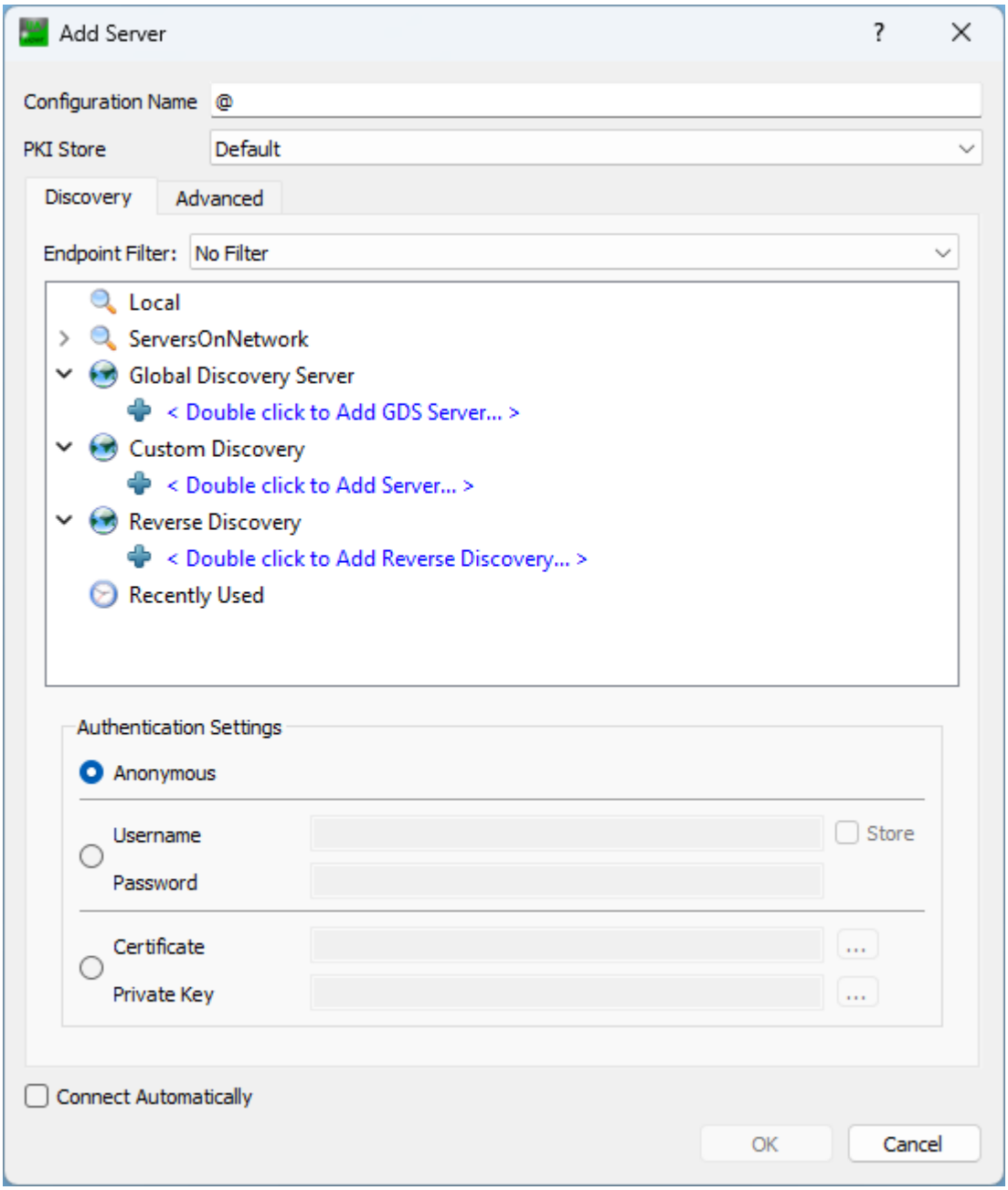


Figure 19-105: UaExpert Add Server dialog box

4. Under **Custom Discovery**, double-click the text.
   The **Enter URL** dialog box opens.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

5. Enter the URL as `opc.tcp://(PCMM2G_IP_Address)` to add the OPC UA Server. (Figure 19-106)

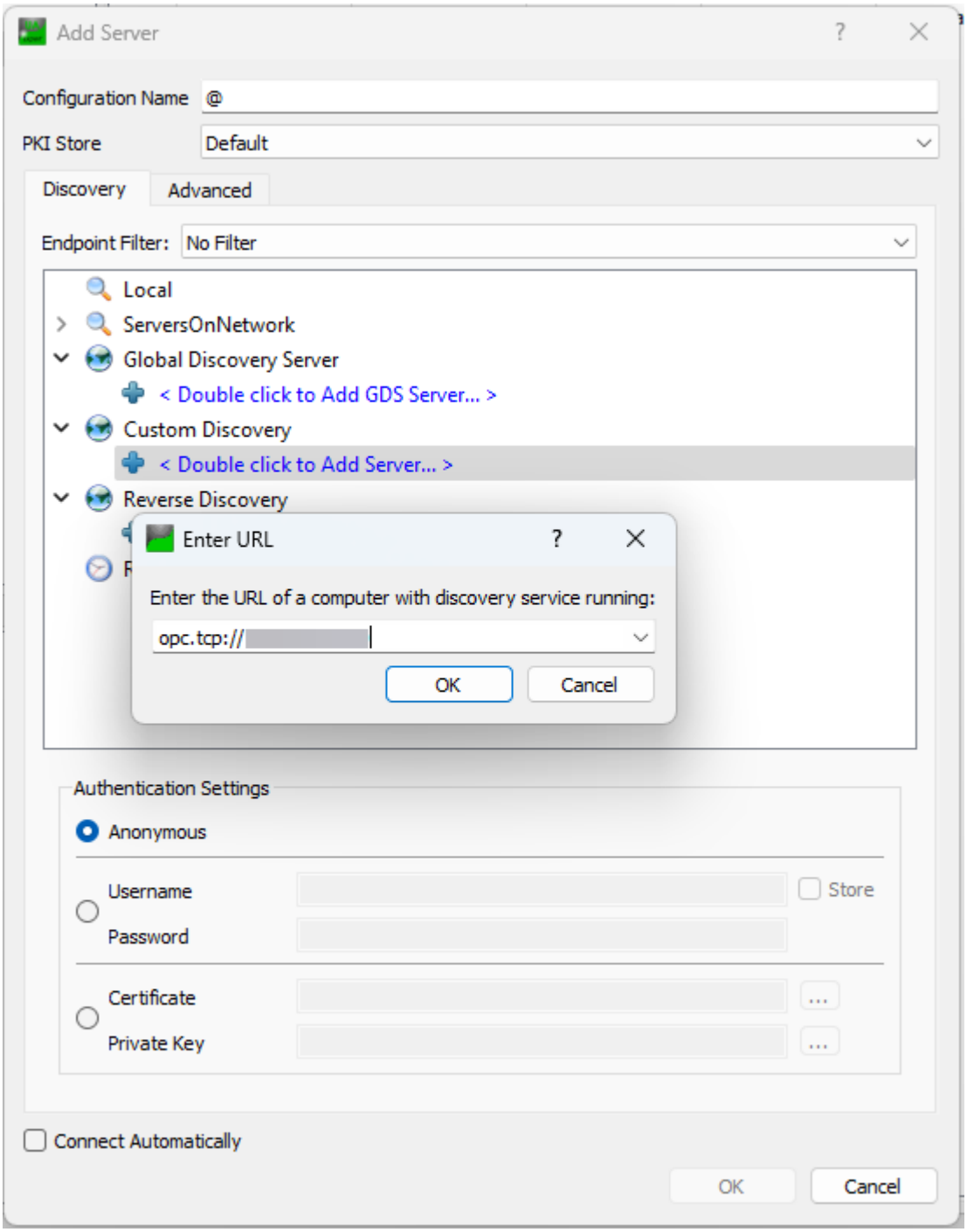


Figure 19-106: Enter URL dialog box

6. Click **OK** to save the changes or selections and close the dialog box.
The **Add Server** dialog box returns.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

NOTE

A Replace Hostname message may appear. (Figure 19-107)

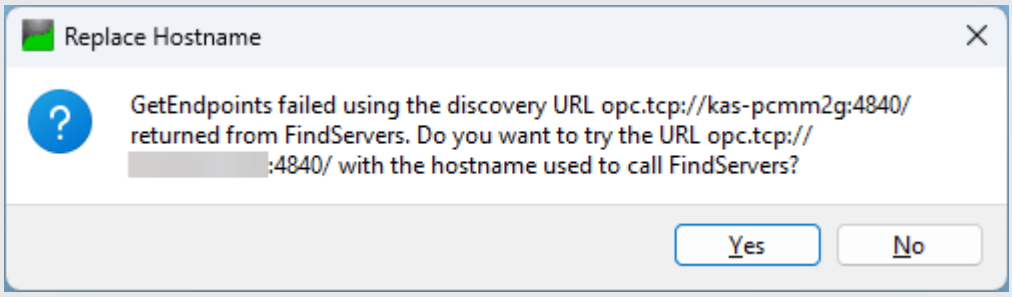


Figure 19-107: Replace Hostname message

This prompt appears because the PCMM2G supports OPC UA device discovery using its IP address, not by its hostname.

Click **Yes** to continue.

IMPORTANT

A **None** security policy **must be** configured for the OPC UA Server in the KAS project.

7. Under **Custom Discovery**, expand the newly added node.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

8. Under the opc.tcp://(PCMM2G_IP_Address), select **None**. (Figure 19-108)

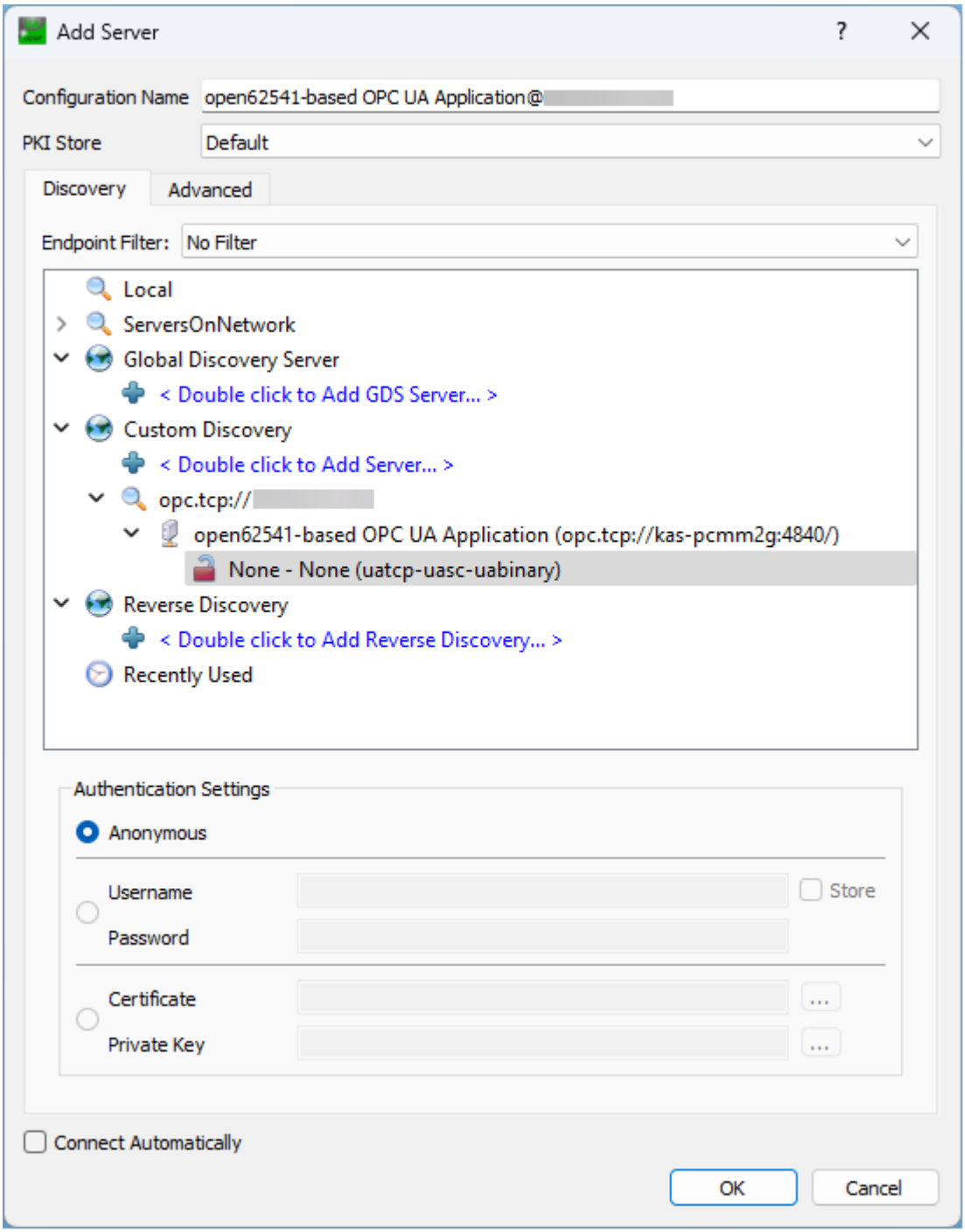


Figure 19-108: Add Server dialog box - None

- A Username and Password can be used if this Authentication Setting option is enabled in the KAS project. (Figure 19-109)
- The password is always encrypted if **Use Certificates** is enabled in the KAS project.

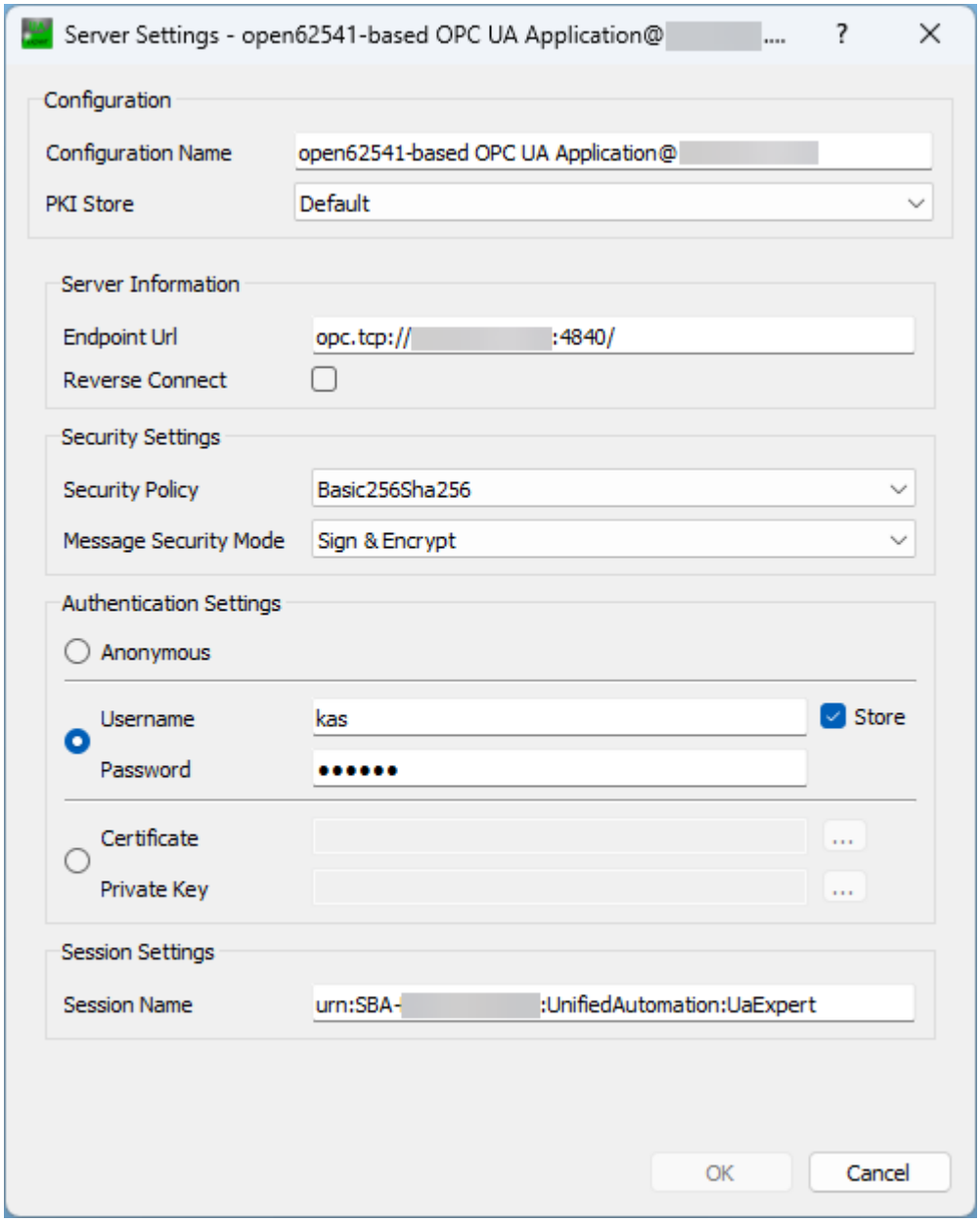


Figure 19-109: Server Settings dialog box - Username and Password

9. Click **OK** to save the changes or selections and close the dialog box.
   The server is now available in the UaExpert project tree.
10. Continue with "3. UaExpert Connection to the PCMM2G (without Certificates)" (➔ p. 989).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

15.14 3. UaExpert Connection to the PCMM2G (without Certificates)

The procedure in this section connects UaExpert to the PCMM2G.

1. Verify the procedure in the "2. UaExpert Configuration (without Certificates)" (→ p. 983) section is completed.
2. On the toolbar, click the **Connect** button ( ) to connect to the server.
   Data is now available from the KAS/PCMM2G OPC UA Server. (Figure 19-110)

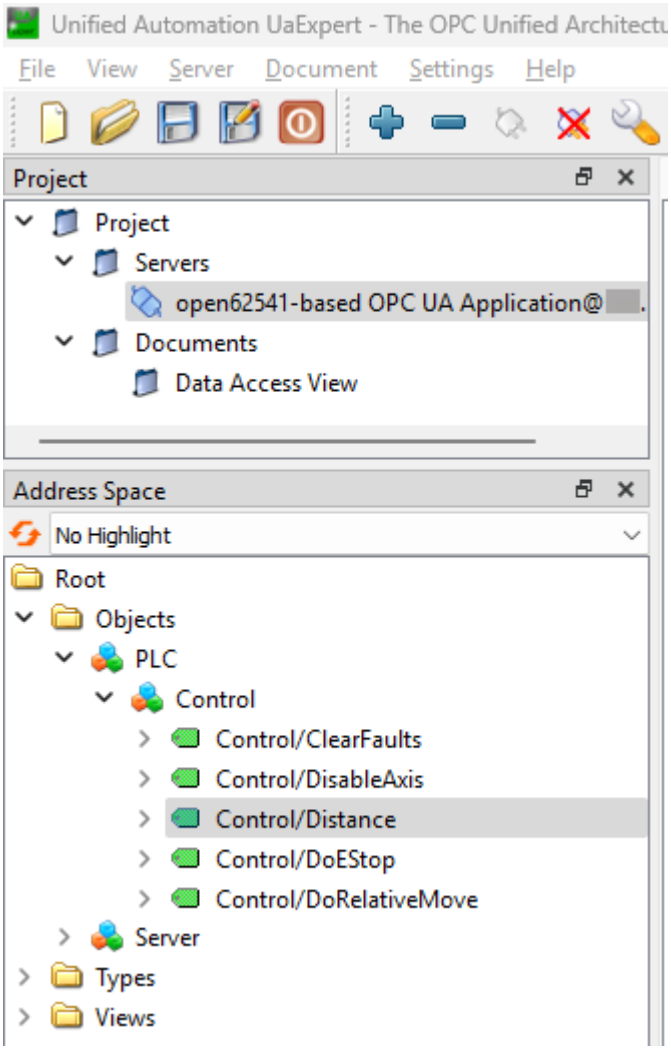


Figure 19-110: UaExpert - KAS OPC UA Data Objects

3. In the **Address Space** scroll box, select the applicable PLC object.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

4. Drag-and-drop the selected objects to the **Data Access View** table. (Figure 19-111)

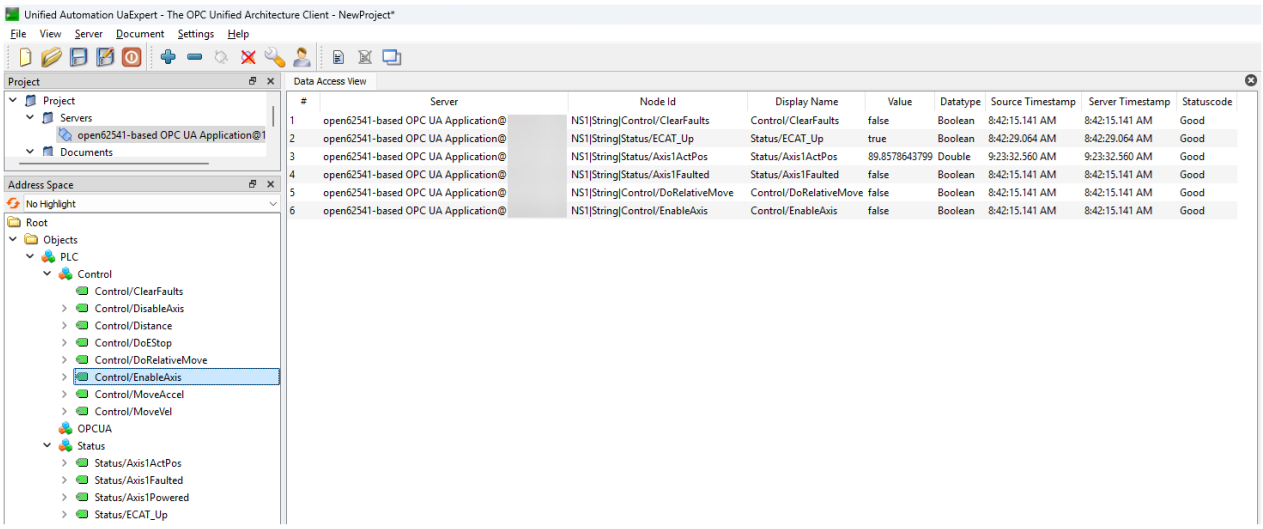


Figure 19-111: Data Access View table

5. Verify the values of PLC objects in the **Data Access View** table matches with the KAS project running on the PCMM2G.
6. In the **Data Access View** table, select a variable.
   The **Attributes** panel appears.
7. Scroll through the variables to locate a variable with an **AccessLevel** set to **CurrentWrite**. (Figure 19-112)

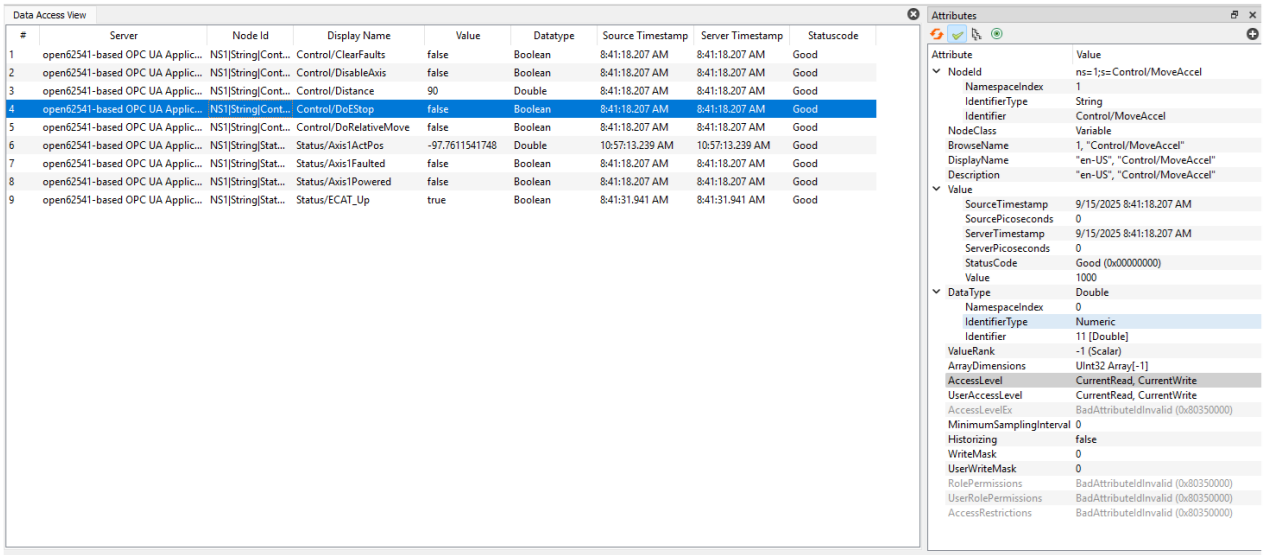


Figure 19-112: Attributes panel

8. Double-click the applicable **Value** column cell and change its value.
9. Switch to the KAS-IDE window.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

10. Navigate to the OPC UA Server Fieldbus Editor. (Figure 19-113)

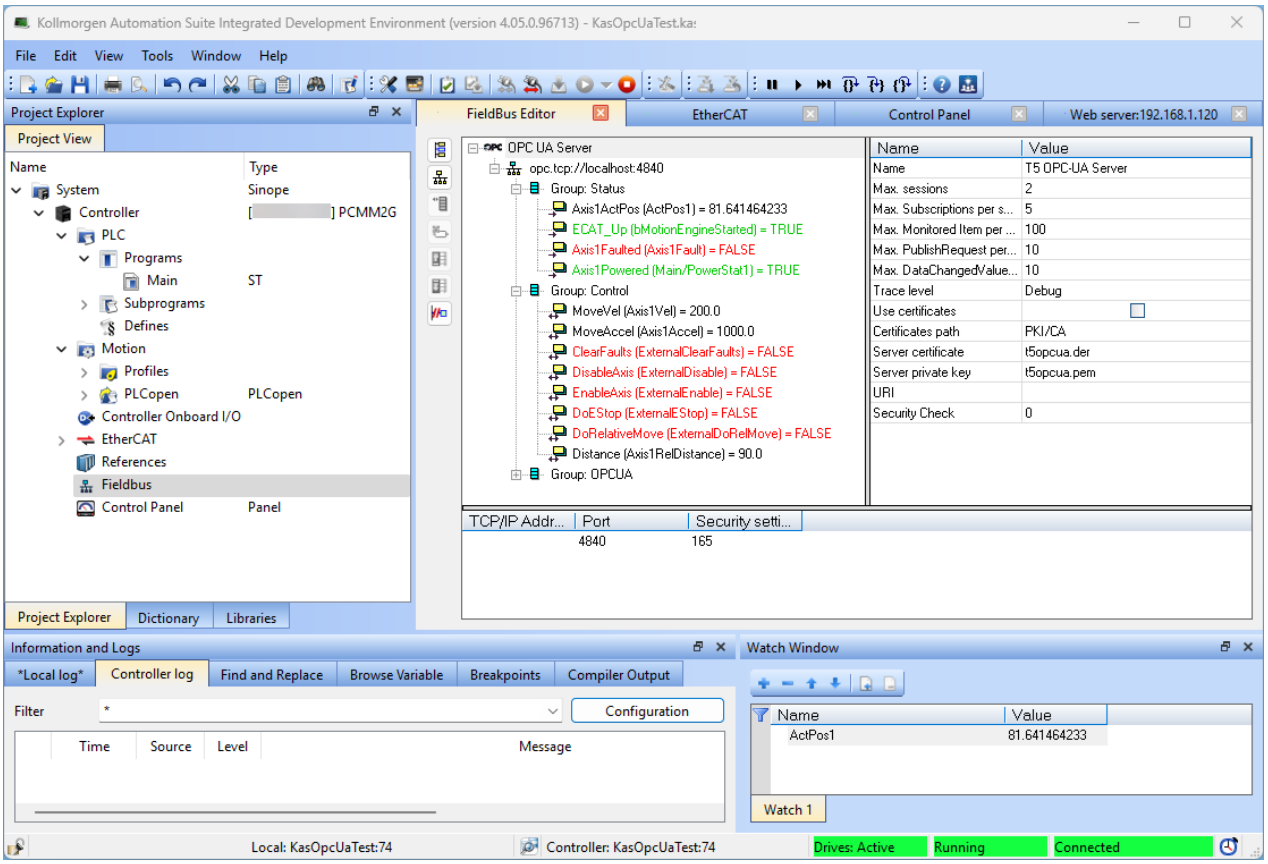


Figure 19-113: KAS Fieldbus Editor

11. Change a variable value in the KAS-IDE.
12. Verify the change in UaExpert.
    A re-browse of the variable value in UaExpert may be necessary to refresh the visible values.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

13. Select the PLC object and right-click to use **Rebrowse**. (Figure 19-114)

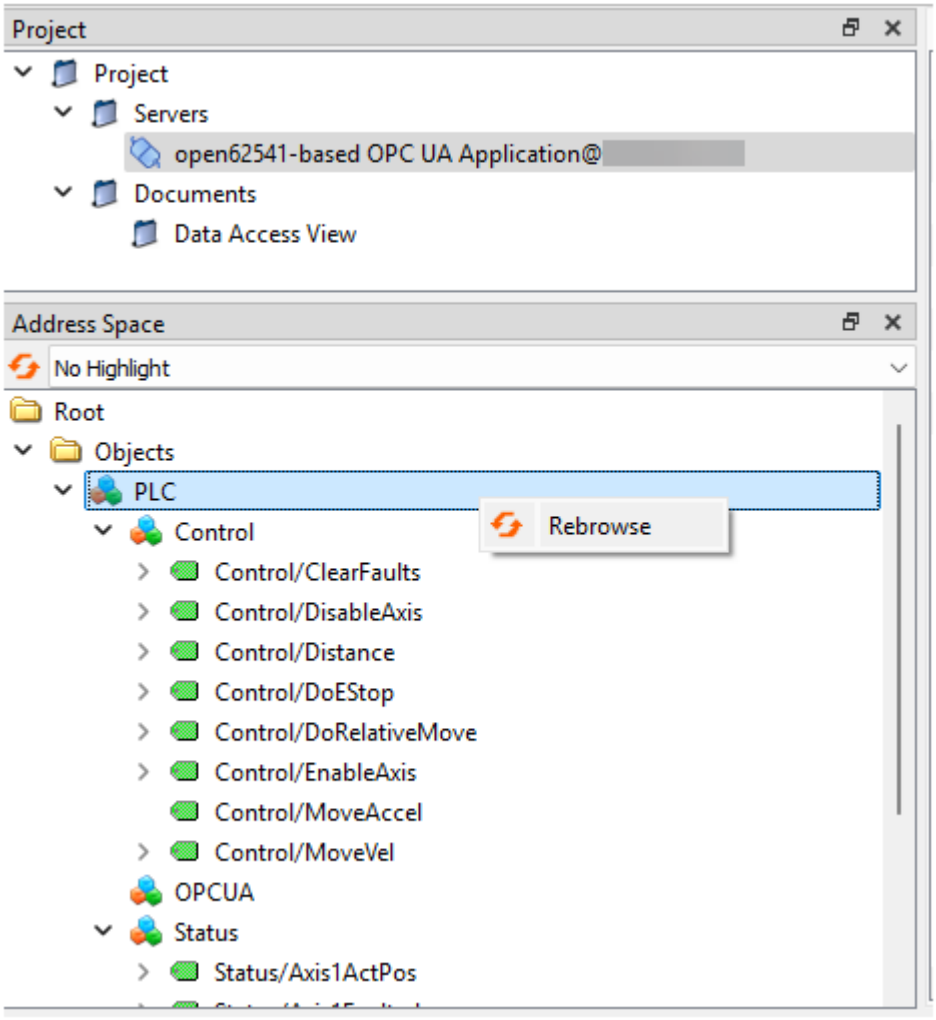


Figure 19-114: Rebrowse option

NOTE

- A maximum of 100 OPC UA client connections are accepted simultaneously.
- Established connections are disconnected if no data is exchanged within 30 seconds.

