Kollmorgen Automation Suite KAS IDE User Guide



Kollmorgen Automation Suite

INTEGRATED DEVELOPMENT ENVIRONMENT

"Because Motion Matters"

Kollmorgen Automation Suite Integrated Development Environment combines a market leading Motion Engine, tightly integrated PLC engine and HMI developer with the ability to configure an array of motion hardware such as motors, drives, gearboxes and actuators as well as automation hardware such as I/O modules and HMI's. Providing both process control capabilities and unsurpassed motion control programming delivers the very best in machine automation technology.

Document Edition: R, December 2020 Valid for KAS Software Revision 3.05 Valid for AKD firmware version: 01-20-00-001 Valid for AKD2G firmware version: 02-05-00-000 Part Number: 959715



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

Trademarks and Copyrights

Copyrights

Copyright © 2009-2020 Kollmorgen

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.

Trademarks

- KAS and AKD are registered trademarks of Kollmorgen.
- Kollmorgen is part of the Altra Industrial Motion Company.
- EnDat is a registered trademark of Dr. Johannes Heidenhain GmbH
- EtherCAT is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH
- Ethernet/IP is a registered trademark of ODVA, Inc.
- Ethernet/IP Communication Stack: copyright (c) 2009, Rockwell Automation
- HIPERFACE is a registered trademark of Max Stegmann GmbH
- PROFINET is a registered trademark of PROFIBUS and PROFINET International (PI)
- SIMATIC is a registered trademark of SIEMENS AG
- Windows is a registered trademark of Microsoft Corporation
- PLCopen is an independent association providing efficiency in industrial automation.
- Codemeter is a registered trademark of WIBU-Systems AG.
- SyCon® is a registered trademark of Hilscher GmbH.

Kollmorgen Automation Suite is based on the work of:

- 7-zip (distributed under the terms of the LGPL and the BSD 3-clause licenses see terms)
- The C++ Mathematical Expression Library (distributed under the MIT License)
- curl software library
- JsonCpp software (distributed under the MIT License see terms)
- Mongoose software (distributed under the GNU GPL v2 see terms)
- Qt cross-platform SDK (distributed under the terms of the LGPL3; Qt source is available on KDN)
- Qwt project (distributed under the terms of the Qwt License)
- U-Boot, a universal boot loader is used by the AKD PDMM and PCMM (distributed under the terms of the GNU General Public License). The U-Boot source files, copyright notice, and readme are available on the distribution disk that is included with the AKD PDMM and PCMM.
- Zlib software library

All other product and brand names listed in this document may be trademarks or registered trademarks of their respective owners.

Disclaimer

The information in this document (Version R published on 12/8/2020) is believed to be accurate and reliable at the time of its release. Notwithstanding the foregoing, 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.

1 Table of Contents

1 Table of Contents	
2 Document History	
3 Preface	
3.1 System Requirements for KAS IDE and KAS Simulator	
3.2 Learning Kollmorgen Automation Suite	
3.2.1 Where to Start	
3.2.1.1 Beginner	
3.2.1.2 Advanced User	
3.2.1.3 Online	
3.2.2 Read KAS Manuals	
4 Introducing Kollmorgen Automation Suite	
4.1 Key Features	
4.1.1 Integrated Development Environment	
4.1.2 KAS Runtime	
4.1.3 KAS Simulator	
4.2 Looking at Kollmorgen Automation Suite	
4.2.1 Physical View	
4.2.2 Logical View	
4.2.3 Architectural View	
4.3 KAS Breakdown	
4.3.1 Human-Machine Interface	
4.3.2 Controllers	
4.3.2.1 AKD PDMM and PCMM Hardware	
4.3.2.2 PDMM or PCMM Rotary Switch	43
4.3.2.3 PDMM & PCMM Web Server	
4.3.2.4 Controller Performance	44
4.3.3 Communication and Fieldbus	
4.3.3.1 Fieldbus	47
4.3.3.2 Motion bus	47
4.3.4 Input Output Bus	47
4.3.5 Drives	48
4.3.6 Kollmorgen Motors	
4.4 Different Implementations	49
4.4.1 Single-Axis with PDMM	49
4.4.2 Multi-Axis with PDMM or PCMM	50
5 Describing KAS Graphical User Interface	51
5.1 Windows and Panels Overview	51
5.1.1 Main Window	51
5.1.1.1 About toolboxes	52
5.1.2 Project Explorer	53
5.1.2.1 System	
5.1.2.2 Controller	55
5.1.2.3 PLC	57
5.1.2.4 Programs	57