See Also

- "OPC-UA - Establish a Server Connection with Certificates - UaExpert" (→ p. 965)
- "OPC UA - Establish Server Connection without Certificates - Exor HMI" (→ p. 993)
- "OPC UA - Troubleshooting" (→ p. 948)

15.14.0.0.1 OPC UA - Establish a Connection using Exor HMI

These procedures are used to communicate with the PCMM2G OPC UA Server using the Exor HMI tool.

- "OPC UA - Establish Server Connection without Certificates - Exor HMI" (→ p. 993)

See Also

- "OPC UA - Troubleshooting" (→ p. 948)
- "OPC-UA - UaExpert Installation" (→ p. 965)

15.14.0.0.2.1 OPC UA - Exor HMI Installation

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

IMPORTANT

Registration is required to download JMobileStudio.

- JMobile Studio is a software application designed to create graphical HMI pages for Exor HMI products.
  - See Automation Products - EXOR Int.
- See JMobile Suite & Software Downloads to download the software and view initial start-up instructions.
  - Registration is required to download JMobileStudio.

15.14.0.0.3.2 OPC UA - Establish Server Connection without Certificates - Exor HMI

1. Verify the "OPC UA - Exor HMI Installation" (→ p. 992) procedure is completed.
2. Start the KAS-IDE.
3. Create a KAS project using an OPC UA Server configuration.
   See "OPC UA - Create New Driver Configuration" (→ p. 931).
   a. Create a new KAS project using any application template.
   b. Configure the controller for the project as a PCMM2G.
   c. In the KAS-IDE Fieldbus Editor, create a valid OPC UA Server configuration with “Anonymous” authentication setting enabled and several PLC variables assigned to it.
4. Compile and download the project to a PCMM2G controller.
5. Start the KAS project.
6. Launch JMobileStudio.
7. Start a new project for the Panel type you have available.
   This example procedure uses eSmart10. (Figure 19-115)

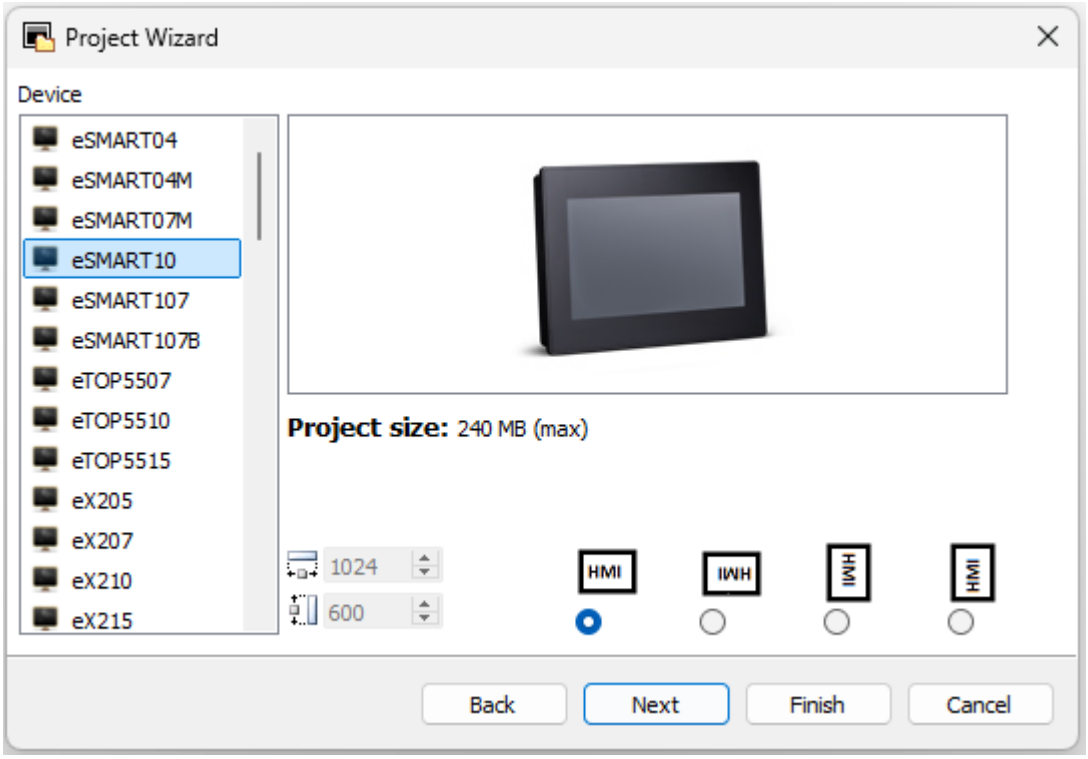


Figure 19-115: Exor Project Wizard - eSMART10

8. Click **Finish**.
   The wizard closes and the ProjectView window opens.
9. In the ProjectView tree, in the **Configuration** section, select **Protocols**.
10. On the ProjectView toolbar, click the + button to open the **Protocols** tab.
11. In the Protocols tab > **PLC** column, click the drop-down arrow and select **OPC UA Client**.
    The **OPC UA Client** dialog box opens.
12. In the **Host** text box, enter the PCMM2G IP address.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

13. Accept the defaults for all other parameters. (Figure 19-116)

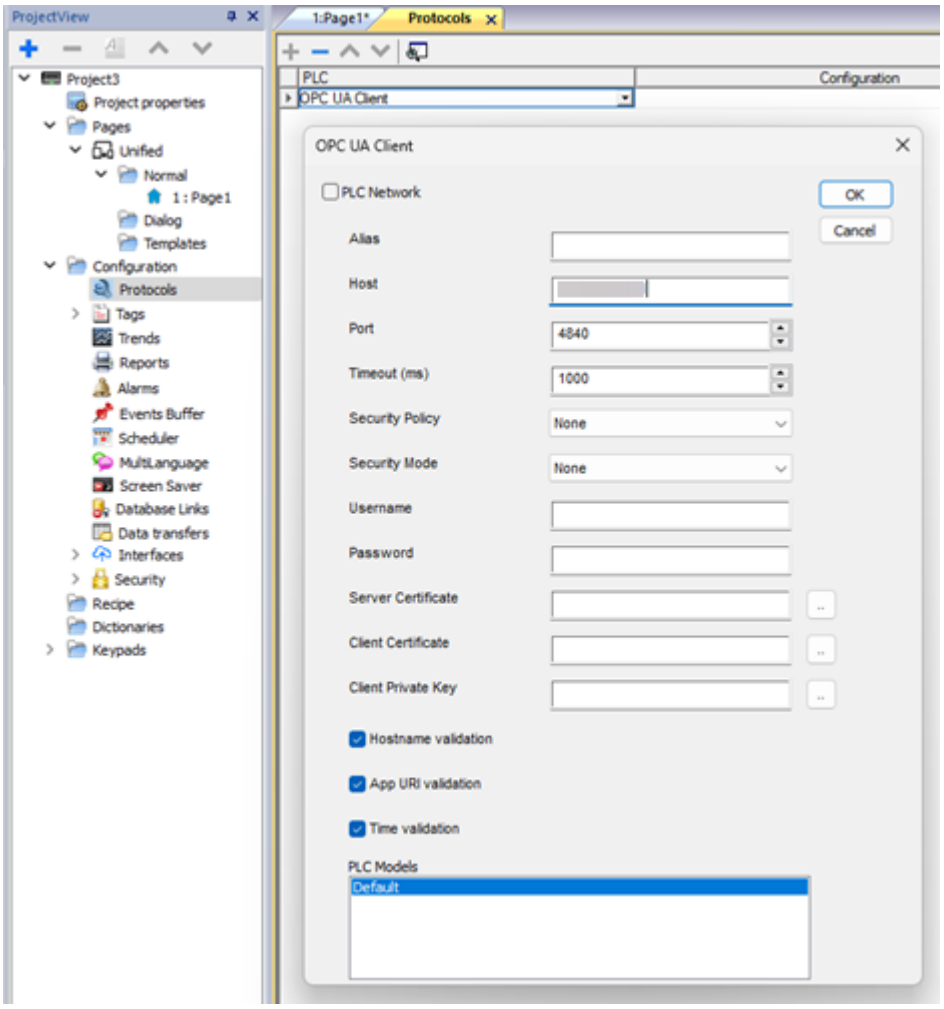


Figure 19-116: OPC UA Client dialog box

14. Click **OK** to save the changes or selections and close the dialog box.
    The ProjectView window returns.
15. In the ProjectView tree, in the **Configuration** section, select **Tags**.
    The **Tags** tab opens.
16. On the **Tags** toolbar, click the >] button.
    The HMISudio dialog box opens.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

17. Accept the default selection of OPC UA Discover v1.0. (Figure 19-117)

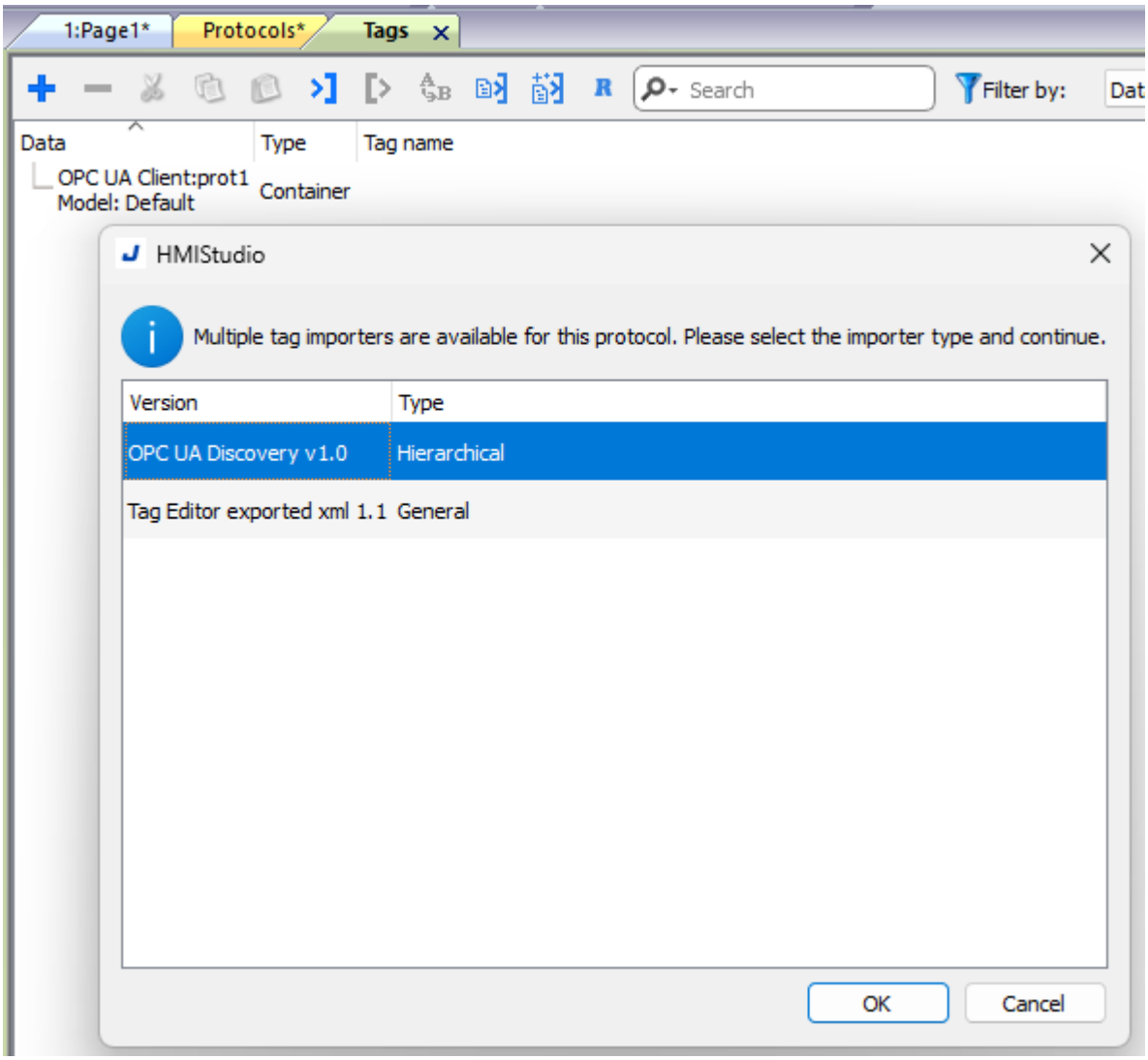


Figure 19-117: HMISudio dialog box

18. Click **OK** to save the changes or selections and close the dialog box.
This imports the dictionary from PCMM2G host.

The **OPCUA Client importer** dialog box opens. (Figure 19-118)

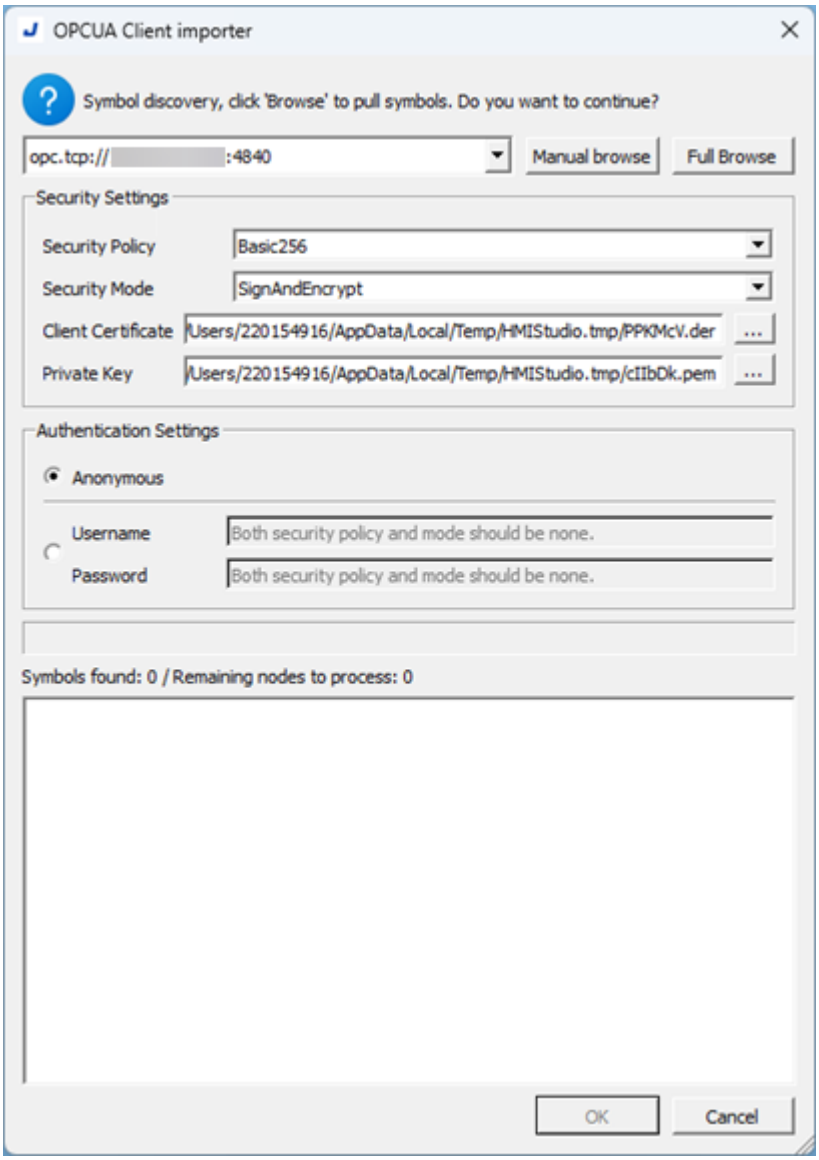


Figure 19-118: OPC UA Client Importer dialog box

19. Click the **Full Browse** button.
The **OPC UA Server Certificate** dialog box opens.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

20. Click the **Accept Temporarily** button. (Figure 19-119)

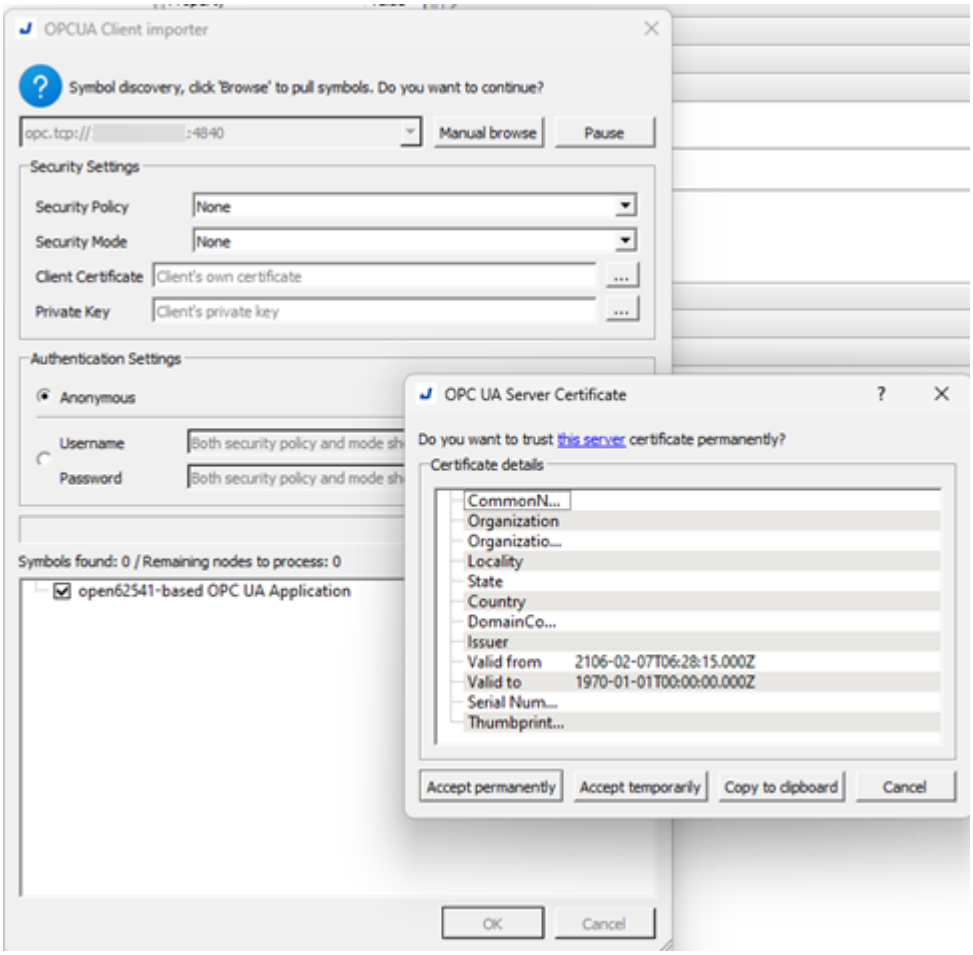


Figure 19-119: OPC UA Server Certificate dialog box

The **OPC UA Server Certificate** dialog box closes.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The **OPCUA Client importer** dialog box returns. (Figure 19-120)

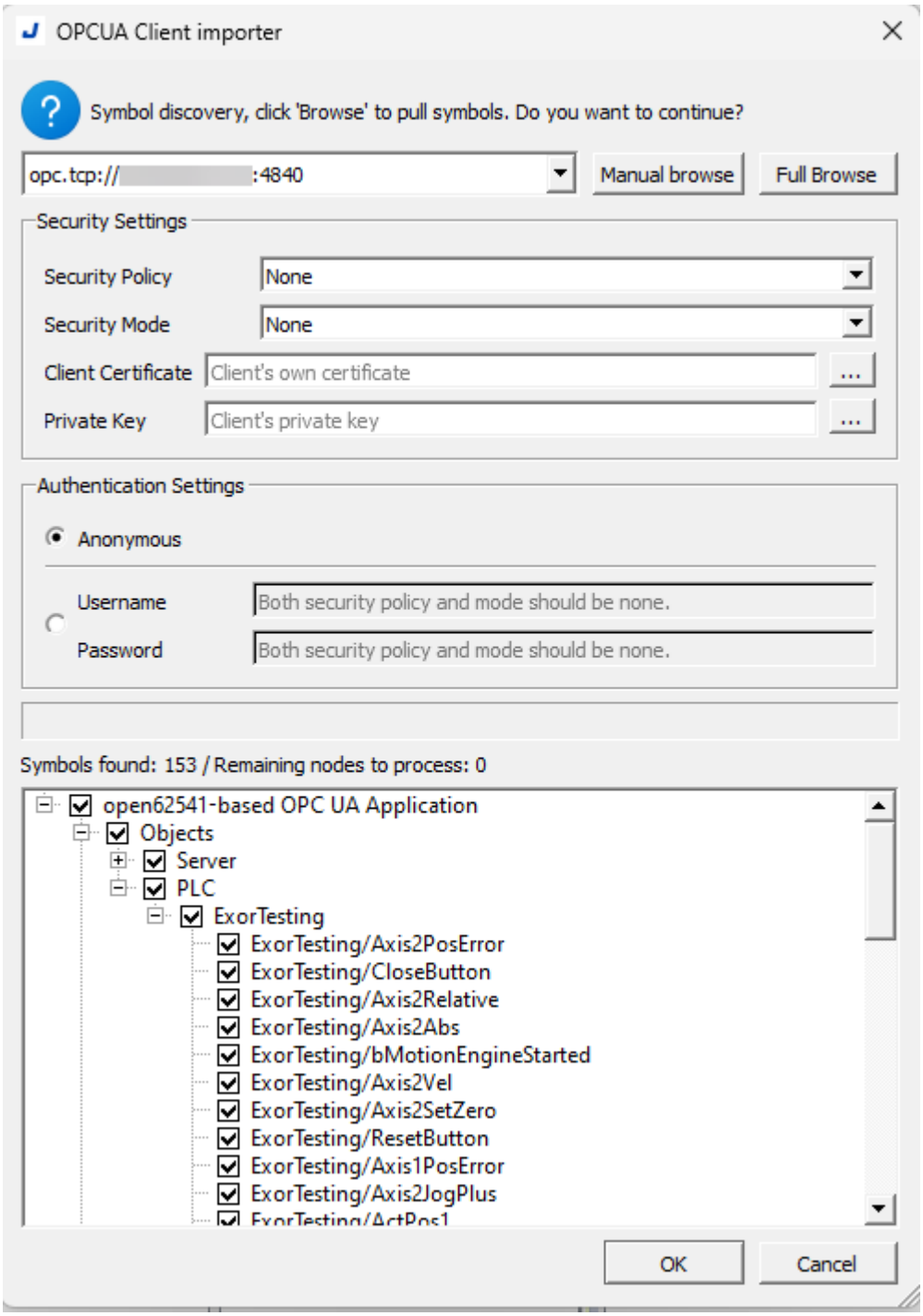


Figure 19-120: OPCUA Client importer dialog box

21. In the **Symbols found** scroll box, verify all Tags setup in the KAS-IDE Fieldbus Editor have been found.
22. Optional: Clear the check boxes of any variables that are not applicable.
23. Click **OK** to save the changes or selections and close the dialog box.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

If successful, a confirmation message appears.
This message shows the number of imported tags. (Figure 19-121)

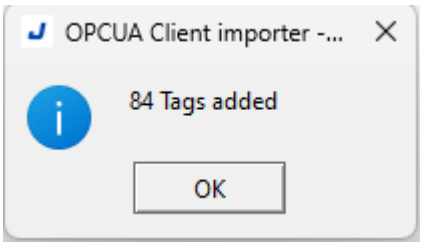


Figure 19-121: Imported tags confirmation message.

24. Click **OK** to close the confirmation message.

    The Tags tab is populated with all imported tags from the KAS-IDE Fieldbus Editor. (Figure 19-122)

    These Tags can be used with objects in the Exor panel screens.

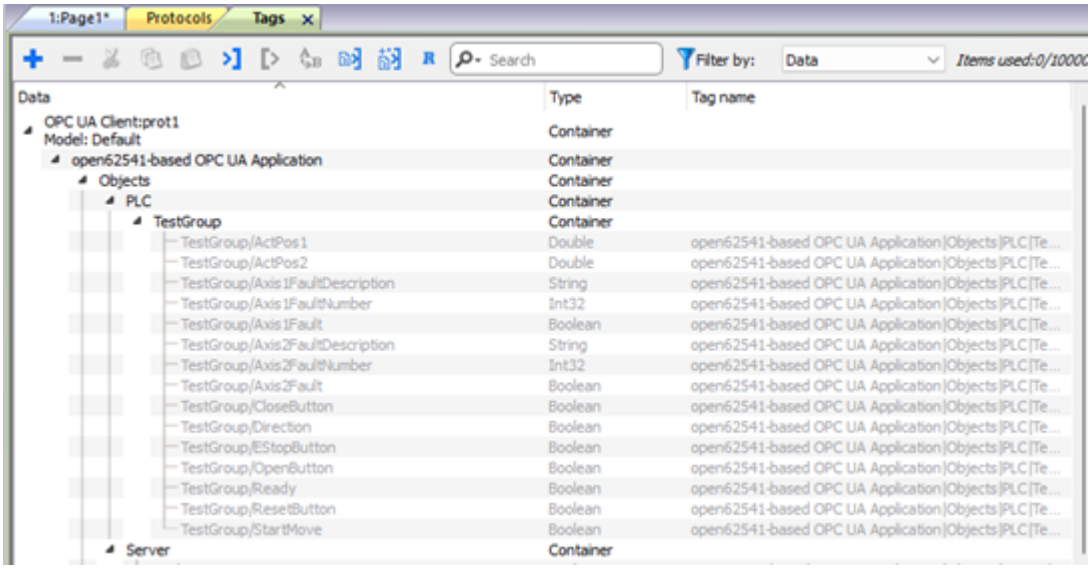


Figure 19-122: Tags tab populated with all imported tags

In the Page tab, the setup screens can now select object and add imported tag to DataLink parameter. (Figure 19-123)

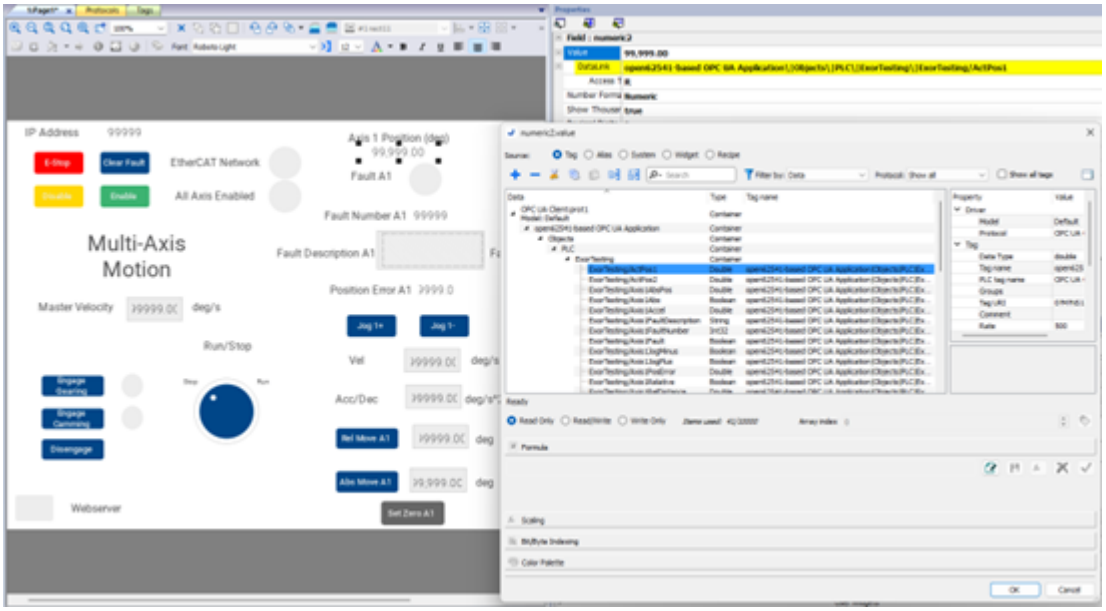


Figure 19-123: Example: 1:Page tab

Example: Figure 19-124 show the screen running an eSmart10 panel with objects mapped to tags through OPC UA.



Figure 19-124: Example: iSMART10 panel

See Also

- "OPC-UA - Establish a Server Connection with Certificates - UaExpert" (→ p. 965)
- "OPC-UA - Establish a Server Connection without Certificates - UaExpert" (→ p. 980)
- "OPC UA - Troubleshooting" (→ p. 948)

15.14.1 Modular EtherCAT Concept

A modular approach to machine design has many benefits, including flexibility, scalability, and improved efficiency.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- Modular machines are split into sub-sections.
- Each sub-section containing all the mechanical and electrical devices, motors, sensors, etc. as well as corresponding application code, to work together to provide a functional segment in the manufacturing workflow.
- The advantages to modular design are realized when the modules can be re-used.
- Examples include multiple module instances in the same machine and modules replicated across multiple machines.

Special consideration should be given to the physical layout, EtherCAT network topology, and the application code architecture to design a modular machine.

Modular machines are well suited to distributed EtherCAT device topologies (e.g., AKD-x-C/N drives) but can also be applied to other EtherCAT devices.

15.14.1.1 Examples of Modular Machine Design Uses

15.14.1.1.1 Machine with Multiple Variants

a. Machine Variant = Base module + one, two, (or more) additional modules.
b. Base Module - Each machine variant has the same base set of EtherCAT devices.
c. Additional Modules - Each module contains one (or more) EtherCAT devices.

15.14.1.1.2 Module Maintenance

a. One (or more) modules are removed during machine operation.
b. The machine continues to operate with the remaining base + modules.

15.14.1.1.3 Share Modules across Machines in a Factory Line

a. Remove a module from machine A.
b. Continue to operate machine A with the remaining base + modules.
c. Move the module to machine B.

15.14.1.2 Modular Design Guidelines

NOTE

- Select the PCMM2G controller Model (Essentials, 24-Axis or 64-Axis) that supports the total number of physical axes when the machine operates at its maximum configuration.

These sections provide guidelines for modular design:

Single project file (.kas) with maximum EtherCAT topology

1. Start with the full topology of EtherCAT devices planned for the maximum machine configuration.
2. Perform an EtherCAT scan to discover the full set of EtherCAT devices and save the project file.

The PLC code in the KAS project can be adapted to handle the cases where modules containing EtherCAT devices are disconnected and removed for different machine configurations, maintenance, etc.

PLCopen/Pipe Network Axis to drive map is the same for all machine variants

Map the axes to the drives for the maximum EtherCAT topology.

The axis to drive map remains consistent if drives are disconnected/removed.

Whether drives are disconnected and removed from the end or from the middle of an EtherCAT network, the axis to drive map relationship remain the same.

Axis is simulated for a disconnected drive

The PLC application code must be adapt to the disconnected/removed devices.

Because disconnected drives behave as simulated axes/drives, the application code can perform many normal axis operations without code adaptations.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This provides flexibility for how the application code is designed to handle disconnected/removed drives.

Safety devices should not be disconnected/removed

FSoE Masters may not support modular EtherCAT designs.

If a safety device is disconnected and removed, the FSoE master does not allow normal machine operation.

See ECATDeviceAction about how to add the ability to connect/disconnect an EtherCAT device to the program.

15.14.1.3 Backing up Modular EtherCAT Devices

This procedure should be followed to backup EtherCAT devices when the modular EtherCAT feature is used to remove devices to create a machine variant.

See "Modular EtherCAT Concept" (➔ p. 1000).

1. Run the KAS application which sets up the correct EtherCAT topology for the machine variant.
2. Stop the KAS application.

IMPORTANT

Do not power cycle or reboot the controller between Steps 2 and 3.

3. Backup the EtherCAT devices.
   See "Backup Procedure" (➔ p. 608).

15.14.1.4 Limitations

- The device topology order cannot be rearranged.
- The Modular EtherCAT features are not supported with imported ENI files.

What happens when the device providing feedback to the digitizing axis is removed?

The feedback position becomes zero if the device providing the Feedback Position for the Digitizing axis is removed using the ECATDeviceAction function block.

The feedback position data coming from the device is automatically connected to the digitizing axis when the device is reconnected to the EtherCAT network.

The Restore function only supports exact matches in the topology of backedup EtherCAT devices.

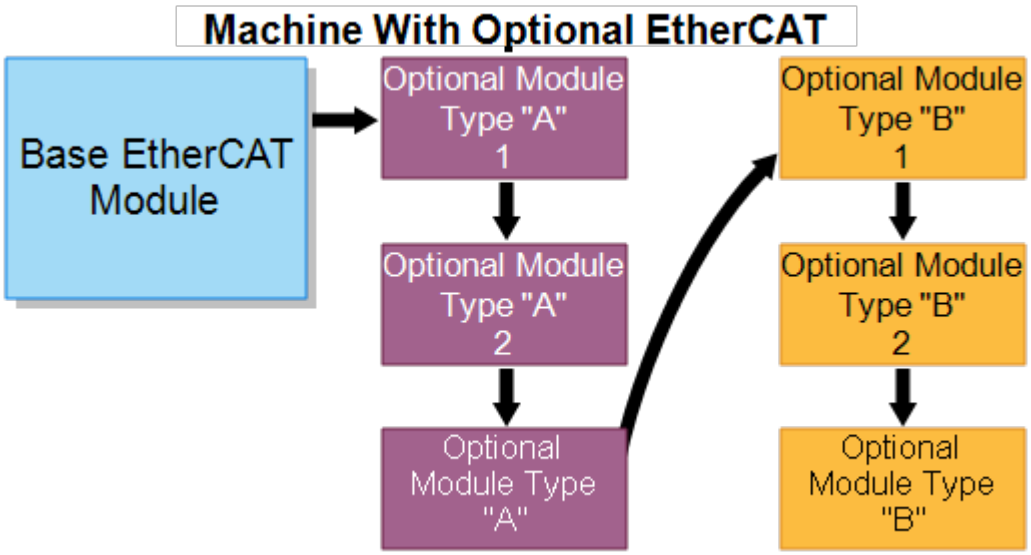
The Restore function only supports conditions where the network topology exactly matches the topology of the EtherCAT devices backup.

- There are a number of possible permutations and combinations when the Modular EtherCAT feature is used to create a machine with multiple variants.
- If a machine has a base EtherCAT module and a varying number of optional EtherCAT modules, then it is recommended that multiple backups are created:
  - A base EtherCAT module backup and an optional EtherCAT module backup.

Example

This image shows a base EtherCAT module and a varying number of optional type A modules and type B modules.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



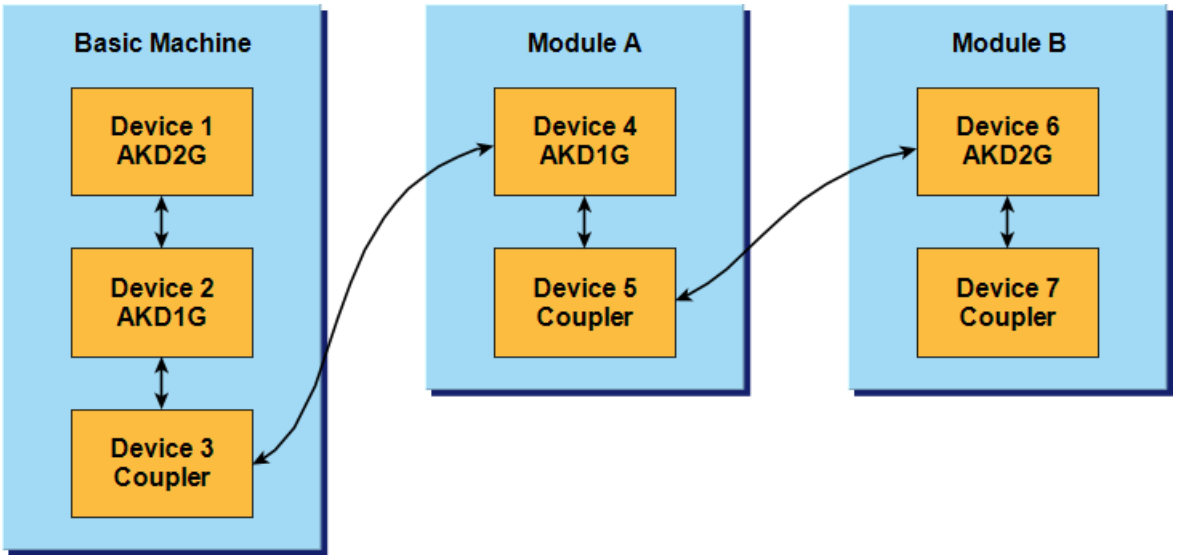
We recommend that EtherCAT Devices backups be made with the following configurations in order to meet different scenarios.

- Backup #1: Base EtherCAT Module + one of Optional Module Type A.
- Backup #2: Base EtherCAT Module + one of Optional Module Type B.

With the two EtherCAT Devices backups, you may be able to restore various combinations by removing the other Optional Modules, keeping only the module you need to restore.

15.14.1.5 Create an Application with Modular EtherCAT

This section discusses how to create an application with modular EtherCAT for various scenarios. We will use the following configuration to describe the scenarios and code snippets. The configuration consists of a full machine with two different modules. The full EtherCAT network consists of seven devices.



15.14.1.5.1 Create a Machine Variant with the Basic Machine

1. Start with a full topology of EtherCAT devices.
   In this example we have Device1 to Device7.
   See "Modular Design Guidelines" (➔ p. 1001).
2. After initializing the motion engine use the ECATDeviceAction function block to disconnect Device 4, Device

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

5, Device 6, and Device 7 **before** calling the MLMotionStart function block.
3. After the devices are disconnected, call **MLMotionStart** and finish the code.

15.14.1.5.2 ST Language Example

```
/////MOTION INITIALIZATION////////////////////////////////////////////////

FirstCycle := TRUE;

On FirstCycle DO //Initialize the motion engine
     MLMotionInit( 1000);
     StatusMessage := 'Initializing Motion Engine';
END_DO;

MotionEngineStatus := MLMotionStatus();//Check the current status of the
motion engine

ON MotionEngineStatus = MLSTATUS_INITIALISED DO//Once motion engine is
initialized, create CAM profiles and defined Axis, then start the motion
engine
     If not FirstInitComplete then
          Profiles( MLPR_CREATE_PROFILES );
          PLCopen( 0 );
          // Disconnect Device 4 through Device 7
          Inst_ECATDeviceAction(True, EtherCAT.Device_4 , DEVICE_ACTION_
DISCONNECT);
          if Inst_ECATDeviceAction.Error then
               //Handle the Error here
          End_if;
          // Repeat the steps done for Device 4 to disconnect Device 5,Device
6, Device 7
          // After disconnecting all the devices that are not part of this
machine variant, start the motion engine.
          MLMotionStart();
          FirstInitComplete := True;
     End_If;
END_DO;
```

15.14.1.5.3 Add a Module

Use this procedure when a module needs to be added.

In this example we will add Module A of our example project.

1. Stop the motion engine using MLMotionStop.
2. Verify that motion engine is stopped.
3. Use the action “DEVICE_ACTION_CONNECT” in ECATDeviceAction to connect the EtherCAT devices (Device 4 and Device 5) corresponding to the module.
4. Start the motion engine using MLMotionStart.

15.14.1.5.4 ST Language Example

```
// To Add a Module first stop the Motion Engine.
    MLMotionStop();
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
    MotionEngineStatus := MLMotionStatus();//Check the current status of the
motion engine

ON MotionEngineStatus = MLSTATUS_STOPPED DO

     // Connect Device 4 and Device 5 to add Module A
     Inst_ECATDeviceAction(True, EtherCAT.Device_4, DEVICE_ACTION_CONNECT);
     If Inst_ECATDeviceAction.Error then
          //Handle the Error here
     End_if;
     Inst_ECATDeviceAction(True, EtherCAT.Device_5, DEVICE_ACTION_CONNECT);
     If Inst_ECATDeviceAction.Error then
          //Handle the Error here
     End_if;

     // Start the Motion Engine
     MLMotionStart();

END_DO;
```

15.14.1.5.5 Remove a Module

Use this procedure when a module needs to be removed.

In this example we will add Module A of our example project.

1. Stop the motion engine using MLMotionStop.
2. b. Verify that motion engine is stopped.
3. Use the action “DEVICE_ACTION_DISCONNECT” in ECATDeviceAction to disconnect the EtherCAT devices (Device 4 and Device 5) corresponding to the module.
4. Start the motion engine using MLMotionStart.

15.14.1.5.6 ST Language Example

```
// To remove a Module first stop the Motion Engine.
     MLMotionStop();
     MotionEngineStatus := MLMotionStatus();//Check the current status of the
motion engine

ON MotionEngineStatus = MLSTATUS_STOPPED DO

     // Disconnect Device 4 and Device 5 to add Module A
     Inst_ECATDeviceAction(True, EtherCAT.Device_4, DEVICE_ACTION_
DISCONNECT);
     If Inst_ECATDeviceAction.Error then
          //Handle the Error here
     End_if;

     Inst_ECATDeviceAction(True, EtherCAT.Device_5, DEVICE_ACTION_
DISCONNECT);
     If Inst_ECATDeviceAction.Error then
          //Handle the Error here
     End_if;
```


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

```
    // Start the Motion Engine
    MLMotionStart();

END_DO;
```

15.14.2 Controller Performance Guidelines

- "EtherNet/IP and PROFINET" (➔ p. 1006)
- "OPC UA" (➔ p. 1007)

15.14.2.1 EtherNet/IP and PROFINET

Time to share the data over EtherNet/IP or PROFINET does not increase as more variables are shared unless either the:

- EtherNet/IP Requested Packet Interval (RPI) time is changed.
- PROFINET Send clock factor is changed.
- PCMM2G processor is overloaded (i.e., no idle time).

Additional controller CPU usage is required when fieldbuses are used to transmit and receive data between network devices and PLC variables.

The impact on the PLC execution time and the CPU load is estimated as:

```
EtherNet/IP or PROFINET Processing Time = PLC Overhead + (Bytes Shared x
Load per Byte)
```

These models are used to calculate the performance of the EtherNet/IP and PROFINET fieldbuses.

- "PCMM2G" (➔ p. 1006)

Notes

- The **Load per Byte** is dependent on both the:
  - EtherNet/IP RPI.
  - PROFINET Send clock factor.
- The **Load per Byte** values are measured with both:
  - EtherNet/IP: RPI = 10.
  - PROFINET: Send clock factor [31,250us] = 32.

15.14.2.1.1 PCMM2G

PCMM2G as an:	PLC Overhead	Load per Byte
EtherNet/IP Adapter (Server)	15 microseconds	0.0922 microsecond/Byte
EtherNet/IP I/O Scanner (Client)	10 microseconds	0.0753 microsecond/Byte
PROFINET IO Controller	22 microseconds	0.156 microsecond/Byte
PROFINET IO Device	13 microseconds	0.119 microsecond/Byte

15.14.2.1.2 Example: EtherNet/IP Performance

This example uses a PCMM2G as an I/O Scanner with 8x Assemblies with:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- 3968 Byte PLC variables all with RPI 10ms.
- 4000 microsecond EtherCAT Cycle Time.

```
PLC Time = (3968 x 0.0922) + 15 = 381 microseconds
CPU Usage % = 381 / 4000 = 0.095 (9.5%)
```

Results of exchanging data over EtherNet/IP:

- Increases the PLC execution time by 381 microseconds.
- Reduces the CPU Idle Time by 9.5%.

15.14.2.2 OPC UA

Sharing data using OPC UA does not introduce additional PLC overhead or increase PLC execution time.

- When EtherCAT operates at a 4 kHz (250 µs) cycle time and large OPC UA data transfers occur, applications may occasionally experience missed:
  - Receive frames (A38).
  - PLC cycles.

15.14.2.2.1 Recommendations to Reduce Missed Receive Frames and PLC Cycles

Limit OPC UA Transfer Size

- Break large data sets into smaller chunks.
- Reduce the frequency of bulk transfers.

Optimize EtherCAT Cycle Time

If possible, use a 500 µs or larger EtherCAT cycle time for applications requiring heavy OPC UA communication.

See Also

"OPC UA Server" (→ p. 930)

15.15 Project Structure Guidelines

By implementing a predefined structure for new projects, KAS tries to achieve these goals:

- Efficiency in developing new applications
- High flexibility to keep only functionalities that are needed and to create the new ones that are required
- Safe applications due to an already tested and approved structure that optimize the resources usage (memory and processor load)
- Reliable framework that supports error, state, data and communication management
- Easier to exchange applications
- Less time needed to understand, maintain and teach an application (from a troubleshooting and support standpoint)
- Less documentation work is required since the main behavior of the Application is already documented (only the specific functionalities need some additional work)

15.15.1 External Files

Some items that belong to the application (shown in the "Project Explorer" (→ p. 85)) are not embedded into the project file.

For the domains listed in this table, KAS-IDE uses some resources stored in external files.

TIP

The links open the relevant topic with more details.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Domain	Description	File
Drive 	• The AKD drive is fully embedded in the Kollmorgen Automation Suite but not all interwoven at one time. • This makes future customization easier to get all the firmware features.	"FAQs" (➔ p. 781)
Fieldbus	Kollmorgen Automation Suite tightly integrates the EtherCAT motion bus (standard Ethernet-based cabling) to define all the network description.	"Import or Export an EtherCAT ENI File" (➔ p. 541)
HMI 	• Using Kollmorgen HMI, tag the variables in the PLC environment to create an export file that describes the data to be exchanged between the PLC and the HMI. • Import this Modbus mapping file into the HMI programming environment and use the variables as if they are local variables.	"Add an HMI Device" (➔ p. 215)
Motion	• The CAM editor is used to create complex CAM profiles online using a “graphical” interface. • It is also possible to import existing CAM profile points into the CAM editor to reuse the existing machine building experience seamlessly.	• "Import a CAM Profile" (➔ p. 706) • "Export the Oscilloscope Data" (➔ p. 756)
PLC	• The PLC programming environment is used to create reusable components (UDFB) and template applications. • These can be customized to suit any given application.	• "Create and Use Custom Libraries" (➔ p. 244) • "Use the Defines List" (➔ p. 523)

15.15.2 Application Software Structure - Definitions

15.15.2.1 Modules to Build up the Structure

- "Structure Overview" (➔ p. 1008)
- "Main Module" (➔ p. 1009)
- "Axis Module" (➔ p. 1010)

15.15.2.1.1 Structure Overview

You normally write the PLC program. In most cases , Kollmorgen application team members create the motion control part.

The global software structure is built with different modules placed on different levels:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

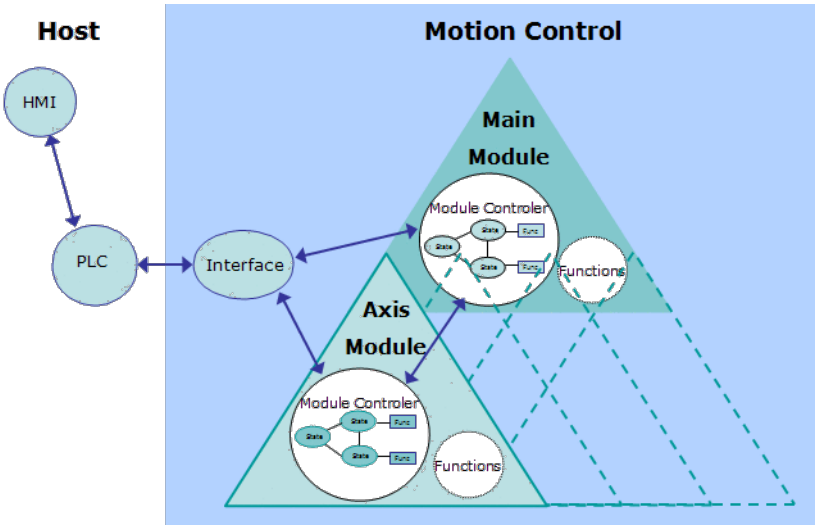


Figure 19-125: Software Structure Overview

Module Definition

- A module is one unit of the software structure (triangle)
- It is controlled by one module from the next higher level and can in turn control several modules in the next lower level
- It never communicates with modules of the same level
- It can generally run independently from any other modules at the same or higher level

To have the structure running as a real application, it needs to be controlled by a PLC. As the PLC is not part of the application structure, only the main and axis modules are described here.

15.15.2.1.2 Main Module

The main module controls the functional work that globally affect the application (e.g. multi axes functions). It receives commands from the PLC and sends back acknowledgements. The main module does not directly act on the physical axes, but controls the axis modules that are linked to them.

Communication between main and axis modules is done via internally defined data channels.

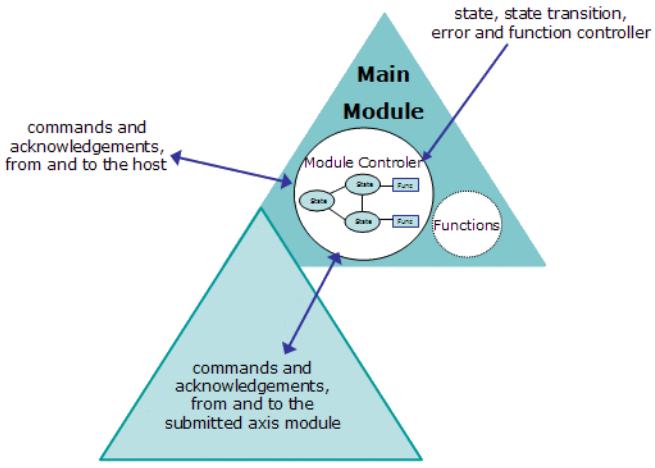


Figure 19-126: Main Module Description

As shown in the "Main Module Description" (→ p. 1009) image, the module consists of two parts:

15.15.2.1.3.1 Module Controller

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- This part is responsible for state, state transition, error and functions handling.
- It receives state transition and function call commands from the host, performs all needed actions and sends back some acknowledgments.
- In case of an error, it reacts by itself and sends a message to the PLC.
- If requested, it activates state transitions and functions in the axis modules, by sending commands to them and waiting for acknowledgment.
- The main module controller also manages the error status of the submitted modules and performs the needed actions.

15.15.2.1.4.2 Functional Part

- This consists of all functionalities needed for the current application.
- These functions can be state dependent (e.g., multi-axes functions) or state independent (e.g., increase a speed value).

15.15.2.1.5 Axis Module

The axis module controls the functional work that affect the application one or more physical axes (e.g., single-axis functions). It receives commands from the PLC and sends back acknowledgments.

The axis module also communicates with its main module via the internally defined data channel.

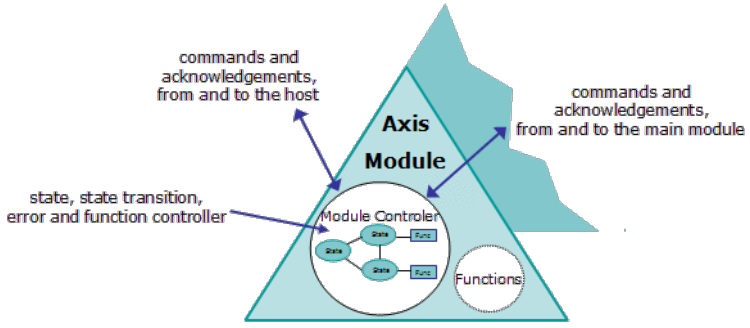


Figure 19-127: Axis Module Description

As shown in the "Axis Module Description" (→ p. 1010) image, the axis module consists of the same "Module Controller" (→ p. 1009) and "Functional Part" (→ p. 1010) two parts as the main module with this Module Controller addition:

15.15.2.1.6.1 Axis Module Controller

- If the axis module is **not** connected to its main module:
  - It receives state transition and function call commands from the host.
  - Performs all needed actions and sends back some acknowledgments.
- If the axis module is connected, state transition commands are received from its main module and not from the host.
  - In case of an error it only reacts by itself, if it is not connected to the main module.

15.15.2.2 State and Function Definitions

A state machine and some functions of general interest are implemented in the software structure.

- They are provided as examples of how to use the structure but can be adjusted to fulfill specific application usage.
  - See (→ # 1017, Add a New State) and (→ # 1013, Add a New Function) for more information.
- All modules have the same states and state transitions.
  - The state of a module is only influenced by other modules, if they are connected with each other.

The state machine has been defined.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

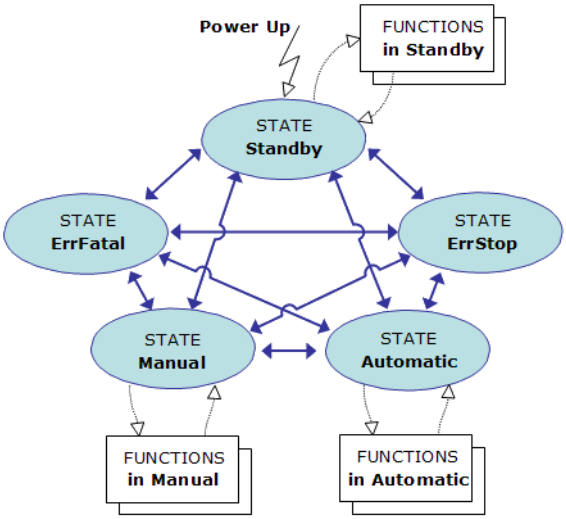
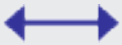
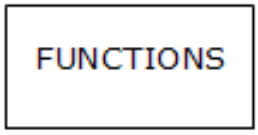


Figure 19-128: State Machine

Image Legend	Description
	Represents the current state of one or more modules (e.g., Standby state means the module is waiting for commands). In addition to the static states, there are two other dynamic states: • **Busy**: When performing a state transition (exit old and enter new state). • **Running**: When working through a function.
State Transition 	It is the transition from one state to another when performing the corresponding action.
	Any needed machine or single-axis functionality (e.g., single-axis jog), which can be called out of one or several states.

15.15.2.2.1 State, State Transitions and Functions

The structure is built so state transitions are possible from the active state to any other existing states (except state ErrStop).

- After leaving state ErrStop (corresponding to a non-fatal error, which causes a stop and power off) the structure automatically recovers the state which was active before entering ErrStop.
  - This means that all characteristics of the previous state are kept.

NOTE

Because functionalities are always specific to the application, none are included in the structure itself.

15.15.3 Application Software Structure - Implementation

This section describes how the software structure described before is implemented. Insofar as all modules are implemented and behave in the same way, only the main module is described in detail here.

15.15.3.1 SFC Children Building up the Software

These files contain all the data to build up the application.

They are all required to ensure a successful compilation.

15.15.3.1.1 Parent SFC

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

File	Description
Main	System start up and SFC children call

15.15.3.1.2 Main Module SFC Children

File	Description
M1_StateController	state and function controller of the main module
M1_ErrorHandling	error handling of the main module
M1_IndependentFunctions	state independent functions of the main module
M1_Interface	interface to PLC

15.15.3.1.3 Axis Module SFC Children

With i = 1... n

File	Description
Ai_StateController	state and function controller of the axis module
Ai_ErrorHandling	error handling of the axis module
Ai_IndependentFunctions	state independent functions of the axis module

15.15.3.2 Functions Linked to States

15.15.3.2.1 Functions Treatment

The "Function Treatment" (➔ p. 1012) image shows how functions (that are state dependent) are treated.

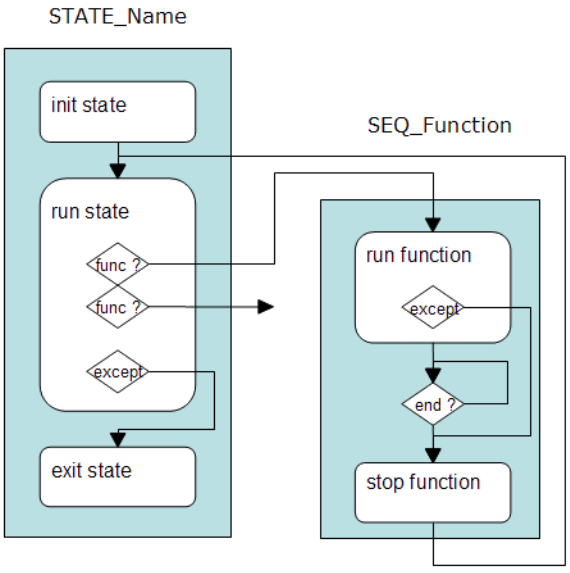


Figure 19-129: Function Treatment

15.15.3.2.2.1 Function (Function Step)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

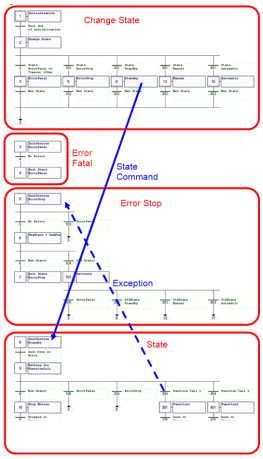
Function	Description	Example
Run Function	• Initializes exceptions on new state M1_CmdState <> M1_ReqState and on errors set in M1_StatusWord. • Goes to exit function when an exception occurs. • Acknowledges running. • Performs all actions needed for the function until the function call command is reset.	
Stop Function	• Performs all actions to properly leave this function. • Acknowledges end of exit, by setting M1_AckState to M1_ReqState. • Returns to last state.	

Figure 19-130: Change State > Error Stop > State

15.15.3.2.3 Add a New Function

1. Copy a similar existing function sequence.
2. Replace the old function name by the new one (e.g., Running replaced with MyFunction).
3. Modify the exit section of the new function to perform the relevant actions.
4. Insert the needed function code into the run part.
5. Add the function call command line to the state sequence where the function is used.

15.15.3.3 Main Module Implementation

In the main module, all necessary state, state transition, error and function handling facilities are implemented for this level.

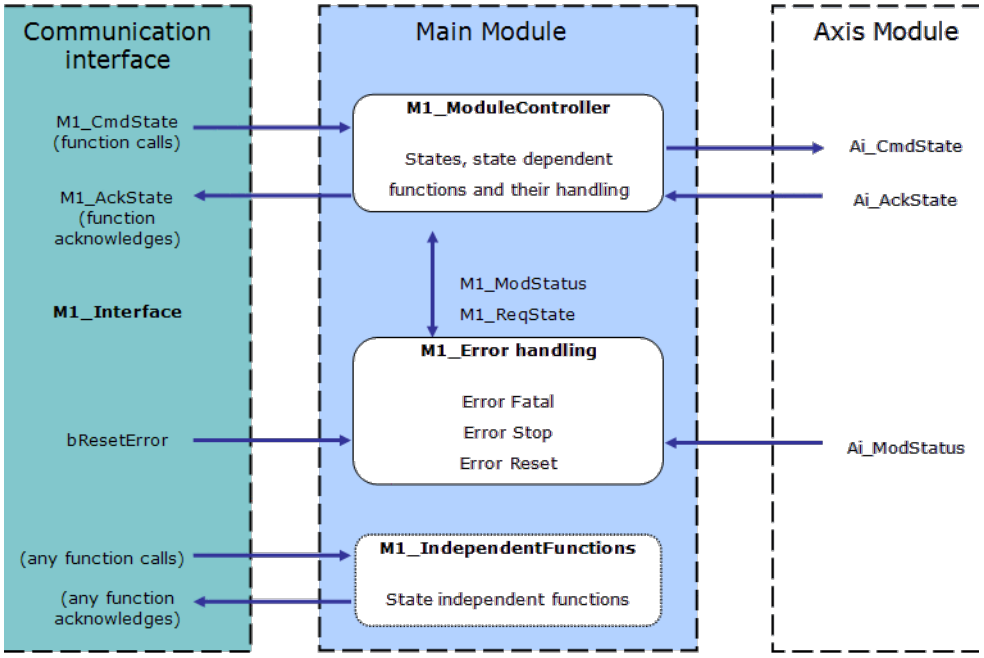


Figure 19-131: Context Diagram for Main Module

These objects (variables, tasks...) are defined in the structure of the main module.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "bErrorReset" (→ p. 1014)
- "M1_AckState" (→ p. 1014)
- "M1_CmdState" (→ p. 1014)
- "M1_ErrorHandling" (→ p. 1014)
- "M1_ModStatus, (Ai_ModStatus)" (→ p. 1015)
- "M1_ModuleController" (→ p. 1015)
- "M1_ReqState" (→ p. 1016)

15.15.3.3.1 bErrorReset

This internal flag variable is used as the error reset command for the main and axis modules. It is reset during power up.

Usage

- Set and reset this flag to activate a reset of the module errors (M1_StatusWord, Ai_StatusWord).

15.15.3.3.2 M1_AckState

This internal word variable contains the actual state acknowledge value, as a result from the **M1_CmdState** state command performed with success.

- Possible values are the same as for the state commands.
  - See "M1_CmdState" (→ p. 1014).

Usage

- The corresponding acknowledgments for the PLC can be created in the communication interface.

15.15.3.3.3 M1_CmdState

This internal word variable contains the actual state command value.

It is automatically set to state **Standby** during power up.

```
//*******************************************
//**             State Defines              **
//*******************************************
#define DEF_StateUndefined                   0
#define DEF_StateStandby                     1
#define DEF_StateManual                      2
#define DEF_StateAutomatic                   3
#define DEF_StateBusy                        4
#define DEF_StateErrorStop             5
#define DEF_StateErrorFatal                  6
```

Usage

- State commands are usually set in the communication interface (see software listing of ACT_M1_Translate and ACT_M1_SimaticSimu) and must not be set directly from the host system.
- If additional or different state commands are needed, the definitions can be modified.

15.15.3.3.4 M1_ErrorHandling

This program is responsible for the main module error handling.

- If an error occurs (in the main module or a submitted axis module), the corresponding bit in the module status (M1_StatusWord) is set.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- This causes the error reaction bits (MSK_Mi_StatusErrorStop, MSK_Mi_StatusErrorFatal) to be set in the module status word.

Usage

- Any additional error which needs to be treated has to be included in this program.
- Remember to modify the corresponding masks (**MSK_M1_StatusErrorFatal**, **MSK_M1_StatusErrorStop**) to cause the correct reaction on errors.

15.15.3.3.5 M1_ModStatus, (Ai_ModStatus)

This internal word variable contains the actual module status and error information.

- It is automatically set to the default value during power up.
- The meaning of the predefined **Module Error Bits** are:

Bit	Description
0	Error stop reported by drive (drive error).
1	Error fatal reported by Drive (lag error).
2	Not used (motor temperature too high).
3	Not used (external stop).
4	Not used (negative limit switch reached).
5	Not used (positive limit switch reach).
6	Not used (not used).
7	Not used (not used).
8	Not used (state HW enable)
9	Not used (state AS enable).
10	Not used (axis is powered on).
11	Not used (axis is homed).
12	Not used (axis is running).
13	Not used (pipe is connected).
14	Error stop (error stop).
15	Error fatal (error fatal).

Usage

- The error bits are usually set only by the error handling (M1_ErrorHandling).
  - The mode bits can be modified where ever needed in the application program (except in the interface).
  - Several bits can be set at the same time.
- Several masks have been defined to test or modify the whole word.
  - For each module, there is one mask to define the bits causing a fatal error (e.g. **MSK_M1_StatusErrorFatal**) and one for the stop error (e.g. **MSK_M1_StatusErrorStop**).
- To add errors and modes, the bits not already assigned by default can be used (i.e. bits 16 to 31).

15.15.3.3.6 M1_ModuleController

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This program is the heart of the whole controller and contains:

- A state manager sequence.
- All state sequences.
- State dependent function sequences of the main module.

Usage

- Some rules have to be followed when using and changing states and functions.
  - See (➔ # 1017, Add a New State) and (➔ # 1013, Add a New Function) for more information.

15.15.3.3.7.1 M1_ReqState

This internal word variable contains the internally active state.

- It is used for internal purpose only, to keep the actual state value (e.g., while performing a function).
  - Possible values are the same as for the state commands.
    - See "M1_CmdState" (➔ p. 1014).

Usage

- Used by system, **do not use** for application purposes.

15.15.3.4 States and Errors

15.15.3.4.1 States and Errors Treatment

The "State and Error Treatments" (➔ p. 1016) image shows how states and errors are treated.

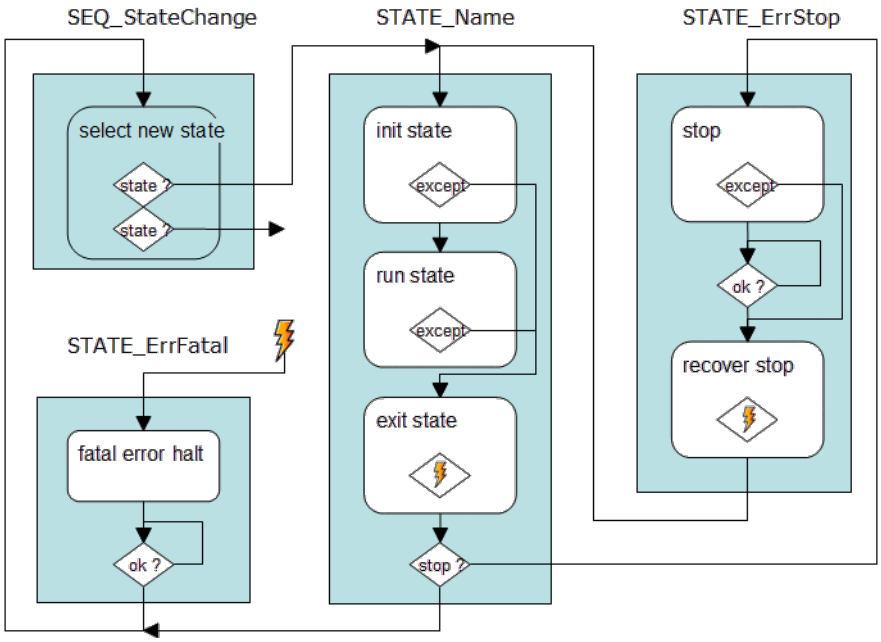


Figure 19-132: State and Error Treatments

StateChange (state manager): Activates the new state required by M1_ReqState.

StateName (state macro).

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

State	Description
Init State	• Initializes exceptions on new state M1_CmdState <>[1] M1_ReqState and on errors set in M1_StatusWord. • Goes to exit state when an exception occurs. • Performs all actions to properly enter this state (e.g., init variables, pipes, etc.) • Sends commands to the submitted axis modules by: • Setting Ai_CmdState to StateName. • Waiting for their acknowledgment in Ai_AckState. • Acknowledges end of initialization by setting M1_AckState to M1_ReqState.
Run State	• Waits for any function calls. • Activates function if called.
Exit State	• Performs all actions to properly leave this state. • Acknowledges running by setting M1_AckState to **busy**. • If error stop occurs, activates STATE_ErrStop. • Otherwise, sets new requested state M1_ReqState to M1_CmdState and activates StateChange.

15.15.3.4.2 Add a New State

1. Copy a similar existing state sequence.
2. Replace the old state name by the new one (e.g., Standby replaced with MyState).
3. Modify both init and exit sections of the new state to perform the relevant actions.
4. Insert the needed function calls into the states run part.
5. Add the state call command line into the state change sequence.
6. Add the state definition values to the general declaration.

15.15.3.5 Variables for the Interface

15.15.3.5.1 List of Variables

- bAi_CallAutomaticFunction1
- bAi_CallAutomaticFunction2
- bAi_CallManualFunction1
- bAi_CallManualFunction2
- bAi_CallStandbyFunction1
- bAi_CallStandbyFunction2
- bErrorReset
- bM1_CallAutomaticFunction1
- bM1_CallAutomaticFunction2
- bM1_CallManualFunction1
- bM1_CallManualFunction2
- bM1_CallStandbyFunction1
- bM1_CallStandbyFunction2
- M1_CmdState

15.15.3.5.2 List of Output Variables

- Ai_StatusWord
- bAi_Running
- bM1_Running

[1] <> means Not Equal

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- M1_AckState
- M1_StatusWord

15.16 Project Templates

KAS provides start up templates to help you get started.

See "Add the Controller" (→ p. 153) to use the project setup wizard.

These are the templates for the Pipe Network and PLCopen motion engines:

- "Coordinated Motion" (→ p. 1018)
- "KAS Runtime" (→ p. 1018)
- "KDN Download" (→ p. 1018)
- "Pipe Network" (→ p. 1018)
- "PLCopen" (→ p. 1019)

15.16.1 Coordinated Motion

Template Name	Description
"Coordinated Motion - 2-Axis Template" (→ p. 1020)	• Linear / Circular • Raster Scan Motion Path • 2 PLCopen axes
"Coordinated Motion - 3-Axis Template" (→ p. 1021)	• Linear / Circular • Raster Scan Motion Path • 1 Pipe Network axis • 2 PLCopen axes
"Coordinated Motion - 3-Axis (3D) Template" (→ p. 1022)	• Linear 3D • Diamond / Square Motion Path • 3 PLCopen axes

15.16.2 KAS Runtime

Template Name	Description
Library	Used to create a custom library. See also see "Create and Use Custom Libraries" on page 244.

15.16.3 KDN Download

Template Name	Description
KDN	1. Click **Finish** to close the dialog. A web browser page opens on KDN with custom-built template files. The template files may have enhanced or specific functionality. 2. Download and save a template file. 3. Open it from KAS-IDE.

15.16.4 Pipe Network

Template Name	Description
"Pipe Network 1-Axis Template - FFLD" (→ p. 1025)	Simple initialization, relative position move.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Template Name	Description
"Pipe Network 1-Axis Template - ST" (→ p. 1025)	Simple initialization, relative position move.
"Pipe Network 2-Axes Template - FFLD" (→ p. 1034)	Position and velocity moves, gearing, camming, and drive status.
"Pipe Network 2-Axes Template - ST" (→ p. 1031)	Position and velocity moves, gearing, camming, and drive status.
"Pipe Network 2-Axes Template - FBD, FFLD, SFC, and ST" (→ p. 1026)	Position and velocity moves, gearing, camming, and drive status with optimized performance.

15.16.5 PLCopen

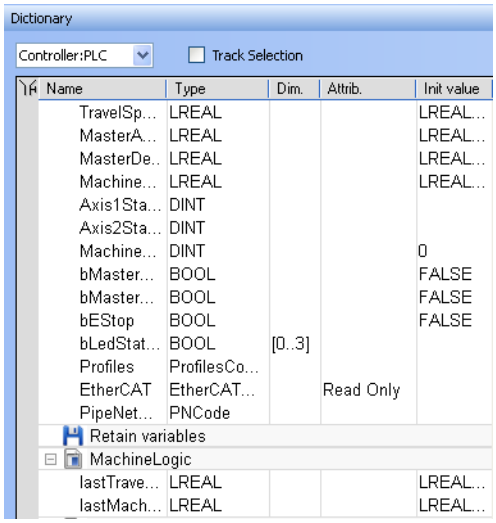
Template Name	Description
"PLCopen1-Axis Template - FFLD" (→ p. 1037)	Simple initialization, relative position move.
"PLCopen 1-Axis Template - ST" (→ p. 1038)	Simple initialization, relative position move.
"PLCopen 2-Axes Template - FFLD" (→ p. 1047)	Position and velocity moves, gearing, camming, and drive status.
"PLCopen 2-Axes Template - ST" (→ p. 1043)	Position and velocity moves, gearing, camming, and drive status.
"PLCopen 2-Axes Template - FFLD and SFC" (→ p. 1039)	Position and velocity moves, gearing, camming, and drive status with optimized performance.

15.16.6 Template Inclusions

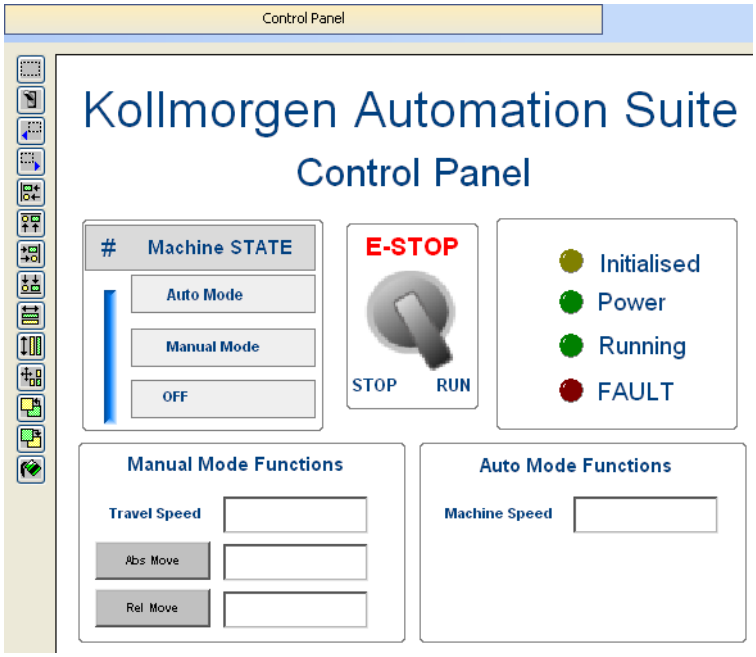
These templates come complete with software to:

- Create two or three axes of servo motion
- Enable the drives
- Perform simple motion

The templates contain variables which support these operations.



Additionally, the templates contain a Control Panel for ease of running motion.



NOTE

The templates may need to be changed to run on specific hardware.

15.16.7 Coordinated Motion

- "Coordinated Motion - 2-Axis Template" (→ p. 1020)
- "Coordinated Motion - 3-Axis Template" (→ p. 1021)
- "Coordinated Motion - 3-Axis (3D) Template" (→ p. 1022)

15.16.7.1 Coordinated Motion - 2-Axis Template

This project controls two axes in coordinated motion (PLCOpenAxis1 and PLCOpenAxis2).

15.16.7.1.1 Programs

The program is a Sequential Function Chart (SFC) containing both Structured Text (ST) and Free Form Ladder Diagram (FFLD) code.

- Steps 1 to 5 of the SFC create and initialize the axes and the coordinated motion axes group.
- Step 6 of the SFC interfaces with the Control Panel and performs a back-and-forth coordinated motion pattern with the two axes.
- The program utilizes coordinated motion direct moves, linear moves, circular moves, transitions and blending.

15.16.7.1.2 Motion Procedure

1. Download and start the application.
2. Press the **Enable** button to enable the axes and the axes group.
3. Optional: Press the **Home** button to move the axes to their zero position.
4. When the **Ready** light is on, turn the **Cycle Start** switch to 1.
   The axes begin moving in the programmed pattern.

15.16.7.1.3 Control Panel

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

#	Description	GUI
1	Used to change the coordinated motion feedrate from 0% to 200%.	
2	Displays the axis positions.	
3	Indicator light. • **Ready** indicates: • The axes are enabled. • The axes group is ready and enabled for coordinated motion.	
4	Begins the coordinated motion pattern.	
5	• Causes an Emergency Stop for all axes. • The axes are disabled.	
6	• Resets the Emergency Stop. • Clears axis errors. • Enables the axes.	
7	Disables the axes or axes group.	
8	Enables the axes and the axes group.	
9	Moves all axes or axes group to their zero position.	
10	Causes a reference by setting the axes' position to zero at their current location.	

15.16.7.2 Coordinated Motion - 3-Axis Template

This project controls two axes in coordinated motion (PLCOpenAxis1 and PLCOpenAxis2), and a third independent axis (VERTICAL_AXIS).

This template demonstrates how to use coordinated motion PLCopen axes and a Pipe Network axis.

15.16.7.2.1 PLC Programs

The Coordinated Motion 3-Axis template has a Sequential Function Chart (SFC) program containing both Structured Text (ST) and Free Form Ladder Diagram (FFLD) code.

- Steps 1 to 5 of the SFC program create and initialize the axes and the coordinated motion axes group plus the Pipe Network axis.
- Step 6 of the SFC interfaces with the Control Panel and performs a back-and-forth coordinated motion pattern with the two axes.
- The program utilizes coordinated motion direct moves, linear moves, circular moves, transitions and blending.
  - It performs basic moves for the third (Pipe Network) axis, to move down/up before and after the coordinated motion pattern.

15.16.7.2.2 Motion Procedure

This template uses both motion engines (Pipe Network and PLCopen) simultaneously.

IMPORTANT

Coordinated motion can only be performed with PLCopen axes, Pipe Network axes do not support this feature. PLCopen axes that perform coordinated motion can be mixed with independent Pipe Network axes.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

There is no axis synchronization at the Motion Engine level between a PLCopen axis and a Pipe Network axis. Any synchronization between the axes must be performed inside the PLC application.

1. Download and start the application.
2. Press the **Enable** button to enable the axes and the axes group.
3. Optional: Press the **Home** button to move the axes to their zero position.
4. When the **Ready** light is on, turn the **Cycle Start** switch to 1.
   The axes begin moving in the programmed pattern.

15.16.7.2.3 Control Panel

#	Description	GUI
1	Used to change the coordinated motion feedrate from 0% to 200%.	
2	Displays the axis positions.	
3	Indicator lights. • **Ready** indicates: • The axes are enabled. • The axes group is ready and enabled for coordinated motion. • Error Stopped indicates the axes group has been stopped and disabled by the E-Stop button.	
4	Begins the coordinated motion pattern.	
5	• Causes an Emergency Stop for all axes. • The axes are disabled.	
6	• Resets the Emergency Stop. • Clears axis errors.	
7	Disables the axes or axes group.	
8	Enables the axes and the axes group.	
9	Moves all axes or axes group to their zero position.	
10	Causes a reference by setting the axes' position to zero at their current location.	

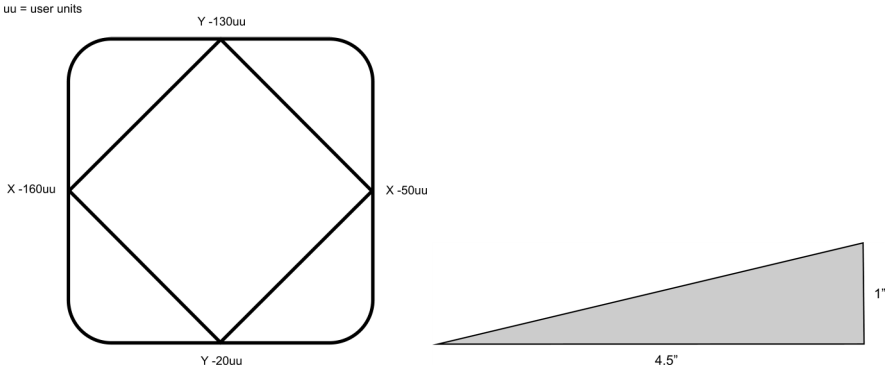
15.16.7.3 Coordinated Motion - 3-Axis (3D) Template

This project template controls three axes in coordinated motion (PLCOpenAxis1, PLCOpenAxis2, and PLCOpenAxis3).

- It demonstrates how to use 3D coordinated motion, transitions, blending and a homing cycle with PLCopen axes.
- The path follows a square and diamond pattern on a plane which is rotated ~12.5 degrees about the Y axis with the center of rotation located at Z=30 X=-160.

This is the pattern and the platform for the plane:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



NOTE

This program is designed to run on the 3-Axis Demonstration machine shown in the video.
It uses the machine's physical end-limit switches for the homing sequence.
To run this program on different hardware, press the **Sim Mode** button on the Control Panel.
This causes the program to execute in simulator mode and bypasses the homing sequence.
The axis scaling parameters (entered in the PLCopen Axis Data dialogs) may need to be changed to accommodate the different hardware.

15.16.7.3.1 PLC Programs

The Coordinated Motion 3-Axis template has a Sequential Function Chart (SFC) program containing both Structured Text (ST) and Free Form Ladder Diagram (FFLD) code.

- Steps 1 to 5 of the SFC program create and initialize the axes and the coordinated motion axes group.
- Step 6 specifies the coordinates of the square and diamond pattern.
  - These coordinates are then rotated about the Y-axis.
- Step 7 monitors the Control Panel and performs two main functions.
  - The first function is to reference the axes to establish a home position.
  - The second function is to perform the 3-axis coordinated motion moves of the square diamond pattern.

This program provides examples of coordinated motion linear moves, transitions, blending, and a homing cycle.

15.16.7.3.2 Motion Pattern Procedure

1. Download and start the application.
2. Press the **Enable** button to enable the axes and the axes group.
3. After the **Ready** light is on:
   - If working with the 3-Axis Demonstration machine, press the **Home** button to reference the axes and move them to their home position.
   - If working with other hardware, press the **Sim Mode** button.
4. After the **Ready** and **Homed** lights are on, turn the **Cycle Start** switch to 1.
   The axes begin moving in the programmed pattern.

15.16.7.3.3 Control Panel

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Image #	Description	GUI
1	Used to change the coordinated motion feedrate from 0% to 200%.	
2	Displays the axis positions.	
3	Ready indicator light. **Ready** indicates: • **Ready** indicates: • The axes are enabled. • The axes group is ready and enabled for coordinated motion.	
4	Homed indicator light. **Homed** indicates: • Each axis has been referenced. • The axis group has moved to its zero position.	
5	Simulation Mode indicator light. **Sim Mode** indicates: • The homing functions have been bypassed. • The program is set up as if the homing function has been performed.	
6	Begins the coordinated motion pattern. **Ready** and **Homed** lights must both be on to execute motion.	
7	Causes an Emergency Stop for all axes. • The axes are disabled. • **Ready** and **Homed** lights are turned off.	
8	• Resets the Emergency Stop. • Clears axis errors.	
9	Disables the axes or axes group. The axes must be at a standstill to disable the group.	
10	Enables the axes and the axes group. This turns on the **Ready** light.	
11	Starts a homing function. • Each axis is referenced. • Moves all axes or axes group to their zero position. • Homing must be completed before executing a cycle start. • When homing is complete, the **Homed** light is turned on.	
12	**Sim Mode** bypasses the homing function. This mode can be used when running on a simulator or when the hardware reference switches are not available. • The current position of each axis is set to zero position. • The homing function is not performed. • The **Sim Mode** light and **Homed** light are turned on.	

15.16.8 Pipe Network Axis Templates

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Pipe Network 1-Axis Template - FFLD" (➔ p. 1025)
- "Pipe Network 1-Axis Template - ST" (➔ p. 1025)
- "Pipe Network 2-Axes Template - FFLD" (➔ p. 1034)
- "Pipe Network 2-Axes Template - ST" (➔ p. 1031)
- "Pipe Network 2-Axes Template - FBD, FFLD, SFC, and ST" (➔ p. 1026)

15.16.8.1 Pipe Network 1-Axis Template - FFLD

The 1-axes Pipe Network template has an FFLD program (called **Main**) that initializes, starts and runs the motion.

There are controls to enable / disable the drive, clear faults, and display fault and axis position information in the Control Panel.

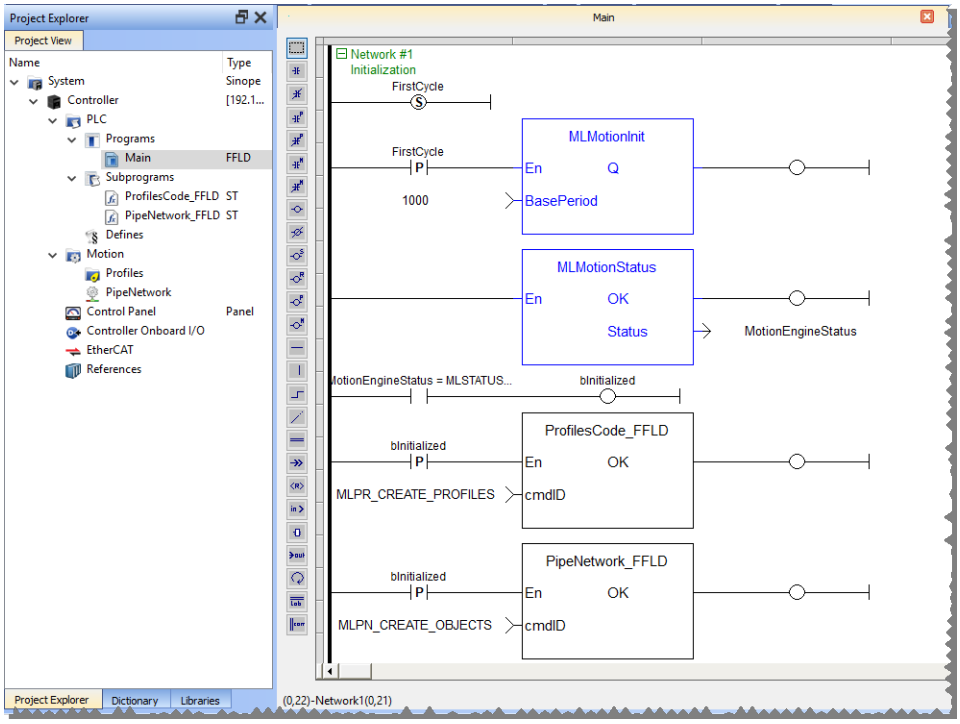
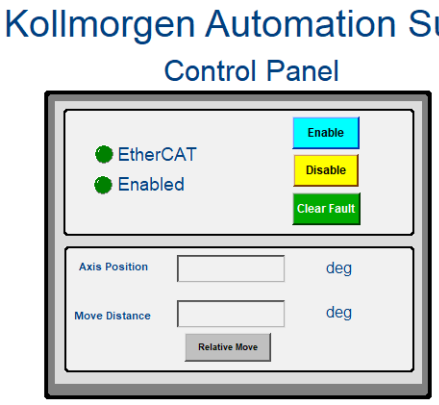


Figure 19-133: PN Template with FFLD - Main



15.16.8.2 Pipe Network 1-Axis Template - ST

The 1-axis Pipe Network template has a ST program (called **Main**) that initializes, starts and runs the motion.

Controls are included to enable/disable the drive, clear faults, and display fault and axis position information in the Control Panel.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

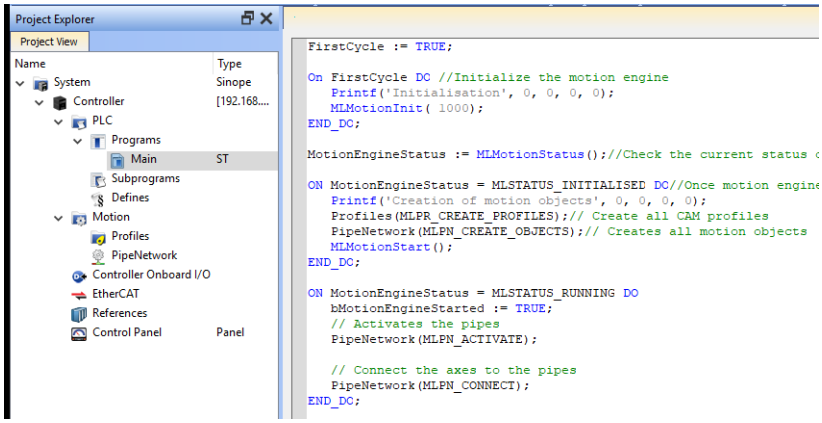
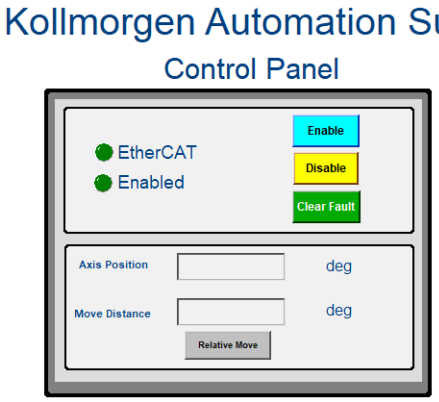


Figure 19-134: PN Template with ST - Main



15.16.8.3 Pipe Network 2-Axes Template - FBD, FFLD, SFC, and ST

Based on the template, the project can be run using either the:

- "KAS Simulator Graphical User Interface" (➔ p. 586).
- Actual drives and motors.
  - In this case, set up the axes first in the EtherCAT part).
    See "Configure an EtherCAT" (➔ p. 535) for more information.

- "Control Panel" (➔ p. 1026)
- "HMI" (➔ p. 1027)
- "Motion" (➔ p. 1028)
- "PLC Programs" (➔ p. 1028)

15.16.8.3.1 Control Panel

See "Design the HMI with the Internal Editor" (➔ p. 715) for more information.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

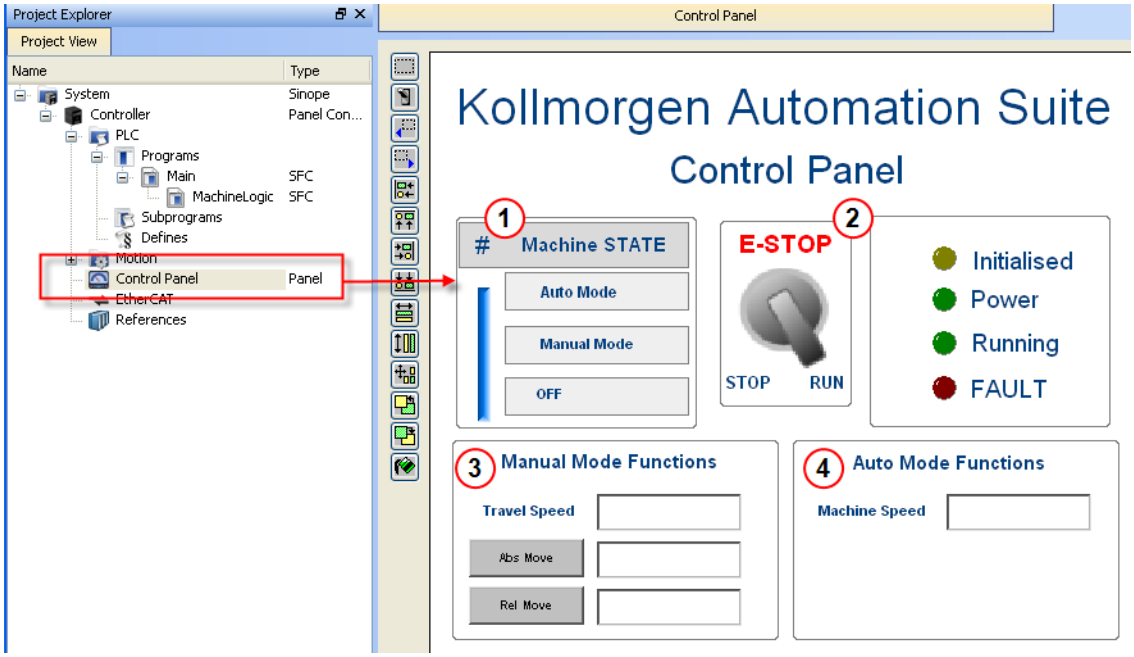


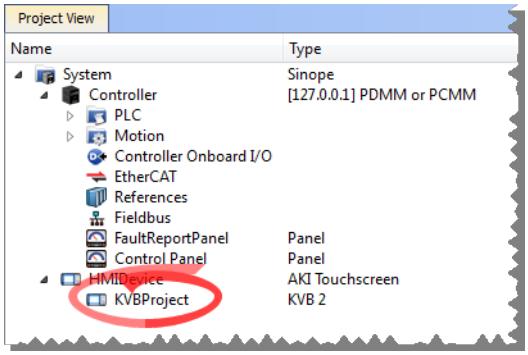
Figure 19-135: PN Template - Control Panel

Call Out#	Description
1	Used to choose how to run the axes between manual (3) and automatic (4) modes.
2	When you click the emergency button: • The machine state becomes OFF (see call out 1). • The two axes stop running.
3	In manual mode, you can set the speed. You can set an absolute and relative move. When you click those commands, the two axes move to the specified position and the speed is reset to 0 (zero).
4	In automatic mode, you can set the speed.

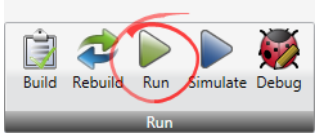
15.16.8.3.2 HMI

In the project template is an HMI screen for the AKI terminals created in the Kollmorgen Visualization Builder (KVB) environment.

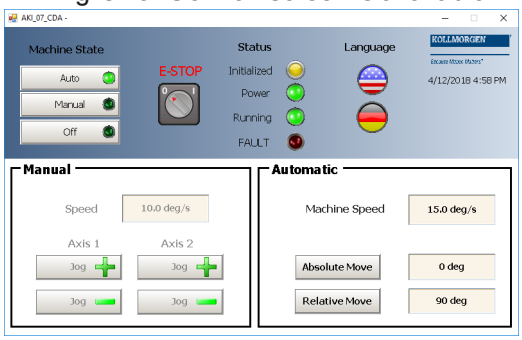
The KVB software is a separate installation from the KAS-IDE. KVB can be opened from within KAS-IDE by double-clicking on the **KVB Project** item in the project tree.



The project can be run in the KVB programming environment after the KAS-IDE program has started running. Click the Run button in the Project ribbon to start the program.



A screen duplicates the controls on the IDE Internal Control panel screen.
An English or German screen is available.



15.16.8.3.3 Motion

The template has a motion profile defined with the graphical Pipe Network editor.

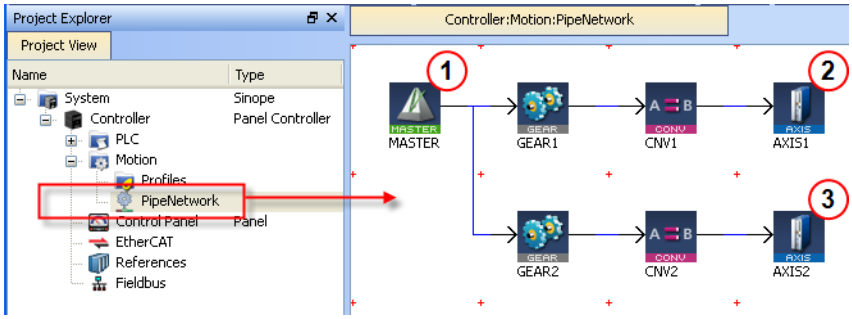


Figure 19-136: PN Template - Motion

The motion profile contains these different pipe blocks:

- The **Master** (see call out 1) is the generator that allows a synchronization between the two pipes (2 and 3).
- The **Gear** modifies (with ratio and offset) the flow of values issued from the Master.
- The **Convertor** controls the position of the axis.
- The **Axis** gives access to the physical remote drive.

15.16.8.3.4 PLC Programs

The 2-axes Pipe Network template has an SFC program (called **Main**) that initializes and starts the motion.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

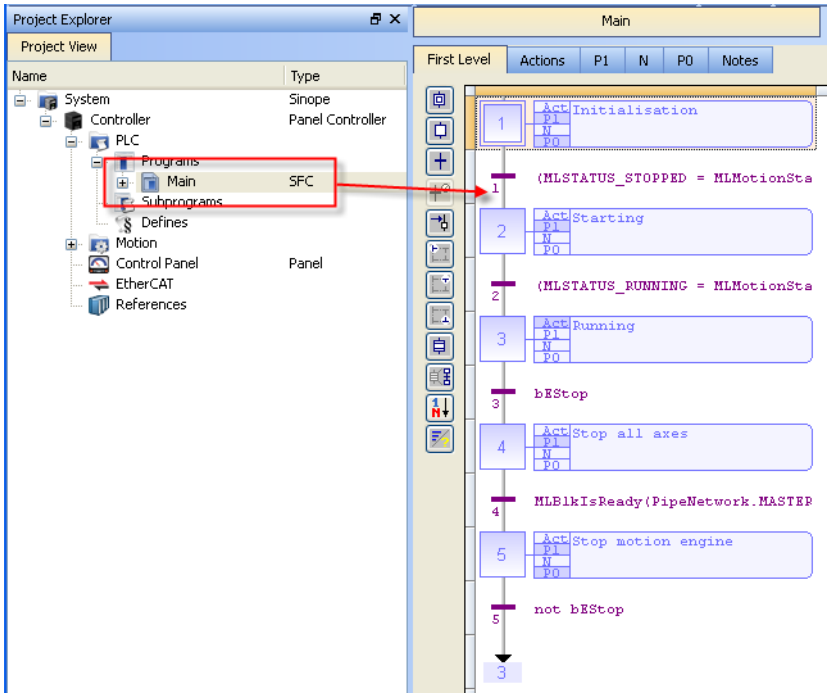


Figure 19-137: PN Template - Main

The Pipe Network Template contains an SFC child program called Machine Logic for running motion.

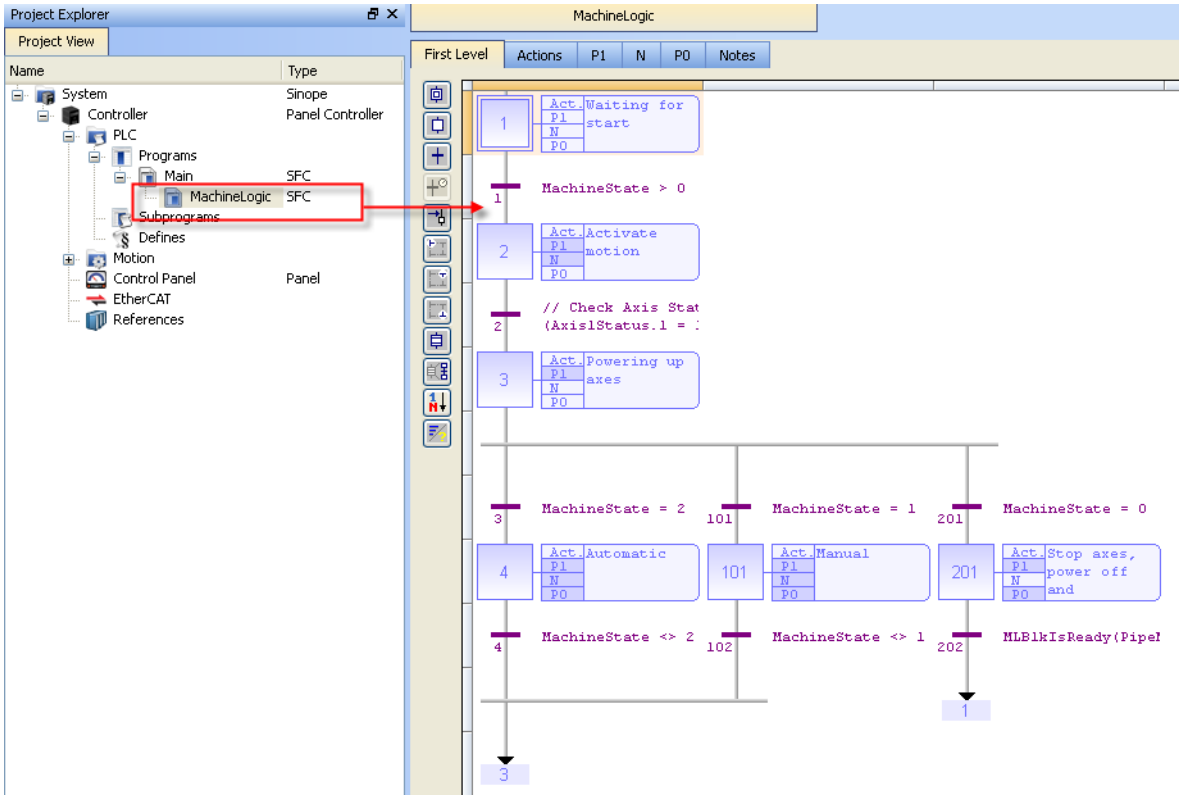
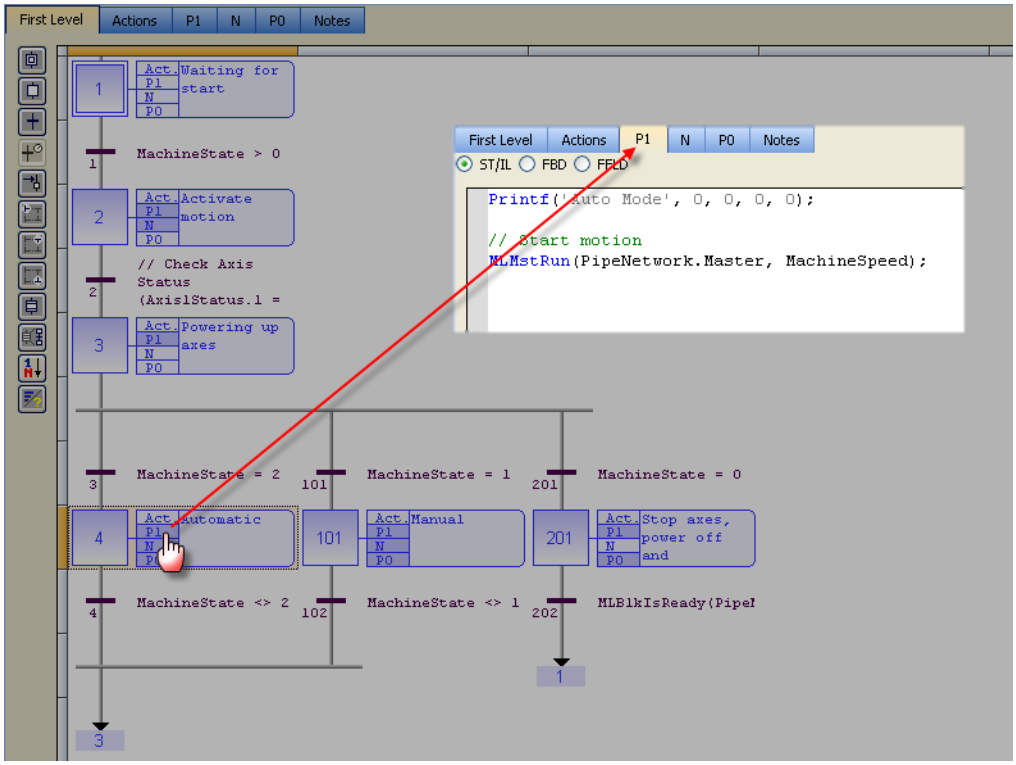


Figure 19-138: PN Template - MachineLogic

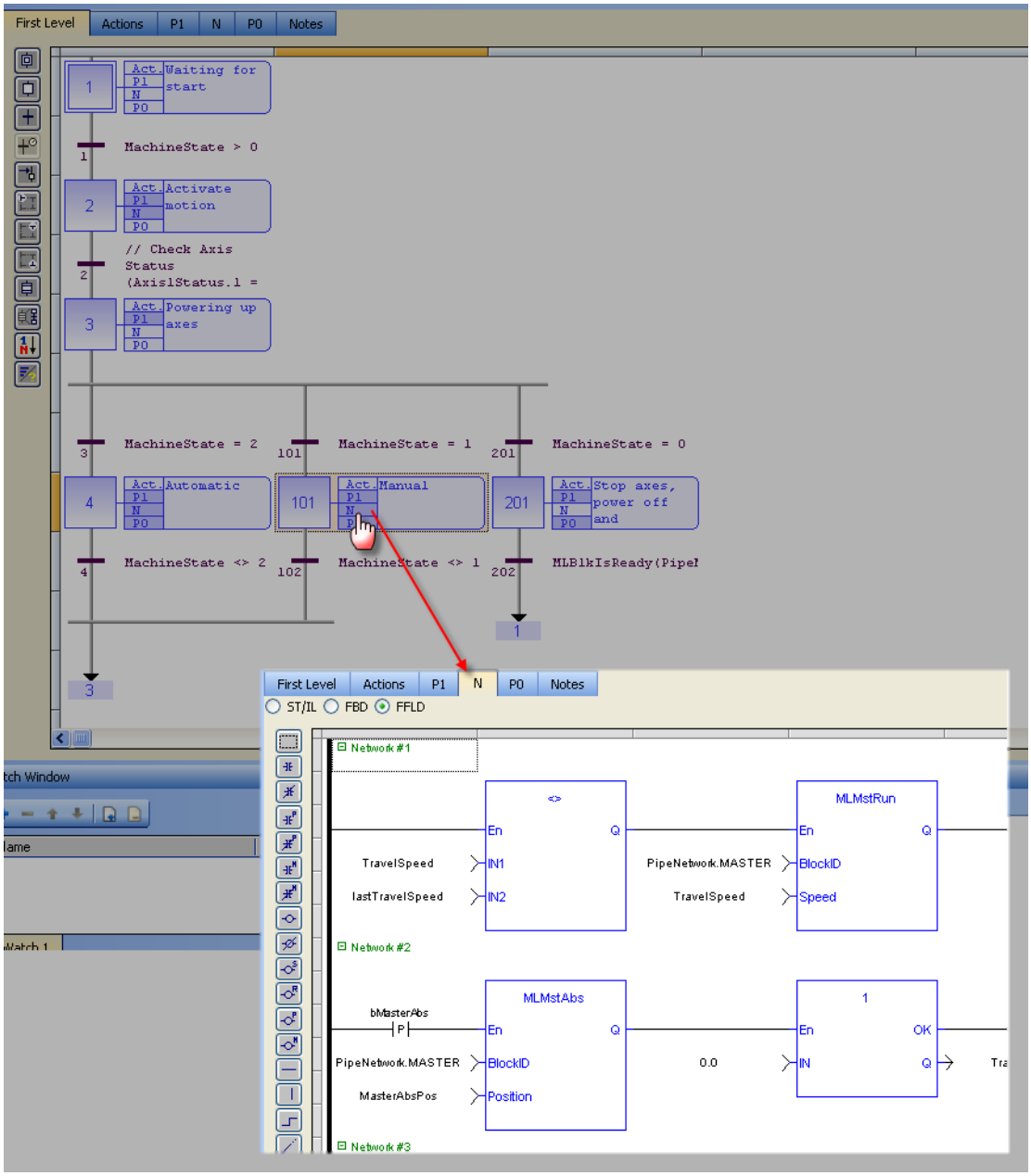
ST programs are found in the P1 and P0 actions for many steps

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



FFLD programs are found in the N action for steps 4 and 101

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



15.16.8.4 Pipe Network 2-Axes Template - ST

Based on the template, the project can be run using either the:

- "KAS Simulator Graphical User Interface" (→ p. 586).
- Actual drives and motors.
  - In this case, set up the axes first in the EtherCAT part).
    See "Configure an EtherCAT" (→ p. 535) for more information.

15.16.8.4.1 Control Panel

See "Design the HMI with the Internal Editor" (→ p. 715) for more information.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

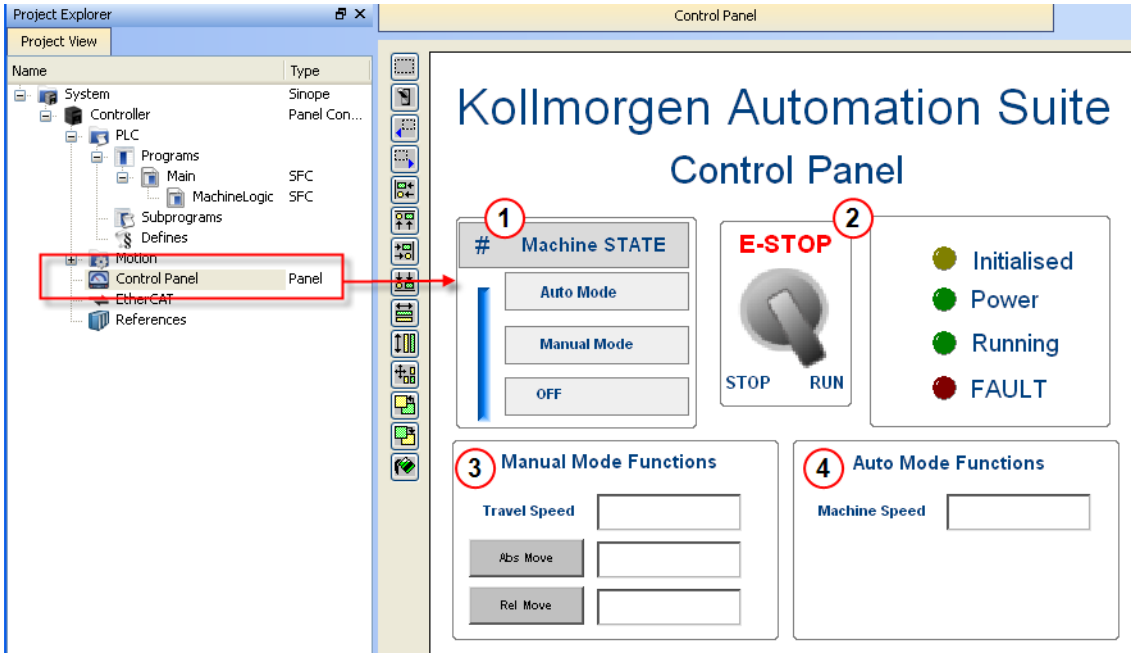


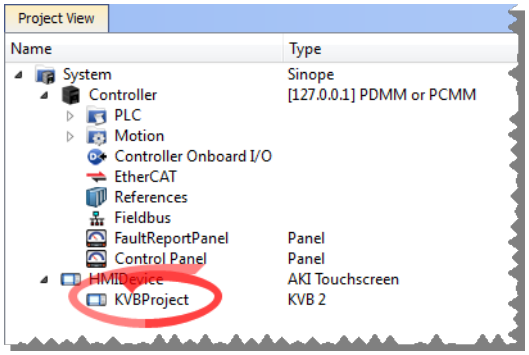
Figure 19-139: PN Template - Control Panel

Call Out#	Description
1	Used to choose how to run the axes between manual (3) and automatic (4) modes.
2	When you click the emergency button: • The machine state becomes OFF (see call out 1). • The two axes stop running.
3	In manual mode, you can set the speed. You can set an absolute and relative move. When you click those commands, the two axes move to the specified position and the speed is reset to 0 (zero).
4	In automatic mode, you can set the speed.

15.16.8.4.2 HMI

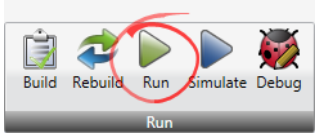
In the project template is an HMI screen for the AKI terminals created in the Kollmorgen Visualization Builder (KVB) environment.

The KVB software is a separate installation from the KAS-IDE. KVB can be opened from within KAS-IDE by double-clicking on the **KVB Project** item in the project tree.

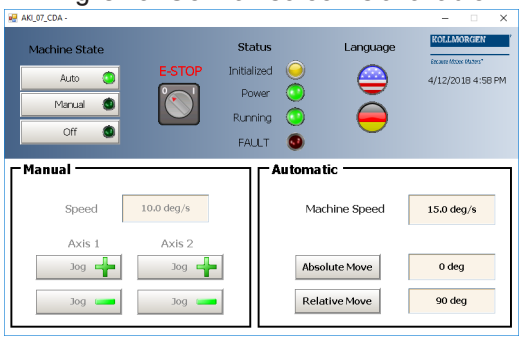


See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

The project can be run in the KVB programming environment after the KAS-IDE program has started running. Click the Run button in the Project ribbon to start the program.



A screen duplicates the controls on the IDE Internal Control panel screen.
An English or German screen is available.



15.16.8.4.3 Motion

The template has a motion profile defined with the graphical Pipe Network editor.

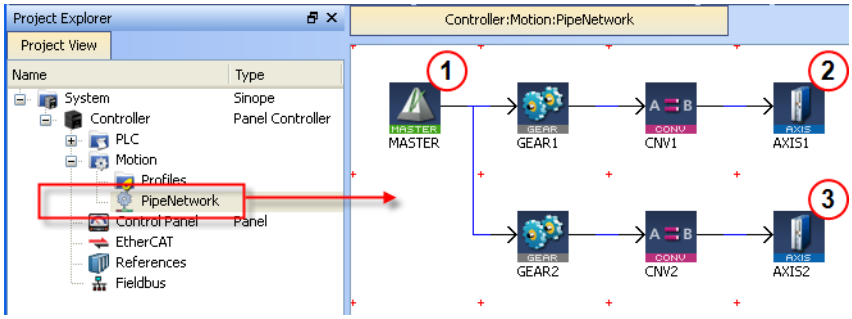


Figure 19-140: PN Template - Motion

The motion profile contains these different pipe blocks:

- The **Master** (see call out 1) is the generator that allows a synchronization between the two pipes (2 and 3).
- The **Gear** modifies (with ratio and offset) the flow of values issued from the Master.
- The **Convertor** controls the position of the axis.
- The **Axis** gives access to the physical remote drive.

15.16.8.4.4 PLC Programs

The 2-axes Pipe Network template has a ST program (called **Main**) that initializes, starts and runs the motion.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

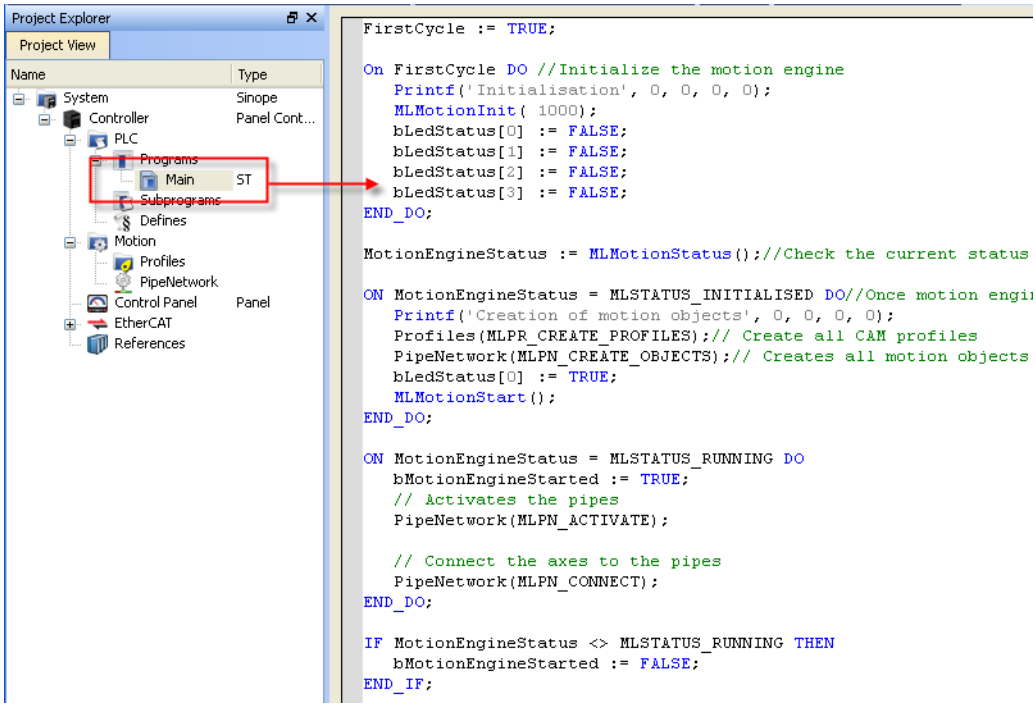


Figure 19-141: PN Template with ST - Main

15.16.8.5 Pipe Network 2-Axes Template - FFLD

Based on the template, the project can be run using either the:

- "KAS Simulator Graphical User Interface" (➔ p. 586).
- Actual drives and motors.
  - In this case, set up the axes first in the EtherCAT part).
    See "Configure an EtherCAT" (➔ p. 535) for more information.

- "Control Panel" (➔ p. 1034)
- "HMI" (➔ p. 1035)
- "Motion" (➔ p. 1036)
- "PLC Programs" (➔ p. 1036)

15.16.8.5.1 Control Panel

See "Design the HMI with the Internal Editor" (➔ p. 715) for more information.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

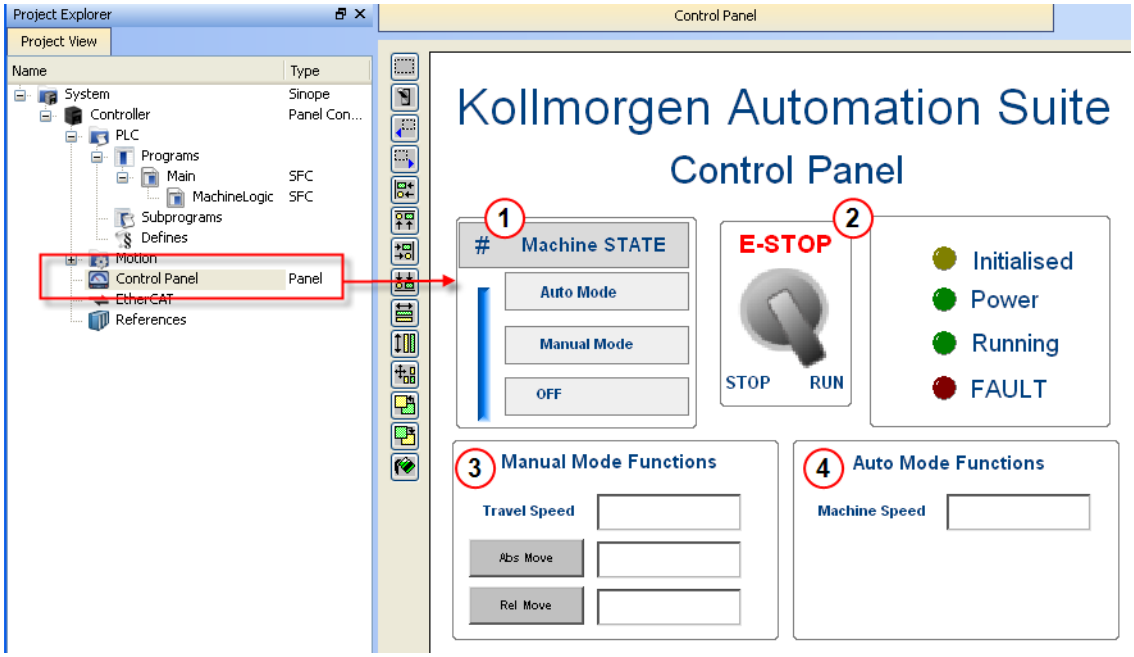


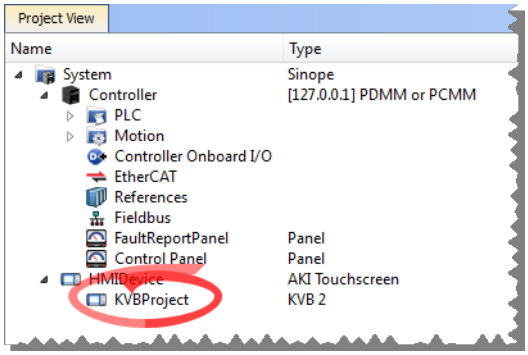
Figure 19-142: PN Template - Control Panel

Call Out#	Description
1	Used to choose how to run the axes between manual (3) and automatic (4) modes.
2	When you click the emergency button: • The machine state becomes OFF (see call out 1). • The two axes stop running.
3	In manual mode, you can set the speed. You can set an absolute and relative move. When you click those commands, the two axes move to the specified position and the speed is reset to 0 (zero).
4	In automatic mode, you can set the speed.

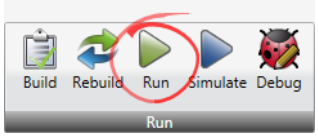
15.16.8.5.2 HMI

In the project template is an HMI screen for the AKI terminals created in the Kollmorgen Visualization Builder (KVB) environment.

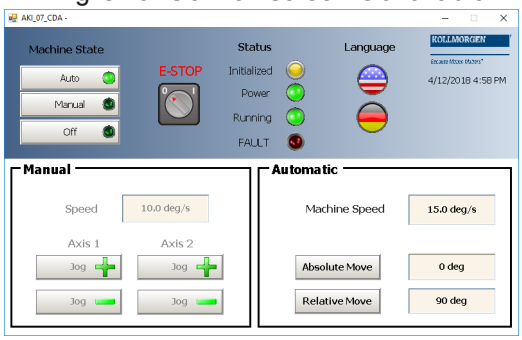
The KVB software is a separate installation from the KAS-IDE. KVB can be opened from within KAS-IDE by double-clicking on the **KVB Project** item in the project tree.



The project can be run in the KVB programming environment after the KAS-IDE program has started running. Click the Run button in the Project ribbon to start the program.



A screen duplicates the controls on the IDE Internal Control panel screen.
An English or German screen is available.



15.16.8.5.3 Motion

The template has a motion profile defined with the graphical Pipe Network editor.

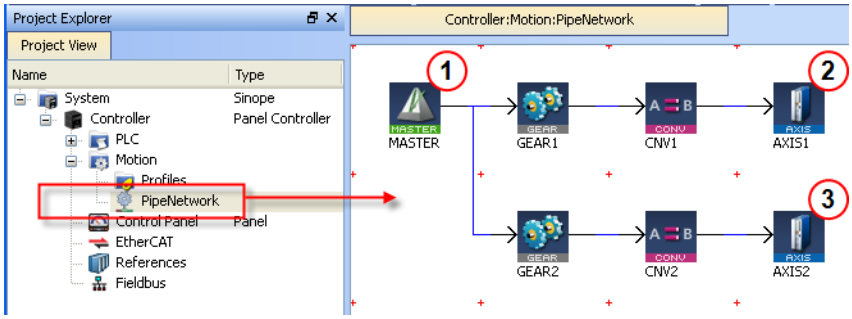


Figure 19-143: PN Template - Motion

The motion profile contains these different pipe blocks:

- The **Master** (see call out 1) is the generator that allows a synchronization between the two pipes (2 and 3).
- The **Gear** modifies (with ratio and offset) the flow of values issued from the Master.
- The **Convertor** controls the position of the axis.
- The **Axis** gives access to the physical remote drive.

15.16.8.5.4 PLC Programs

The 2-axes Pipe Network template has a FFLD program (called **Main**) that initializes, starts and runs the motion.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

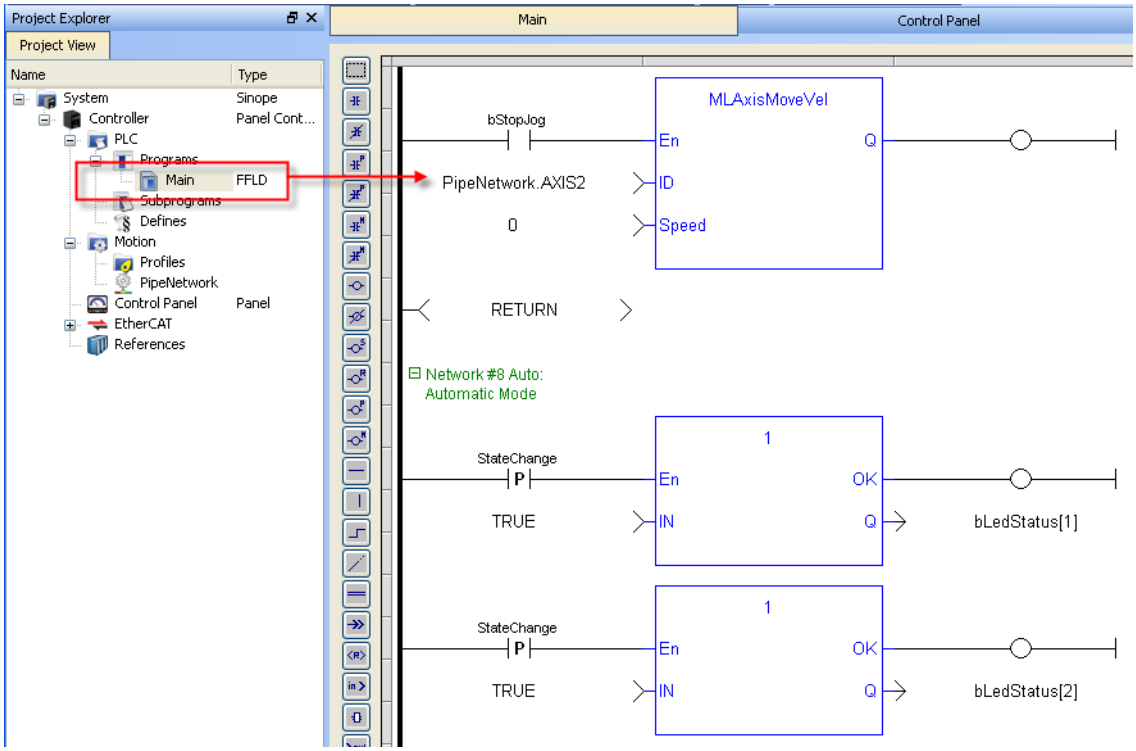


Figure 19-144: PN Template with FFLD - Main

15.16.9 PLCopen Axis Templates

- "PLCopen1-Axis Template - FFLD" (➔ p. 1037)
- "PLCopen 1-Axis Template - ST" (➔ p. 1038)
- "PLCopen 2-Axes Template - FFLD" (➔ p. 1047)
- "PLCopen 2-Axes Template - ST" (➔ p. 1043)
- "PLCopen 2-Axes Template - FFLD and SFC" (➔ p. 1039)

15.16.9.1 PLCopen1-Axis Template - FFLD

The 1-axis PLCopen template has an FFLD program (called **Main**) that initializes and starts the motion.

- Controls are included to enable/disable the drive, clear faults, and display fault and axis position information in the Control Panel.
- A second program, FaultMonitor, reads the drive fault and position information.
- The programs are commented to ease their use.

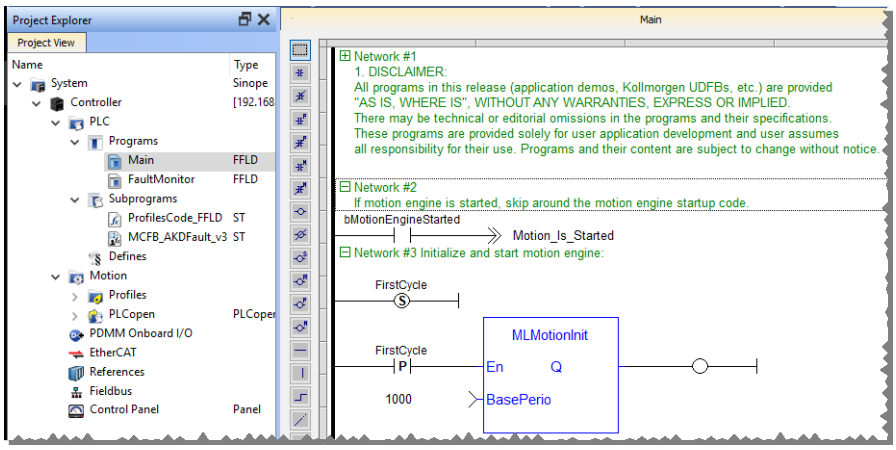
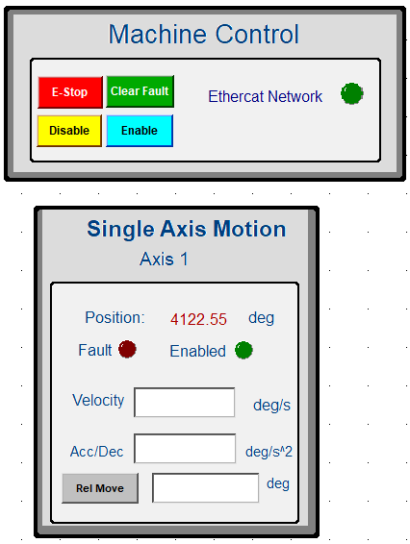


Figure 19-145: PLCopen Template with FFLD - Main



15.16.9.2 PLCopen 1-Axis Template - ST

This 1-axis PLCopen template has a ST program (called **Main**) that initializes, starts and runs the motion.

- Controls are included to enable/disable the drive, clear faults, and display fault and axis position information in the Control Panel.
- A subprogram reads the drive fault and position information.
- The programs are commented to ease their use.

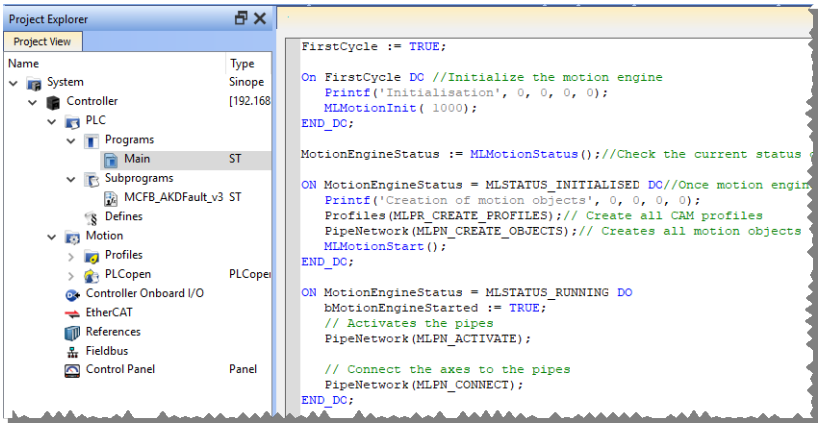
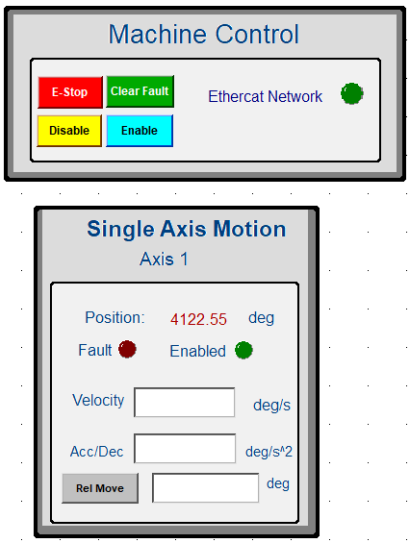


Figure 19-146: PLCopen Template with ST - Main

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



15.16.9.3 PLCopen 2-Axes Template - FFLD and SFC

This template project contains a two-axis template for getting started with a KAS application.

- It contains key project elements that save development time.
- The templates are designed to add code to the programs and Control Panels in the template or to new templates.
- A project can contain many programs, and each program can be written in FBD, FFLD, IL, SFC or ST programming languages.
- Additional axes can be added to the template applications that have three, four, five, or more axes.

This project contains two axes where Axis 2 is slaved to Axis 1 at a 2:1 ratio.

15.16.9.3.1 PLC Programs

The 2-axes PLCopen template has an SFC program (called **Main**) that initializes and starts the motion.

- The motion includes single axis jogging, incremental and absolute moves, and gearing motion between axes.
- Controls are included to enable/disable the drive, clear faults, and display fault and axis position information in the Control Panel.

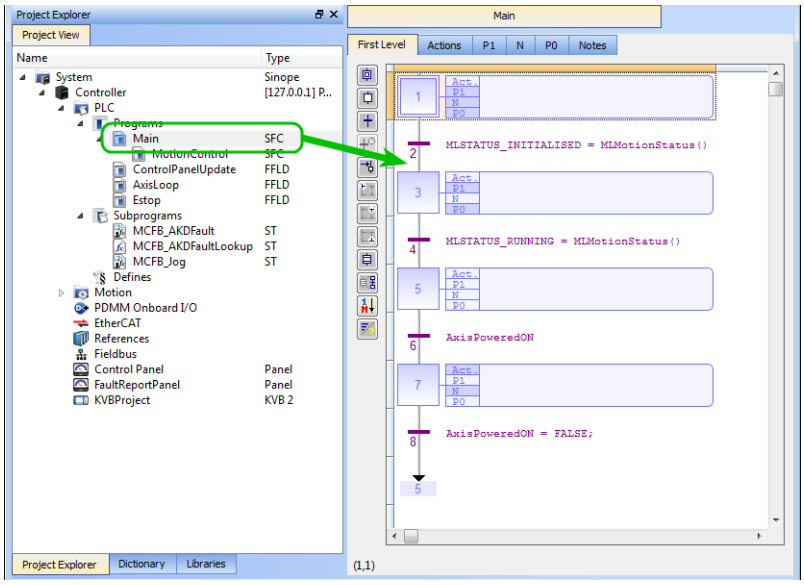


Figure 19-147: PLCopen - Template Main

Step 5 of the Main program in the PLCopen template contains the FFLD code for running the motion. As defined below with the MoveVelocity function block, the motion profile is based on a trapezoidal acceleration / deceleration.

See "S-curve and Trapezoidal Acceleration / Deceleration" (➔ p. 376).

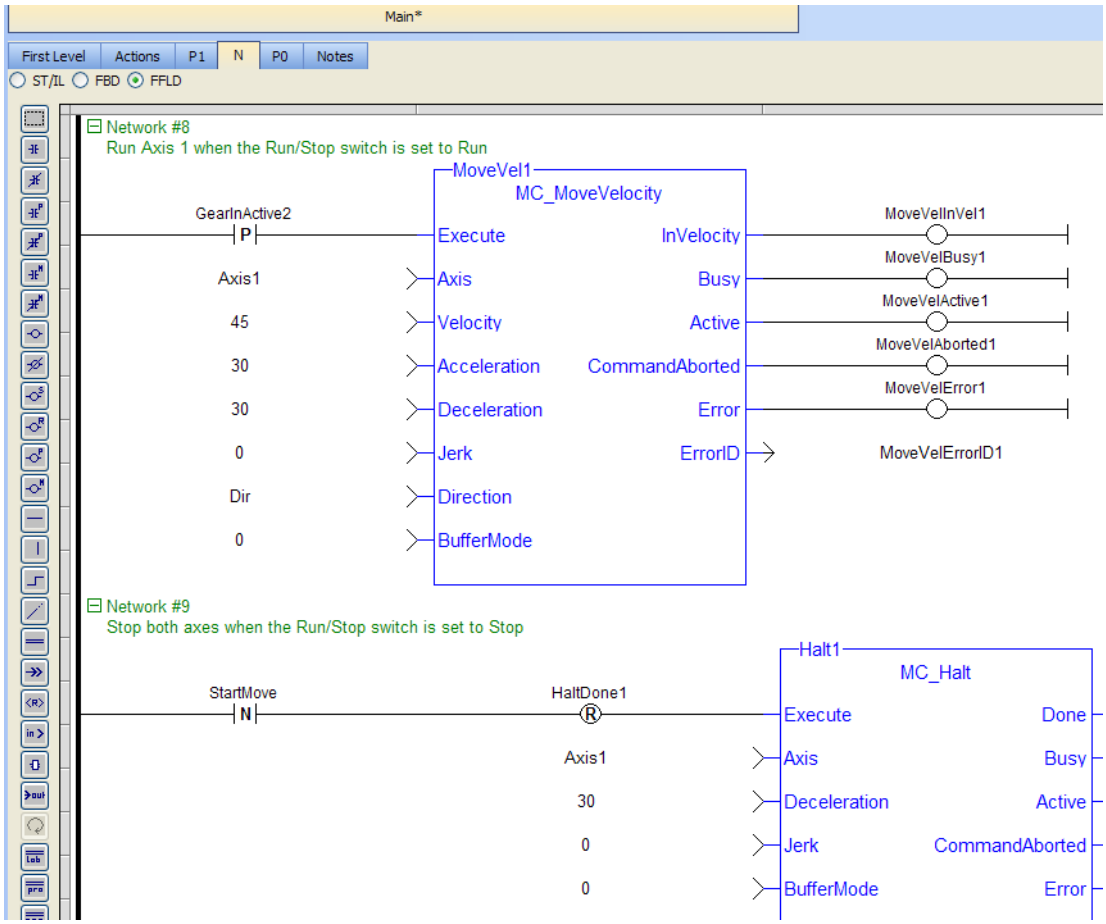


Figure 19-148: PLCopen Template - Step 5 of the Main

The template contains two PLCopen Servo axes where User Units, Update Rate, Rollover Position, and Axis Limits are defined as:

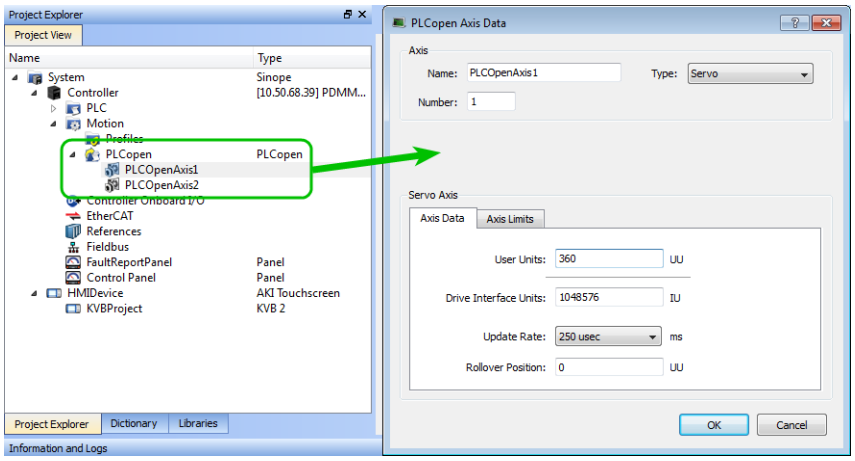


Figure 19-149: PLCopen Template - Motion

See "Axis Data Parameters" (➔ p. 202) for more information about PLCopen axis parameters.

The template contains a Control Panel which works inside of the IDE when running a project.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

It is a useful tool to run and debug programs.

15.16.9.3.2.1 Control Panel Screens

The Control Panel has these screens:

- **Control Panel**
- **Fault Report Panel**

15.16.9.3.2.2 Control Panel Sections

The Control Panel has these sections:

- **Multi-Axis Commands**: Controls motion and shows gearing.
- **Single Axis Motion Commands**: Shows motion at the single axis level and resetting the axis position.
- **Axis Status**: Shows basic axis information.

See "Design the HMI with the Internal Editor" (➔ p. 715) for more information.

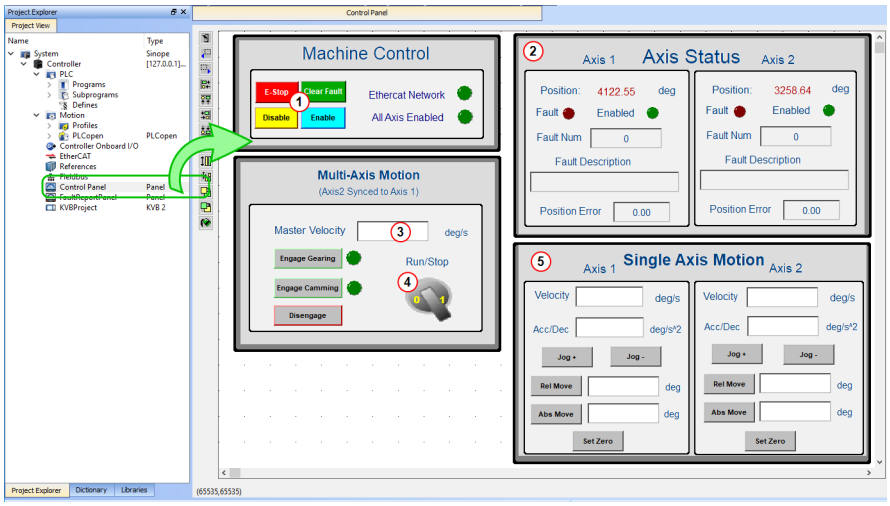


Figure 19-150: PLCopen Template - Control Panel

Call out#	Description
1	Allows for enabling or disabling the axes. After an emergency stop, select the Reset and Enable commands before running the axes.
2	Displays the actual position for each axis.
3	Used to set the speed.
4	Start or stop the motion on the condition that the axes are enable. The green light must be switched on.
5	Control the axis motion.

Fault Report Panel provides more detailed information about when a drive fault occurs.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

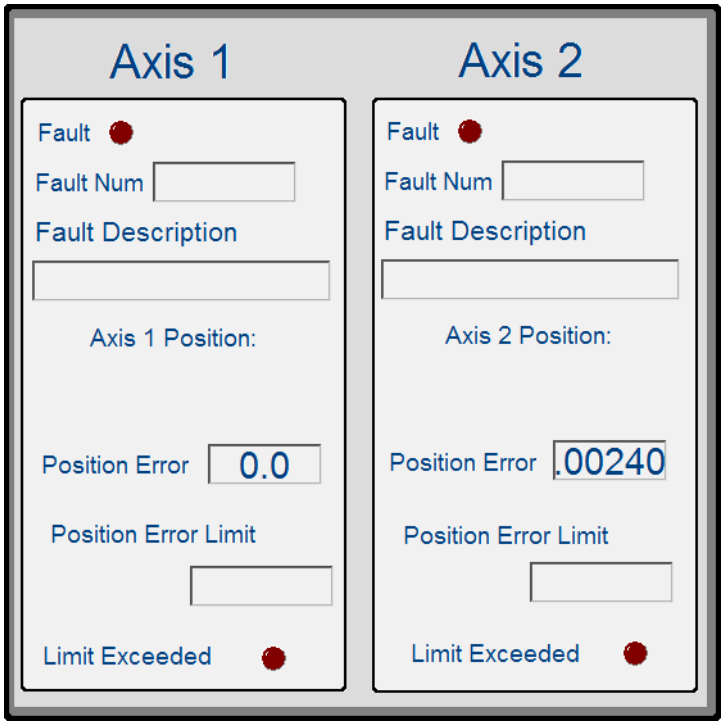


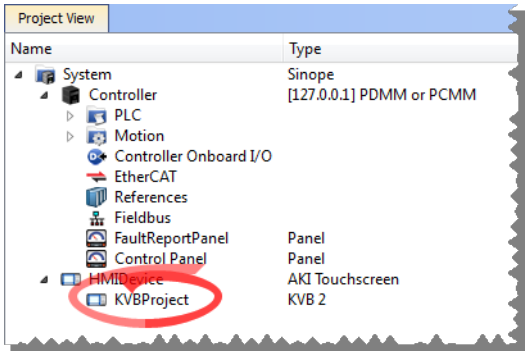
Figure 19-151: PLCopen Template - Fault Report Panel

Based on the template, the project can be run using either the:

- "KAS Simulator Graphical User Interface" (→ p. 586).
- Actual drives and motors.
  - In this case, set up the axes first in the EtherCAT part).
    See "Configure an EtherCAT" (→ p. 535) for more information.

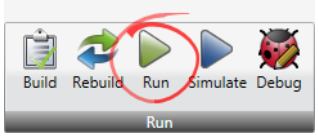
Included in the project template are HMI screens for the AKI terminals created in the KVB (Kollmorgen Visualization Builder) environment.

- The KVB software is a separate installation from the KAS-IDE.
- In KAS-IDE, double-click the **KVB Project** item in the project tree to open KVB.



The project can be run in the KVB programming environment after the KAS-IDE program has started running.

Click on the Run button in the Project ribbon to start the program.

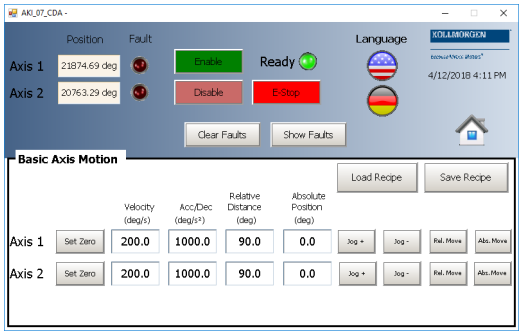


These screens, localized in English and German, are included that duplicate the controls on the IDE Internal Control panel screens.

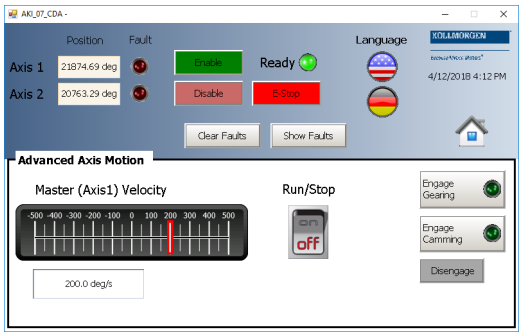
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



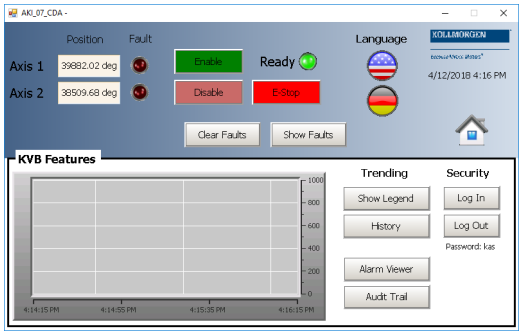
Home Screen



Basic Motion



Advanced Motion



KVB Features

15.16.9.4 PLCopen 2-Axes Template - ST

This template project contains a two-axis template for getting started with a KAS application.

- It contains key project elements that save development time.
- The templates are designed to add code to the programs and Control Panels in the template or to new templates.
- A project can contain many programs, and each program can be written in FBD, FFLD, IL, SFC or ST programming languages.
- Additional axes can be added to the template applications that have three, four, five, or more axes.

15.16.9.4.1 PLC Programs

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

This 2-axis PLCopen template has a ST program (called **Main**) that initializes, starts and runs the motion. The motion includes single axis jogging, incremental and absolute moves, and gearing motion between axes. Also included are controls to enable/disable the drives, clear faults, and display fault and axis position information in the Control Panel. A second program, FaultMonitor, reads the drive fault and position information.

The programs are commented to ease their use.

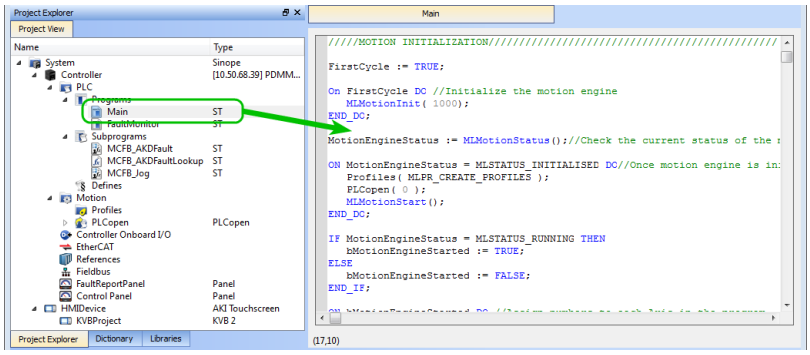


Figure 19-152: PLCopen Template with ST - Main

Also included are three subprograms (**MCFB_AKDFault**, **MCFB_AKDFaultLookup**, and **MCFB_Jog**) that are Kollmorgen UDFBs and can be found in the Project Library section of the IDE.

The template contains two PLCopen Servo axes where User Units, Update Rate, Rollover Position, and Axis Limits are defined as:

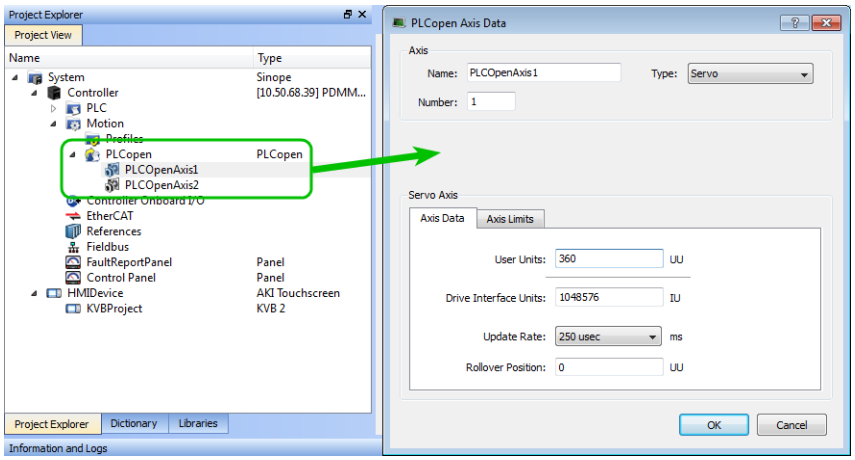


Figure 19-153: PLCopen Template - Motion

See "Axis Data Parameters" (➔ p. 202) for more information about PLCopen axis parameters.

The template contains a Control Panel which works inside of the IDE when running a project.

It is a useful tool to run and debug programs.

15.16.9.4.2.1 Control Panel Screens

The Control Panel has these screens:

- **Control Panel**
- **Fault Report Panel**

15.16.9.4.3.2 Control Panel Sections

The Control Panel has these sections:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- **Multi-Axis Commands**: Controls motion and shows gearing.
- **Single Axis Motion Commands**: Shows motion at the single axis level and resetting the axis position.
- **Axis Status**: Shows basic axis information.

See "Design the HMI with the Internal Editor" (➔ p. 715) for more information.

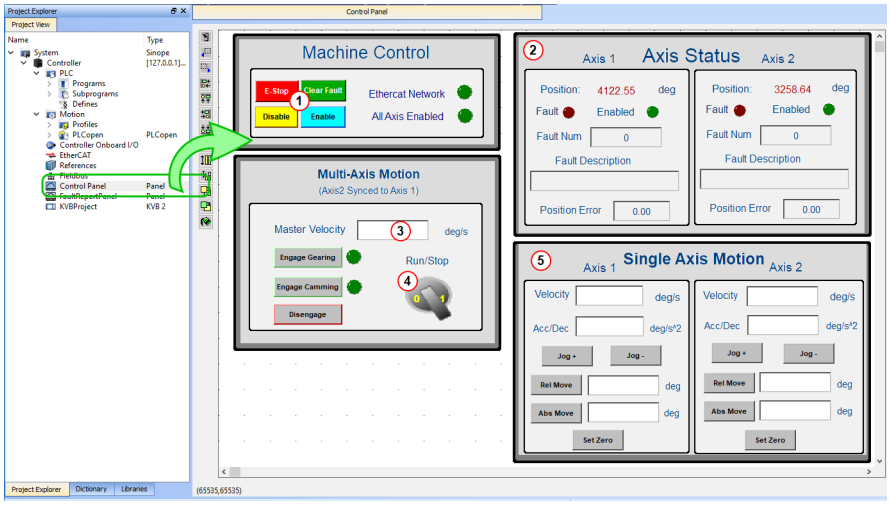


Figure 19-154: PLCopen Template - Control Panel

Call out#	Description
1	Allows for enabling or disabling the axes. After an emergency stop, select the Reset and Enable commands before running the axes.
2	Displays the actual position for each axis.
3	Used to set the speed.
4	Start or stop the motion on the condition that the axes are enable. The green light must be switched on.
5	Control the axis motion.

Fault Report Panel provides more detailed information about when a drive fault occurs.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

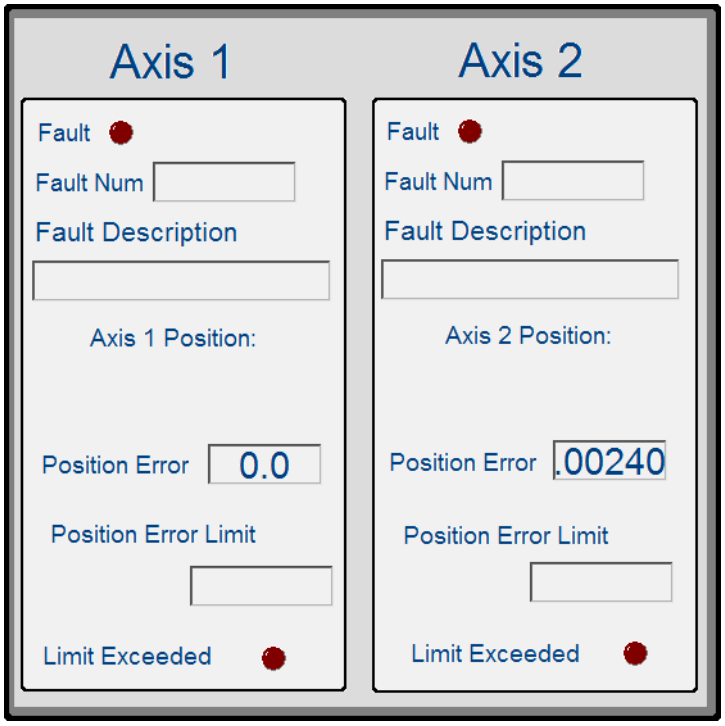
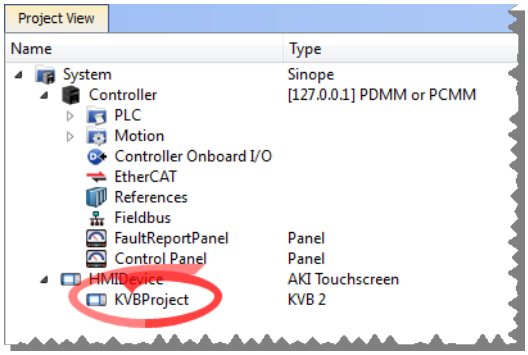


Figure 19-155: PLCopen Template - Fault Report Panel

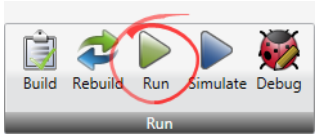
Included in the project template are HMI screens for the AKI terminals created in the KVB (Kollmorgen Visualization Builder) environment.

- The KVB software is a separate installation from the KAS-IDE.
- In KAS-IDE, double-click the **KVB Project** item in the project tree to open KVB.



The project can be run in the KVB programming environment after the KAS-IDE program has started running.

Click on the Run button in the Project ribbon to start the program.

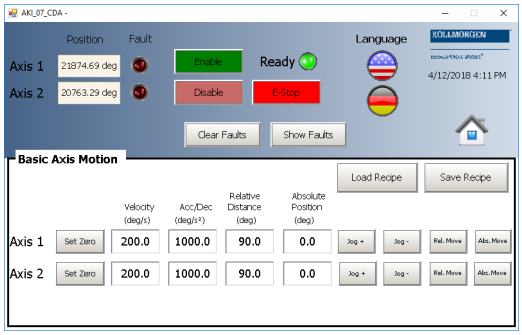


These screens, localized in English and German, are included that duplicate the controls on the IDE Internal Control panel screens.

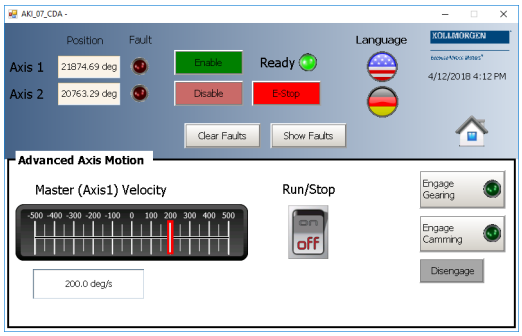
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



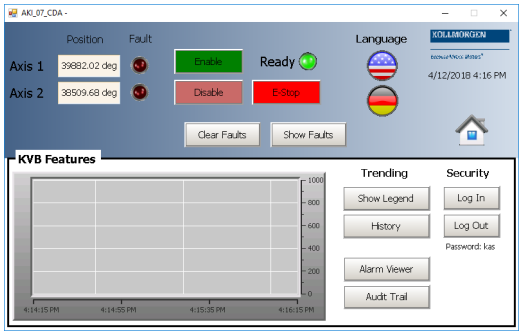
Home Screen



Basic Motion



Advanced Motion



KVB Features

15.16.9.5 PLCopen 2-Axes Template - FFLD

This template project contains a two-axis template for getting started with a KAS application.

- It contains key project elements that save development time.
- The templates are designed to add code to the programs and Control Panels in the template or to new templates.
- A project can contain many programs, and each program can be written in FBD, FFLD, IL, SFC or ST programming languages.
- Additional axes can be added to the template applications that have three, four, five, or more axes.

This project contains two axes where Axis 2 is slaved to Axis 1 at a 2:1 ratio.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "Control Panel" (➔ p. 1048)
- "HMI" (➔ p. 1049)
- "Motion" (➔ p. 1050)
- "PLC Programs" (➔ p. 1051)

15.16.9.5.1 Control Panel

The template contains a Control Panel which works inside of the IDE when running a project.

It is a useful tool to run and debug programs.

15.16.9.5.2.1 Control Panel Screens

The Control Panel has these screens:

- **Control Panel**
- **Fault Report Panel**

15.16.9.5.3.2 Control Panel Sections

The Control Panel has these sections:

- **Multi-Axis Commands**: Controls motion and shows gearing.
- **Single Axis Motion Commands**: Shows motion at the single axis level and resetting the axis position.
- **Axis Status**: Shows basic axis information.

See "Design the HMI with the Internal Editor" (➔ p. 715) for more information.

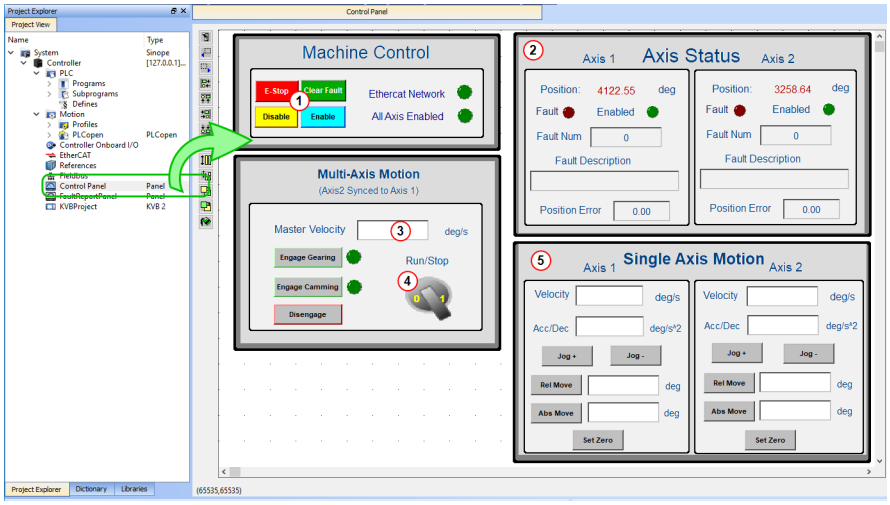


Figure 19-156: PLCopen Template - Control Panel

Call out#	Description
1	Allows for enabling or disabling the axes. After an emergency stop, select the Reset and Enable commands before running the axes.
2	Displays the actual position for each axis.
3	Used to set the speed.
4	Start or stop the motion on the condition that the axes are enable. The green light must be switched on.
5	Control the axis motion.

Fault Report Panel provides more detailed information about when a drive fault occurs.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

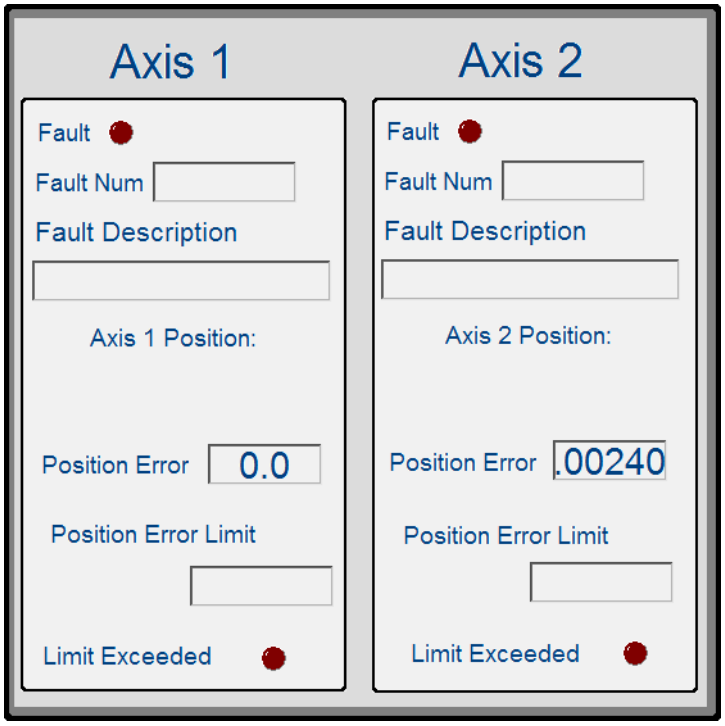
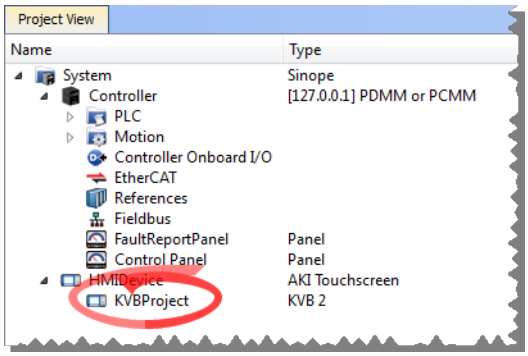


Figure 19-157: PLCopen Template - Fault Report Panel

15.16.9.5.4 HMI

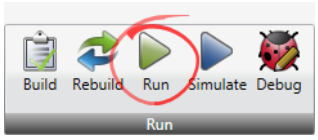
Included in the project template are HMI screens for the AKI terminals created in the KVB (Kollmorgen Visualization Builder) environment.

- The KVB software is a separate installation from the KAS-IDE.
- In KAS-IDE, double-click the **KVB Project** item in the project tree to open KVB.



The project can be run in the KVB programming environment after the KAS-IDE program has started running.

Click on the Run button in the Project ribbon to start the program.

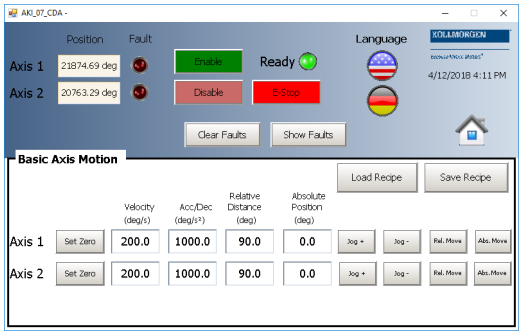


These screens, localized in English and German, are included that duplicate the controls on the IDE Internal Control panel screens.

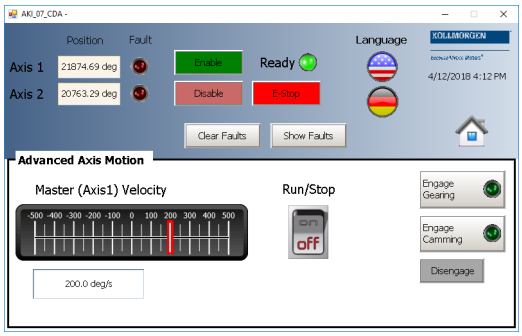
See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.



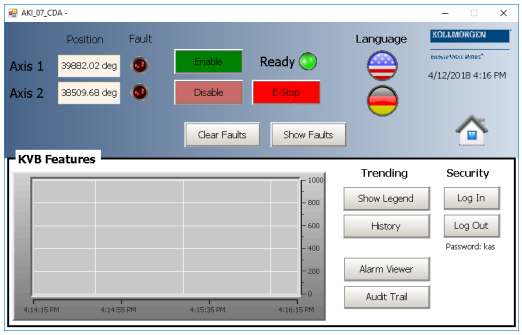
Home Screen



Basic Motion



Advanced Motion



KVB Features

15.16.9.5.5 Motion

The template contains two PLCopen Servo axes where User Units, Update Rate, Rollover Position, and Axis Limits are defined as:

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

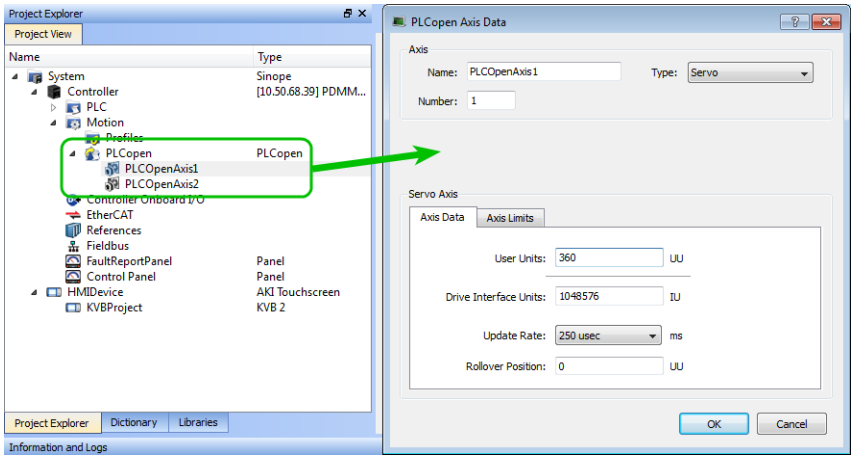


Figure 19-158: PLCopen Template - Motion

See "Axis Data Parameters" (➔ p. 202) for more information about PLCopen axis parameters.

15.16.9.5.6 PLC Programs

The 2-axes PLCopen template has an FFLD program (called **Main**) that initializes and starts the motion.

- The motion includes:
  - Single axis jogging, incremental and absolute moves, and gearing motion between axes.
  - Controls to enable/disable the drives, clear faults, and display fault and axis position information in the Control Panel.

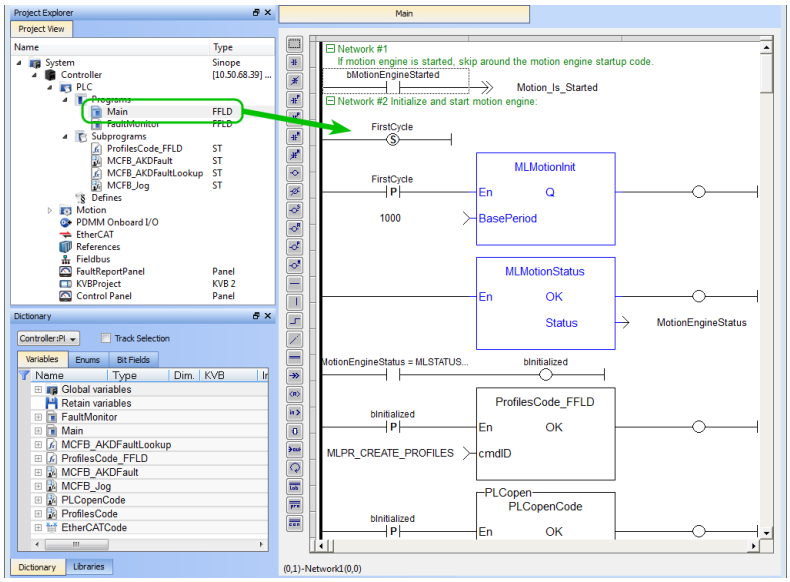


Figure 19-159: PLCopen Template with FFLD - Main

16 Appendices: Lists of Manuals, Content, and Support Information

This section provides lists of available manuals, tables, figures, how-to's, etc.

16.1 PCMM2G - File Naming Conventions

There are different KAS files used with the PCMM2G.

Each of these files have unique naming conventions to identify what is included in the file.

NOTICE

- The PCMM2G firmware image is common across all available controller models.
- Model-specific functionality and axis count limits are determined by the controller configuration and licensing, not by separate firmware images.

16.1.1 PCMM2G Firmware Files

- "Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (→ p. 1053)
- "Recovery Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (→ p. 1054)
- "KAS Runtime (RT) Only - File Naming Convention" (→ p. 1056)
- "Boot Loader - File Naming Convention" (→ p. 1056)

16.1.2 PCMM2G Operating Source (OS) and License Files

- "Operating System (OS) Build Sources - File Naming Convention" (→ p. 1052)
- "Operating System (OS) License Sources - File Naming Convention" (→ p. 1053)

16.1.3 Operating System (OS) Build Sources - File Naming Convention

16.1.3.1 File Contents

This .ZIP file contains the PCMM2G's OS build source scripts.

16.1.3.2 Usage

The contents of this .ZIP file are used to build the PCMM2G's OS image.

NOTE

See the README.md file in the .ZIP file for build instructions and details about what is included in the OS image.

16.1.3.3 File Format and Example

File Name Format	Example File Name
`KAS-PCMM2G-OS-Build-Sources-{OS-Version}.zip`	`KAS-PCMM2G-OS-Build-Sources-1.0.0.00289.zip`

See Also

- "PCMM2G - File Naming Conventions" (→ p. 1052)
- "Operating System (OS) License Sources - File Naming Convention" (→ p. 1053)
- "Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (→ p. 1053)
- "Recovery Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (→ p. 1054)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- "KAS Runtime (RT) Only - File Naming Convention" (➔ p. 1056)
- "Recover Using the Web server" (➔ p. 672)

16.1.4 Operating System (OS) License Sources - File Naming Convention

16.1.4.1 File Contents

This .ZIP file contains:

- The source code of the PCMM2G's OS, bootloader (U-Boot), and their software components.
- Kollmorgen's custom modifications to the OS sources.
- Copyright notices, license texts, and disclaimers of warranty.

16.1.4.2 Usage

The contents of this .ZIP file are used to:

- Review the licenses of all the packages used in the PCMM2G OS.
- Identify custom modifications made to these packages.
- The files in this .ZIP are read-only.

16.1.4.3 File Format and Example

File Name Format	Example File Name
`KAS-PCMM2G-OS-License-Sources-{OS-Version}.zip`	`KAS-PCMM2G-OS-License-Sources-1.0.0.00289.zip`

See Also

- "PCMM2G - File Naming Conventions" (➔ p. 1052)
- "Operating System (OS) License Sources - File Naming Convention" (➔ p. 1053)
- "Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (➔ p. 1053)
- "Recovery Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (➔ p. 1054)
- "KAS Runtime (RT) Only - File Naming Convention" (➔ p. 1056)
- "Recover Using the Web server" (➔ p. 672)

16.1.5 Operating System (OS) + KAS Runtime (RT) - File Naming Convention

16.1.5.1 File Contents

This .ZIP file contains the PCMM2G's OS and RT files.

16.1.5.2 Usage

The contents of this .ZIP file are used to update the PCMM2G from the web server interface.

NOTICE

- The PCMM2G firmware image is common across all available controller models.
- Model-specific functionality and axis count limits are determined by the controller configuration and licensing, not by separate firmware images.

16.1.5.3 File Format and Example

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

File Name Format	Example File Name
KAS-PCMM2G-{Processor/RAM Code}-{Storage-Code}-{OS-Version}-{RT-Version}.zip	KAS-PCMM2G-Cx-08-OS-1.0.0.00288-RT-4.01.0.91998.zip

16.1.5.4 Model Code

This table shows the specified model's CPU speed, number of cores, RAM, and storage size.

Processor / RAM / Storage Code	CPU Speed	Core	RAM	Storage
Cx-08	1.5GHz	Quad	2 / 4 / 8GB	8GB
Cx-16	1.5GHz	Quad	2 / 4 / 8GB	16GB
Cx-32	1.5GHz	Quad	2 / 4 / 8GB	32GB

See Also

- "PCMM2G - File Naming Conventions" (→ p. 1052)
- "Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (→ p. 1053)
- "Operating System (OS) License Sources - File Naming Convention" (→ p. 1053)
- "Recovery Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (→ p. 1054)
- "KAS Runtime (RT) Only - File Naming Convention" (→ p. 1056)
- "Recover Using the Web server" (→ p. 672)

16.1.6 Recovery Operating System (OS) + KAS Runtime (RT) - File Naming Convention

16.1.6.1 File Contents

This .BZ2 file contains the factory default OS and the RT files.

16.1.6.2 Usage

The contents of this .BZ2 file are used to rebuild the PCMM2G when the main flash becomes corrupted.

Customers flash this .BZ2 file onto a USB flash drive and use it to recover their broken PCMM2G.

NOTICE

- The PCMM2G firmware image is common across all available controller models.
- Model-specific functionality and axis count limits are determined by the controller configuration and licensing, not by separate firmware images.

16.1.6.2.1 Reset Changes

The changes which occur during a reset are:

Erased Items

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

- User data files.
- Any application previously downloaded.
- Retained variables.
- Shared directory settings.
- Log files.

Defaults Restored

- Password for both the web page and SSH login.
  - See "SSH Login to a Controller" (→ p. 617) and "User Authentication" (→ p. 623).
- Controller network configuration.
  - Examples: IP address, Subnet, and Gateway settings.
- Auto-Start option.

Security

- The SSH keys are regenerated.
  - SSH clients logged on during the factory reset are forced to log in again.
  - All Date and Time settings are reset to default (NTP enabled).

16.1.6.3 File Format and Example

File Name Format	Example File Name
`KAS-PCMM2G-{Processor/RAM Code}-{Storage Code}-Recovery-{OS-Version}-{RT-Version}.wic.bz2`	`KAS-PCMM2G-Cx-08-Recovery-OS-1.0.0.00288-RT-4.01.0.91998.wic.bz2`

16.1.6.4 Model Code

This table shows the specified model's CPU speed, number of cores, RAM, and storage size.

Processor / RAM / Storage Code	CPU Speed	Core	RAM	Storage
Cx-08	1.5GHz	Quad	2 / 4 / 8GB	8GB
Cx-16	1.5GHz	Quad	2 / 4 / 8GB	16GB
Cx-32	1.5GHz	Quad	2 / 4 / 8GB	32GB

See Also

- "PCMM2G - File Naming Conventions" (→ p. 1052)
- "Recovery Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (→ p. 1054)
- "Operating System (OS) License Sources - File Naming Convention" (→ p. 1053)
- "Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (→ p. 1053)
- "KAS Runtime (RT) Only - File Naming Convention" (→ p. 1056)
- "Recover Using the Web server" (→ p. 672)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

16.1.7 Boot Loader - File Naming Convention

16.1.7.1 File Contents

This .ZIP file contains the PCMM2G's boot loader file.

The PCMM2G controller uses a Kollmorgen customized version of the U-Boot boot loader.

See "3rd-party Licenses" (➔ p. 1069) for licensing terms and downloading instructions.

16.1.7.2 Usage

The contents of this .ZIP file are used to update the PCMM2G's boot loader from the Web server interface.

16.1.7.3 File Format and Example

File Name Format	Example File Name
`KAS-PCMM2G-{Processor/RAM Code}-{Storage-Code}-bootimage-{Version}.zip`	`KAS-PCMM2G-Cx-08-bootimage-2022.01.01.zip`

16.1.7.4 Model Code

This table shows the specified model's CPU speed, number of cores, RAM, and storage size.

Processor / RAM / Storage Code	CPU Speed	Core	RAM	Storage
Cx-08	1.5GHz	Quad	2 / 4 / 8GB	8GB
Cx-16	1.5GHz	Quad	2 / 4 / 8GB	16GB
Cx-32	1.5GHz	Quad	2 / 4 / 8GB	32GB

See Also

- "PCMM2G - File Naming Conventions" (➔ p. 1052)
- "Operating System (OS) License Sources - File Naming Convention" (➔ p. 1053)
- "Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (➔ p. 1053)
- "Recovery Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (➔ p. 1054)
- "Recover Using the Web server" (➔ p. 672)
- "Upgrade the Controller's Firmware" (➔ p. 668)

16.1.8 KAS Runtime (RT) Only - File Naming Convention

This is an unofficial runtime patch file.

16.1.8.1 File Contents

This .ZIP file contains only the RT software.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

16.1.8.2 Usage

This file may be sent to individual users when troubleshooting specific issues.

16.1.8.3 File Format and Example

File Name Format	Example File Name
`KAS-PCMM2G-{Processor/RAM Code}-{Storage Code}-{RT-Version}.zip`	`KAS-PCMM2G-Cx-08-RT-4.01.0.91998.zip`

16.1.8.4 Model Code

This table shows the specified model's CPU speed, number of cores, RAM, and storage size.

Processor / RAM / Storage Code	CPU Speed	Core	RAM	Storage
Cx-08	1.5GHz	Quad	2 / 4 / 8GB	8GB
Cx-16	1.5GHz	Quad	2 / 4 / 8GB	16GB
Cx-32	1.5GHz	Quad	2 / 4 / 8GB	32GB

See Also

- "PCMM2G - File Naming Conventions" (→ p. 1052)
- "Operating System (OS) License Sources - File Naming Convention" (→ p. 1053)
- "Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (→ p. 1053)
- "Recovery Operating System (OS) + KAS Runtime (RT) - File Naming Convention" (→ p. 1054)
- "Recover Using the Web server" (→ p. 672)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

17 List of Documentation

This is the available documentation for these components:

- "KAS Documentation" (→ p. 1061)
- "AKD, AKD2G, S300, and S700 Documentation" (→ p. 1058)
- "HMI Documentation" (→ p. 1058)
- "IO Documentation" (→ p. 1059)
  - "AKT (K-Bus) Terminals" (→ p. 1059)
  - "AKT2G-IO (EtherCAT) Terminals" (→ p. 1060)
- "KVB Documentation" (→ p. 1062)
- "PCMM2G Documentation" (→ p. 1063)

17.1 AKD, AKD2G, S300, and S700 Documentation

Drive Manuals	Description
AKD EtherCAT Manual	Describes the installation, setup, range of functions, and software protocol for the EtherCAT AKD product series.
AKD Installation Manual	• Installation manual for AKD and AKD PDMM drives. • Describes the AKD series of digital drives and includes mechanical, electrical, and software installation information needed to safely install AKD.
AKD, AKD2G, S700 (in NA) Accessories Manual	Describes the accessories for Kollmorgen digital drive systems and servo drive motors.
AKD2G EtherCAT Manual	Describes the installation, setup, range of functions, and software protocol for the AKD2G product series.
AKD2G Installation Manual	• Describes the AKD2G series of digital drives. • Includes mechanical, electrical, software, and functional safety options.
MKD Installation Manual	• Installation manual for MKD drive system modules: • MKD-C power supply. • MKD-N servo drives. • It includes mechanical, electrical and software installation information needed to safely install the MKD system modules.
S300 Reference Documentation	Kollmorgen website with access to all S300 manuals.
S700 Reference Documentation	Kollmorgen website with access to all S700 manuals.

17.2 HMI Documentation

NOTE

Download the applicable documentation from here: KAS HMI Panel Downloads.

HMI Part Number	Description	KVB
AKI-CDC-MOD-21T-000	Graphical Display 21.5" TFT LCD, Touchscreen	v2.0
AKI-CDC-MOD-15T-000	Graphical Display 15.4" TFT LCD, Touchscreen	v2.0
AKI-CDC-MOD-12T-000	Graphical Display 12.1" TFT LCD, Touchscreen	v2.0
AKI2G-CDA-MOD-07T-000	Graphical Display 5" TFT LCD, Touchscreen	v2.20
AKI2G-CDA-MOD-05T-000	Graphical Display 7" TFT LCD, Touchscreen	v2.20

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

HMI Part Number	Description	KVB
AKI2G-CDB-MOD-12T-000	Graphical Display 12" TFT LCD, Touchscreen	v2.20
AKI2G-CDB-MOD-07T-000	Graphical Display 7" TFT LCD, Touchscreen	v2.20

17.3 IO Documentation

- "AKT2G-IO (EtherCAT) Terminals" (➔ p. 1060)
- "AKT (K-Bus) Terminals" (➔ p. 1059)

17.3.1 AKT (K-Bus) Terminals

IMPORTANT

- 24-volt power is NOT passed through AKT-AN-200-000 and AKT-AN-400-000 thermocouple modules.
- These are the solutions to get 24 VDC power to devices that need it (e.g., an AKT_AT-410-000 Output module):
  - Place the module requiring 24 VDC before the thermocouple module.
  - Add a power feed module (AKT-PS-024-000 or AKT-PSF-024-000) after the thermocouple module.

I/O Terminal Part Number	Description	Replaced By
AKT-AN-200-000	2-channel thermocouple input module	AKT2G-AN-240-000
AKT-AN-400-000	4-channel thermocouple input module	AKT2G-AN-400-000
AKT-AN-410-000	• 4-channel analog input module • 0 to 10 VDC	AKT2G-AN-430-000
AKT-AN-420-000	• 4-channel analog input module • 0 to 20 mA	AKT2G-AN-430-000
AKT-AN-810-000	• 8-channel analog input module • 0 to 10 VDC	AKT2G-AN-430-000
AKT-AN-820-000	• 8-channel analog input module • 0 to 20 mA	AKT2G-AN-430-000
AKT-AT-220-000	• 2-channel analog output module • 0 to 20 mA	N/A
AKT-AT-410-000	• 4-channel analog output module • 0 to 10 VDC	N/A
AKT-AT-420-000	• 4-channel analog output module • 0 to 20 mA	N/A
AKT-AT-810-000	• 8-channel analog output module • 0 to 10 VDC	AKT2G-AT-410-000
AKT-AT-820-000	• 8-channel analog output module • 0 to 20 mA	N/A
AKT-DN-004-000	• 4-channel digital input module • 24 VDC • 3.0ms	AKT2G-DN-008-000
AKT-DN-008-000	• 8-channel digital input module • 24 VDC • 3.0ms	AKT2G-DN-008-000

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

I/O Terminal Part Number	Description	Replaced By
AKT-DNH-004-000	• 4-channel digital input module • 24 VDC • 0.2ms	AKT2G-DNH-008-000
AKT-DNH-008-000	• 8-channel digital input module • 24 VDC • 0.2ms	AKT2G-DNH-008-000
AKT-DT-004-000	• 4-channel digital output module • 24 VDC • 0.5 A	AKT2G-DT-008-000
AKT-DT-008-000	• 8-channel digital output module • 24 VDC • 0.5 A	AKT2G-DT-008-000
AKT-DT-2RT-000	• 2-channel Relay Output Module • 230VAC • 2.0A Rel.2NO Pot.-Free	N/A
AKT-ECT-000-000	EtherCAT Bus Coupler	AKT2G-ECT-000-000
AKT-EM-000-000	Standard-Bus End Terminal	AKT2G-EM-000-000
AKT-IM-000-000	Isolation / Separation Terminal	N/A
AKT-PS-024-000	Power Supply, 24 VDC	AKT2G-PSF-024-000
AKT-PSF-024-000	Fused Power Supply with diagnostics, 24 VDC	AKT2G-PSF-024-000
AKT-SM-L15-000	• Stepper motor terminal • 24 VDC • 1.5 A	AKT2G-SM-L15-000
AKT-SM-L50-000	• Stepper motor terminal • 50 VDC • 5A	AKT2G-SM-L50-000

17.3.2 AKT2G-IO (EtherCAT) Terminals

I/O Terminal Part Number	I/O Terminal Description
AKT2G-AC-FAN-001	Fan cartridge for EtherCAT and Bus Terminals.
AKT2G-AN-240-000	• 2-channel input terminal PT100 (RTD) for resistance sensors. • 16-bit, 2-, 3-wire system.
AKT2G-AN-400-000	• 4-channel thermocouple input terminal. • Preset to type K, with wire breakage detection. • 16-bit
AKT2G-AN-430-000	• 4-channel analog input. • Parameter capable • -10/0 to 10V • -20/0/+4 to +20mA • 16-bit
AKT2G-AT-410-000	• 4-channel analog output terminal. • 0 to 10V • 1-wire system • 12-bit

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

I/O Terminal Part Number	I/O Terminal Description
AKT2G-AT-425-000	• 4-channel analog output terminal. • -10V to +10V • 4 x 2-wire system • 12-bit
AKT2G-BRC-000-000	Brake Chopper Terminal
AKT2G-DN-002-000	• Up/down counter 24 VDC • 100 kHz • 32-bit counter depth
AKT2G-DN-008-000	• 8-channel digital input terminal 24 VDC. • Filter 3.0ms • 1-wire system
AKT2G-DNH-008-000	• 8-channel digital input terminal 24 VDC. • Filter 10µs • 1-wire system
AKT2G]-DT-008-000	• 8-channel digital output terminal 24 VDC • 0.5 A • 1-wire system
AKT2G-ECT-000-000	EtherCAT Coupler for E-bus terminals
AKT2G-EM-000-000	• Bus end cover for E-bus terminals • Cover for power and E-bus contacts, gray
AKT2G-ENC-180-000	• 1-channel incremental encoder interface • 32-bit
AKT2G-ENC-190-000	• Incremental encoder interface with differential input • 16-bit / 32-bit
AKT2G-PSF-024-000	• Power supply terminal with fuse • 24 VDC
AKT2G-SDI-004-000	• 4-channel digital input terminal 24 VDC. • Safety
AKT2G-SDO-004-000	• 4-channel digital input terminal 24 VDC. • Safety • 0.5 A
AKT2G-SM-L15-000	• Stepper motor terminal • 24 VDC • 1.5 A • Vector control
AKT2G-SM-L50-000	• Stepper motor terminal • 50 VDC • 5A • Vector control

17.4 KAS Documentation

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

Document Title	Description
Release Notes	The KAS Release Notes contain a summary of: • New features, fixed and known limitations, workarounds • Information on all hardware and software components that have been updated, changed, or added in the release. • See KAS Release Notes EN.
Getting Started	Contains these main steps to get the KAS system up and running. • **Hardware (HW) Installation** (Connection and Wiring): Wiring and hardware details, connectors, system diagrams. • **HW Configuration**: Basic configuration and settings needed to start the HW components (e.g., HMI, Industrial PC, Fieldbus, I/O). • **Software (SW) Installation**: KAS software setup. • See KAS IDE Getting Started EN.
30 Minutes to Motion	Contains the main topics to get started quickly with KAS-IDE. The objective is to familiarize you with the basic principles and the way the program works by creating a simple motion application project. See KAS IDE 30 Minutes to Motion EN about: • Key Features in KAS. • Explore the Workspace • See "KAS Graphical User Interface Description" (→ p. 83). • Build a Motion Project • **Start Projects** - Create a project from scratch or modify an existing project. See Create a Project. • **Add Components** - Add necessary elements to build the project and control the motion part of the system. This includes: "Create Variables" (→ p. 515), "Add Motion" (→ p. 187), and "Programs: Structure and Syntax" (→ p. 481). • **Build Output** - Select a device and generate the application to deliver to users. See "Run the Project" (→ p. 257). • **Run Output** - Make the output accessible to end-users.
KAS-IDE User Manual	Contains content to help with KAS-IDE except the topics included in the Reference Manuals. See KAS-IDE User Manual EN.
KAS-IDE PLC Library Reference Manual	Contains Technical References on **PLC** Programming Languages and Library. See KAS-IDE PLC Library Reference Manual EN.
KAS-IDE Motion Library Reference Manual	Contains Technical References on **Motion** Library for Pipe Network and PLCopen. See KAS-IDE Motion Library Reference Manual EN.

17.5 KVB Documentation

KVB Manuals	Description
Quick Start Guide	The Quick Start covers these important points to: • Install and use Kollmorgen Visualization Builder. • Configure HMI Panels and PC operated control applications.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

KVB Manuals	Description
User Manual	Contains all the content to help with Kollmorgen Visualization Builder.

17.6 PCMM2G Documentation

Product Guides - PCMM2G	
Document	**Description**
PCMM2G Installation Manual	Installation and setup of the controller.
PCMM2G Safety Notes	Safety requirements of the controller.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

18 List of How-Tos

- "Advanced Motion How-Tos" (➜ p. 1064)
- "EtherCAT Fieldbus How-Tos" (➜ p. 1064)
- "Hardware How-Tos" (➜ p. 1064)
- "PLC Code How-Tos" (➜ p. 1064)
- "Run the Application How-Tos" (➜ p. 1064)

18.1 Advanced Motion How-Tos

- "How do I implement Torque Feed-forward?" (➜ p. 793)
- Use "Fast Inputs with Pipe Network Motion" (➜ p. 859)
- Use "Fast Inputs with PLCopen Motion" (➜ p. 859)
- "Digitizing Axis - PLCopen Setup" (➜ p. 855)

18.2 EtherCAT Fieldbus How-Tos

- "Create a Digitizing Axis" (➜ p. 851)

18.3 Hardware How-Tos

- "Download the KVB Panel" (➜ p. 714)
- "FAQs" (➜ p. 781)
- "Verify the NVRAM Space" (➜ p. 771)

18.4 PLC Code How-Tos

- "Change a Pipe Network Link" (➜ p. 188)
- Control an SFC Child Program
- "Create a PLCopen Axis" (➜ p. 195)
- "Draw SFC Divergences" (➜ p. 507)
- "Read the Output of an MC Function Block in ST" (➜ p. 514)
- "How can variables be arranged and/or sorted?" (➜ p. 105)
- Start and Stop a PLC Application using a Command Line Tool
- "Toggle the Connection to Make it Negative" (➜ p. 488)
- "Location Details" (➜ p. 131)

18.5 Run the Application How-Tos

- "Activate an Online Change" (➜ p. 864)
- "Change the Priority between PLC Programs" (➜ p. 452)
- "Log Messages Settings" (➜ p. 124)
- "Export the Oscilloscope Data" (➜ p. 756)
- "Plugging Probes, Mapping Variables" (➜ p. 749)
- "Plug Motion Variables" (➜ p. 274)
- "Plug PLC Variables" (➜ p. 275)
- Set "Breakpoints" (➜ p. 270)
- Specify the Duration of a Cycle using the "EtherCAT Master Settings tab" (➜ p. 539)

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

19 Kollmorgen Documentation Legal Information

The information in these pages is only for the technical documentation (e.g., Safety Notes, User Manuals, etc.) legal information.

This information does not represent Kollmorgen company-wide legal information.

19.1 KAS - Copyrights, Trademarks, Patents, Licenses, and Disclaimers 1066
19.2 Kollmorgen Trademarks 1066
19.3 Kollmorgen Patents 1066
19.4 PCMM2G 1067
19.5 Other Trademarks 1068
19.6 3rd-party Licenses 1069
19.7 Disclaimers 1071

19.1 KAS - Copyrights, Trademarks, Patents, Licenses, and Disclaimers

Copyright 2026 Regal Rexnord Corporation, All Rights Reserved.

Information in this document is subject to change without notice. The software package described in this document is furnished under a license agreement. The software package may be used or copied only in accordance with the terms of the license agreement.

This document is the intellectual property of Kollmorgen and contains proprietary and confidential information. The reproduction, modification, translation or disclosure to third parties of this document (in whole or in part) is strictly prohibited without the prior written permission of Kollmorgen.

Copyright © 2009-2026 Regal Rexnord Corporation, All Rights Reserved.

19.2 Kollmorgen Trademarks

Regal Rexnord and Kollmorgen are trademarks of Regal Rexnord Corporation or one of its affiliated companies.

These trademarks or trade names are owned by or under the control of Regal Rexnord Corporation or any of its affiliates:

- Kollmorgen Essentials System
- Kollmorgen Essentials Cable
- Kollmorgen Essentials Drive
- Kollmorgen Essentials Motor
- AKD
- AKD2G
- AKD PDMM
- AKM
- AKMA
- AKME
- AKMH
- BLM
- Cartridge DDR
- DDL
- DDR
- Goldline
- Kollmorgen Automation Suite
- KAS
- Kollmorgen Visualization Builder
- KVB
- MKD
- Motioneering
- P8000
- PCMM
- PCMM2G
- PowerMax
- Servostar
- SafeMotion Monitor
- SMM
- TBM
- TBM2G
- WorkBench

19.3 Kollmorgen Patents

Any of these Kollmorgen patents may be used and/or identified in Kollmorgen products, firmware, and/or software.

- US Patent 2017/0211640 (method and apparatus for power saving, fail-safe control of an electromechanical brake), patent pending.
- US Patent 8,154,228 (Dynamic Braking For Electric Motors).
- US Patent 8,214,063 (Auto-tune of a Control System Based on Frequency Response).
- US Patent 8,566.415 (Safe Torque Off over network wiring).
- US Patent 10,374,468 (System and method for improved DC power line communication).
- US Patent 10,520,050 (Method and apparatus for power-saving, fail-safe control of an electromechanical brake).
- US Patent 10,840,696 (Method and apparatus for limiting the output voltages of switch mode power supplies).
- US Patent 16,247,478 (method and apparatus for limiting the output voltages of switching mode power supplies), patent pending.

Patents referring to fieldbus functions are listed in the matching fieldbus manual.

19.4 PCMM2G

The PCMM2G's Operating System (OS) and boot loader (U-Boot) are based on free and open-source software.

- The individual copyright notices, license texts, and disclaimers of warranty for the software components are located in the controller, under the directory: /usr/share/common-licenses/.
  - For more information about accessing the PCMM2G's files, see SSH Login to a Controller.
- The OS, bootloader, and their software component's source codes including modifications, copyright notices, license texts, disclaimers of warranty, and the compilation scripts to build the OS image are available at the Kollmorgen webpage PCMM2G Operating System (OS) License Sources.
- The OS image and its corresponding sources file is identified by an "OS-Sources" designator, followed by its version number: OS-Sources-x.xx.x.xxxxx.
- The compilation scripts and sources file are available at the Kollmorgen webpage PCMM2G Operating System (OS) Build Sources.
- See PCMM2G - File Naming Conventions in the KAS online help.

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.

19.5 Other Trademarks

Any of these other trademarks and/or trade names may be used and/or identified in Kollmorgen products, firmware, and/or software.

These are believed to be the trademarks and/or trade names of their respective owners and are not owned or controlled by Regal Rexnord Corporation:

- BiSS: iC-Haus GmbH.
- CANopen: CAN in Automation (CiA).
- DRIVE-CLiQ: Siemens Aktiengesellschaft.
- EnDat: Dr. Johannes Heidenhain GmbH.
- EtherCAT und Safety over EtherCAT: Beckhoff Automation GmbH.
- EtherCAT: Beckhoff Automation GmbH.
- EtherNet/IP Kommunikationsstack: Rockwell Automation.
- EtherNet/IP: ODVA, Inc.
- Ghostscript: Artifex Software, Inc. (unter der AGPL Lizenz verbreitet).
- HIPERFACE DSL: SICK AG.
- HIPERFACE: SICK AG.
- HUMMEL: HUMMEL AG.
- Hysol: Hysol Huawei Electronics Co,.LTD.
- Instapak: Sealed Air Corporation.
- INTERCONTEC: INTERCONTEC Pfeiffer Industrie-Steckverbindungen GmbH.
- LOCTITE 640: Henkel AG & CO. KGAA.
- LOCTITE: Henkel AG & CO. KGAA.
- Modbus: Schneider Electric USA, Inc.
- MOLYKOTE: DDP Specialty Electronic Materials US 9, LLC.
- Mylar: DUPONT Teijin Films U.S.
- P3-topax: Henkel KGaA.
- PHOENIX CONTACT: Phoenix Contact GmbH & Co. KG.
- PLCopen: An association registered with the Chamber of Commerce in The Netherlands (Reg. 40240111).
- PROFINET: PROFIBUS Nutzerorganisation e.V.
- Scotch-Weld 2214: 3M Company.
- sercos: Sercos International E.V.
- SIMATIC: Siemens Aktiengesellschaft.
- SpeedTec: Liqui-Moly GmbH or TE Connectivity Industrial GmbH.
  If SpeedTec is used as a lubricant, it belongs to Liqui-Moly GmbH.
  If SpeedTec is used as a connector, it belongs to TE Connectivity Industrial GmbH
- Teflon: The Chemours Company FC, LLC.
- Viton: The Chemours Company FC, LLC.
- Windows: Microsoft Corporation.

19.6 3rd-party Licenses

Any of these 3rd-party licenses may be used and/or identified in Kollmorgen products, firmware, and/or software.

Library	Product	License
7-zip	https://www.7-zip.org/	7-zip - License
Adafruit core graphics library	https://github.com/adafruit	https://github.com/adafruit/Adafruit-GFX-Library/blob/master/license.txt
Adafruit display library	https://github.com/adafruit	https://github.com/adafruit/Adafruit-ST7735-Library/blob/master/README.txt
Arduino Map function	http://creativecommons.org	http://creativecommons.org/licenses/by-sa/3.0/
C++ Mathematical Expression Library	http://www.partow.net/programming/exprtk/index.html	MIT License
civetweb: Embedded C/C++ web server	https://github.com/civetweb/civetweb	MIT License
CoreSight Library	http://www.apache.org	http://www.apache.org/licenses/LICENSE-2.0
curl software library		curl license
DockPanelSuite	http://sourceforge.net/projects/dockpanelsuite/	https://opensource.org/licenses/MIT
EtherNet/IP Communication Stack		Software Distribution License
FatFs	https://github.com/abbrev/fatfs	https://github.com/stm32duino/FatFs/blob/master/Licence.md
FileHelpers	http://sourceforge.net/projects/filehelpers	https://opensource.org/licenses/MIT
Font Awesome	https://fontawesome.com	https://fontawesome.com/license/free
Ghostscript	https://ghostscript.com/index.html	AGPL
Google Fonts Roboto Condensed	http://www.apache.org	http://www.apache.org/licenses/
GSL - Guideline Support Library	https://github.com/Microsoft/GSL	https://github.com/Microsoft/GSL/blob/master/LICENSE

Library	Product	License
libsodium	https://libsodium.org	https://github.com/jedisct1/libsodium/blob/master/LICENSE
jsoncpp software		jsoncpp License
log4net	http://logging.apache.org/log4net/	http://www.apache.org/licenses/LICENSE-2.0
LVGL	https://github.com/lvgl/lvgl	https://github.com/lvgl/lvgl/blob/master/LICENCE.txt
LwIP	https://savannah.nongnu.org	https://savannah.nongnu.org/projects/lwip/
mbed Microcontroller Library	http://www.apache.org	http://www.apache.org/licenses/LICENSE-2.0
mbed TLS	http://www.apache.org	http://www.apache.org/licenses/LICENSE-2.0
Microsoft Edge WebView2	https://developer.microsoft.com/en-us/microsoft-edge/webview2	Microsoft Edge WebView2 Runtime LICENSE
MigraDoc	http://www.pdfsharp.net/migradocoverview.ashx	http://www.pdfsharp.net/MigraDoc_License.ashx
Mongoose Library	https://github.com/cesanta/mongoose/tree/	Mongoose license
MVVM Light Toolkit	https://github.com/lbugnion/mvvmlight	https://galasoft.ch/license_MIT.txt
OpENer EtherNet/IP	https://github.com/EIPStackGroup/OpENer	http://eipstackgroup.github.io/OpENer/de/d7e/license.html
PdfSharp	http://www.pdfsharp.net/migradocoverview.ashx	http://www.pdfsharp.net/PDFsharp_License.ashx
Qt Framework	https://www.qt.io/product/framework	terms of the LGPL3
Qwt project	http://qwt.sourceforge.net/	Qwt
SharpZipLib	https://github.com/icsharpcode/SharpZipLib	https://github.com/icsharpcode/SharpZipLib/blob/master/LICENSE.txt
SNMP		http://www.net-snmp.org/about/license.html
SQLite	https://system.data.sqlite.org	https://system.data.sqlite.org/index.html/doc/trunk/www/copyright.wiki
U-Boot	http://www.denx.de/wiki/U-Boot	GNU General Public License 2.0
Zed Graph	http://sourceforge.net/projects/zedgraph/	https://opensource.org/licenses/LGPL-2.0
Zlib software library	https://www.zlib.net/	Zlib License

19.7 Disclaimers

Technical changes which improve the performance of the device may be made without prior notice!

This document is the intellectual property of Kollmorgen. All rights reserved. No part of this work may be reproduced in any form (by photocopying, microfilm or any other method) or stored, processed, copied or distributed by electronic means without the written permission of Kollmorgen.

The information in this document is believed to be accurate and reliable at the time of its release. Kollmorgen assumes no responsibility for any damage or loss resulting from the use of this help, and expressly disclaims any liability or damages for loss of data, loss of use, and property damage of any kind, direct, incidental or consequential, in regard to or arising out of the performance or form of the materials presented herein or in any software programs that accompany this document.

All timing diagrams, whether produced by Kollmorgen or included by courtesy of the PLCopen organization, are provided with accuracy on a best-effort basis with no warranty, explicit or implied, by Kollmorgen. The user releases Kollmorgen from any liability arising out of the use of these timing diagrams.

Technische Änderungen, die der Verbesserung der Geräte dienen, vorbehalten!

Dieses Dokument ist geistiges Eigentum von Kollmorgen. Alle Rechte vorbehalten. Kein Teil dieses Werkes darf in irgendeiner Form (Fotokopie, Mikrofilm oder in einem anderen Verfahren) ohne schriftliche Genehmigung von Kollmorgen reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Die Genauigkeit und Verlässlichkeit der Informationen in diesem Dokument entspricht dem Kenntnisstand zum Zeitpunkt der Veröffentlichung. Kollmorgen übernimmt keine Verantwortung für Schäden oder Verluste, die sich aus der Verwendung dieser Hilfe ergeben, und lehnt ausdrücklich jegliche Haftung sowie Schadensersatzansprüche für Datenverlust, entgangene Nutzung und Sachschäden jeglicher Art ab, inklusive direkter, zufälliger oder Folgeschäden im Zusammenhang mit der Funktion oder Form des hierin oder in einem der mit diesem Dokument gelieferten Software enthaltenen Materials.

Es posible que se implementen sin aviso previo cambios técnicos que mejoran el rendimiento del dispositivo.

La propiedad intelectual de este documento pertenece a Kollmorgen. Todos los derechos reservados. Se prohíbe la reproducción de este documento por ningún medio (fotocopia, microfilm u otro método), así como su almacenamiento, procesamiento, copia o distribución por medios electrónicos, sin el consentimiento por escrito de Kollmorgen.

La información incluida en este documento se considera correcta y confiable en el momento de su publicación. Kollmorgen no asume ninguna responsabilidad por ningún daño o pérdida causados por el uso de este documento de ayuda y rechaza explícitamente toda responsabilidad o daño por pérdida de datos, pérdida de uso y daños en la propiedad de cualquier tipo, ya sea directos, incidentales o consecuentes, en relación con el rendimiento o la forma de los materiales aquí publicados, o con los programas de software que acompañan a este documento, o que surjan de estos.

Les modifications techniques qui améliorent les performances du dispositif peuvent être apportées sans avis préalable !

Ce document est la propriété intellectuelle de Kollmorgen. Tous droits réservés. Aucune partie de ce document ne peut être reproduite sous quelque forme que ce soit (photocopie, microfilm ou autre méthode) ni stockée, traitée, copiée ou distribuée de manière électronique sans l'autorisation écrite de Kollmorgen.

Les informations contenues dans ce document sont considérées exactes et fiables au moment de leur publication. Kollmorgen n'assume aucune responsabilité pour tout dommage ou perte résultant de l'utilisation de cette aide, et rejette expressément toute responsabilité ou tous dommages pour la perte de données, la perte d'utilisation et les dommages matériels de toute nature, directs, accessoires ou indirects, en ce qui concerne ou découlant des performances ou de la forme des matériaux présentés ici ou dans tout programme logiciel accompagnant ce document.

Modifiche tecniche volte a migliorare le prestazioni del dispositivo possono essere apportate senza preavviso.

Il presente documento è proprietà intellettuale di Kollmorgen. Tutti i diritti riservati. È vietata la riproduzione di qualsiasi parte di quest'opera in qualsiasi forma (fotocopia, microfilm o qualunque altro metodo) così come la memorizzazione, il trattamento, la copia o la distribuzione tramite mezzi elettronici senza l'autorizzazione scritta di Kollmorgen.

Le informazioni contenute nel presente documento sono ritenute accurate e affidabili al momento della pubblicazione. Kollmorgen non si assume alcuna responsabilità per eventuali danni o perdite risultanti dall'uso di questa guida e declina espressamente qualsiasi responsabilità o danni dovuti a perdita di dati, mancato utilizzo e danni materiali di qualunque tipo, diretti, accidentali o consequenziali, in relazione a o derivanti dal funzionamento o dalla forma dei materiali presentati qui o in eventuali programmi software che accompagnano il presente documento.

デバイス性能の向上のため、予告なく技術的な変更が実装される場合があります。

本書の知的財産権はKollmorgenに帰属しており、無断複写・転載は禁じられています。本書のいずれの部分も、Kollmorgenの書面による許可なしに、いかなる複製(写真、マイクロフィルムまたはその他の手段)、保管、改変、複写または電子的配布を行うことはできません。

本書に記載されている情報は、発行時において正確かつ信頼できるものですが、Kollmorgenはこのヘルプの使用により生じるいかなる損害や損失についても責任を負いません。また、本書の履行、資料形式または付随するあらゆるソフトウェアプログラムに関連して、あるいはこれらが原因で発生する、直接的、偶発的または結果的な、いかなるデータの損失、利用機会の喪失および財産の損害に対しても、一切の責任や損害賠償を明示的に否認します。

기기의 성능을 향상시키는 기술적 변경 사항은 사전 공지 없이 이루어질 수 있습니다!

본 문서의 지적 재산권은 Kollmorgen에 있습니다. All rights reserved. 본 저작물의 어떤 부분도 Kollmorgen의 서면 허가 없이 어떠한 형태(사진 복사, 마이크로필름 또는 기타 방법)로든 복제하거나 전자 수단으로 저장, 처리, 복사 또는 배포할 수 없습니다.

본 문서의 정보는 공개 시점을 기준으로 정확하고 신뢰할 수 있는 것으로 간주됩니다. Kollmorgen은 본 도움말의 사용으로 인한 어떠한 피해나 손실에 대해 책임을 지지 않으며, 본 문서 또는 본 문서와 함께 제공되는 소프트웨어 프로그램에 제시된 자료의 성능 또는 형태와 관련하여 또는 그로 인해 직접적, 우발적 또는 결과적으로 발생한 모든 유형의 데이터 손실, 사용 손실, 재산상의 피해에 대한 책임이나 손해를 명시적으로 부인합니다.

Alterações técnicas que melhoram o desempenho do dispositivo podem ser feitas sem aviso prévio!

Este documento é propriedade intelectual da Kollmorgen. Todos os direitos reservados. Nenhuma parte deste trabalho pode ser reproduzida sob qualquer forma (por fotocópia, microfilme ou qualquer outro método) ou armazenada, processada, copiada ou distribuída por meios eletrônicos sem a permissão escrita da Kollmorgen.

Acredita-se que as informações contidas neste documento sejam precisas e confiáveis no momento de sua divulgação. A Kollmorgen não assume nenhuma responsabilidade por quaisquer danos ou perdas resultantes do uso desta ajuda, e se isenta expressamente de qualquer responsabilidade ou danos por perda de dados, perda de uso e danos materiais de qualquer espécie, direta, incidental ou subsequente em relação a ou resultante do desempenho ou da forma dos materiais apresentados aqui ou em quaisquer programas de software que acompanhem este documento.

我们可能会进行技术更改以提高设备性能，恕不另行通知！

本文档知识产权归 Kollmorgen 所有，Kollmorgen 保留所有权利。未经 Kollmorgen书面许可，不得以任何形式（影印、缩微胶片或其他任何方式）复制本文档任何内容，亦不得以任何电子手段存储、处理、复制或分发本文档的任何内容。

据我们所知，本文档所含信息截至发布日是准确可靠的。对于因使用本帮助文档而导致的任何损害或损失，Kollmorgen概不承担任何责任。Kollmorgen特此明确声明，对于因本文档中的任何材料或随附本文档之任何软件程序的性能或形式导致的或与之相关的任何类型的数据丢失、使用中断损失或财产损害（包括直接、间接和衍生性损失或损害），Kollmorgen不承担任何法律责任或损害赔偿责任。

Support and Services

About Kollmorgen

When you need motion and automation systems for your most demanding applications and environments, count on Kollmorgen - the innovation leader for more than 100 years. We deliver the industry's highest-performing, most reliable motors, drives, AGV control solutions and automation platforms, with over a million standard and easily modifiable products to meet virtually any motion challenge. We offer manufacturing facilities, distributors and engineering expertise in all major regions around the world, so you can bring a better machine to market faster and keep it profitable for many years to come.

Kollmorgen Support Network



See the Kollmorgen Support Network for product support, product downloads, knowledge base answers, and information.

Kollmorgen Support Locations

North America
Kollmorgen Corporation
201 West Rock Road
Radford, VA 24141, USA

Web:	www.kollmorgen.com
Email:	kollmorgen.support@regalrexnord.com
Tel.:	+1-540-633-3545
Fax:	+1-540-639-4162

Europe
Kollmorgen Europe GmbH
Pempelfurtstr. 1
40880 Ratingen, Germany

Web:	www.kollmorgen.com
Email:	kollmorgen.tech.emeai@regalrexnord.com
Tel.:	+49-2102-9394-0
Fax:	+49-2102-9394-3155

South America
Altra Industrial Motion do Brasil Equipamentos Industriais Ltda.
Avenida João Paulo Ablas, 2970
Jardim da Glória, Cotia – SP
CEP 06711-250, Brazil

Web:	www.kollmorgen.com
Email:	kollmorgen.contato@regalrexnord.com
Tel.:	(+55 11) 4615-6300

China and SEA
Kollmorgen (Shanghai)
Room 1201, Building A,
Mangoo Hub, No.7 Longai Road,
Xuhui District, Shanghai, China
(Service available in English and Chinese)

Web:	www.kollmorgen.cn
Email:	kollmorgen.motion.sales.china@regalrexnord.com
Tel.:	+86-400 668 2802

See Copyrights, Trademarks, Patents, Licenses, and Disclaimers for legal ownership.