5.1.2.5 Subprograms	58
5.1.2.6 Defines	58
5.1.2.7 Motion	58
5.1.2.8 Profiles	58
5.1.2.9 Pipe Network	59
5.1.2.10 PLCopen	61
5.1.2.11 Control Panel	62
5.1.2.12 Controller Onboard I/O	62
5.1.2.13 EtherCAT	62
5.1.2.14 AKD2G Drive	63
5.1.2.15 AKD Drive	63
5.1.2.16 AKD-C Central Power Supply	63
5.1.2.17 AKD-N Drive	63
5.1.2.18 AKT2G (EtherCAT) I/O Coupler	63
5.1.2.19 AKT (K-Bus) I/O Coupler	64
5.1.2.20 I/O Slice	64
5.1.2.21 Device	64
5.1.2.22 References	65
5.1.2.23 Fieldbus	65
5.1.2.24 HMI Device	65
5.1.2.25 KVB Project	65
5.1.2.26 SafePLC2 Project	65
5.1.3 Libraries	69
5.1.3.1 Function Blocks	69
5.1.3.2 Controls	69
5.1.3.3 Properties	69
5.1.4 Dictionary	70
5.1.4.1 Variables tab	70
5.1.4.2 Enum Tab	84
5.1.4.3 Bit Fields Tab	86
5.1.5 Information and Logs	88
5.1.5.1 Log Messages	88
5.1.5.2 Log Messages Settings	90
5.1.5.3 Find and Replace Tab	
5.1.5.4 Find and Replace Operations	
5.1.5.5 Browse Variable Tab	
5.1.5.6 Breakpoints tab	
5.1.5.7 Compiler Output	
5.1.6 Watch Window	
5.1.6.1 Multiple Watch Windows	104
5.1.6.2 Access Structure and Arrays	
5.1.6.3 Add Variable	
5.1.6.4 Add an Expression	
5.1.6.5 Force a Variable	108
5.1.7 AKD Drive	
5.1.7.1 WorkBench Toolbar	110
5.1.7.2 WorkBench Status Bar	110

	5.1.8 Stat	us Bar	
	5.1.8.1	Local Version	111
	5.1.8.2	Controller Version	111
	5.1.8.3	Drives State	111
	5.1.8.4	Controller State	112
	5.1.8.5	Connection State	112
	5.1.8.6	Color Codes	112
6 (Getting St	arted	113
6.		E to Runtime Compatibility	
6.	2 KAS PL	C Code to Runtime Compatibility	
6.	3 KAS ID	E to Project File Compatibility	116
6.	4 Startin	g the KAS IDE	117
	6.4.1 Find	KAS IDE Version	
	6.4.2 Acce	ess Help System	118
	6.4.3 KAS	GUI	119
	6.4.4 KAS	IDE Log Window	119
	6.4.4.1	Log Information	119
	6.4.5 Crea	ate a Project	119
	6.4.6 Add	and Configure a Controller	
	6.4.6.1	Add the Controller	120
	6.4.6.2	Configure the Controller	
	6.4.7 Con	figure EtherCAT Motion Bus	123
	6.4.7.1	EtherCAT Scan and Association	
	6.4.8 Ove	rview: Create Programs	139
	6.4.8.1	Overview: Create Variables	
		Overview: Create and/or Call Functions & Function Blocks	
		Overview: Using Constants	
		Overview: Map Input & Output to Variables	
		ing Motion	
		Design Motion with Pipe Network	
		Design Motion with PLCopen Axis	
	6.4.9.3	Design Coordinated Motion	161
		d an HMI Device	
		Create KVB Project	
		2 Map Variables to HMI	
		B Design KVB Panel with Kollmorgen Visualization Builder	
		rking with SafePLC2 Projects	
		Connecting to a SafePLC2 Project - Direct Method	
		2 Connecting to a SafePLC2 Project - Import Method	
		Adding or Importing a SafePLC2 Project	
		Exporting a SafePLC2 Project	
		5 Connect SafePLC2 to an SCU-1-EC	
		5 Put It Together: KAS, SCU-1-EC, and the AKD2G Working Together	
		erview: Compile, Connect, Download, and Run	
		vanced Project Creation	
		d and Configure Drive	
	6.4.14.1	Add the Drive	181

	Add and Configure Drive	
6.4.14.3	Configure the AKD2G/AKD Drive	183
6.4.14.4	Save Parameters to Non-Volatile Memory	. 185
	AKD Setup Wizard	
6.4.14.6	Configure Onboard I/O	186
	Digital Input Mode	
6.4.15 Add	and Configure I/O Slices	188
6.4.15.1	Add an AKT (K-Bus) I/O Coupler	188
6.4.15.2	Add an AKT (K-Bus) I/O Slice	188
6.4.15.3	Configure the I/O Slice	189
6.4.16 Add	& Configure Third Party Devices	190
6.4.16.1	Add Modules to Third Party EtherCAT Devices	192
6.4.16.2	Modules — General Properties Tab	193
	Pre-defined Libraries	
6.4.18 Crea	te and Use Custom Libraries	195
6.4.18.1	Create the Custom Library	195
	Use the Custom Library	
	ng Cam Profiles	
6.4.19.1	Create Cam Profiles	199
	Use Cam Profiles	
6.4.19.3	Cam Profile Switching	203
6.4.20 Defir	ne Scheduling	204
	Periodicity	
	Order of Execution	
	Define the PLC Cycle	
	About Parent-Child relationships and execution order	
-	the Project	
	e Compilation Options	
	Conditional Compiling	
•	ile the Application	
	h KAS Simulator	
	ect to the Controller	
	ctions to Prevent Compatibility Issues	
	pplication Status Bar	
	lessage Window	
	load the Application	
	pplication Status Bar	
	e Control	
	tart/stop the Device	
	og Window	
-	and Debugging the Project	
	By-Step Debugging	
	Breakpoints	
6.6.2.1 A	bout Breakpoints	220
	g, Removing, Enabling, and Disabling Breakpoints	
6.6.3.1 H	low to Set Breakpoints	222
6.6.3.2 H	low to Enable/Disable a Breakpoint	222

6.6.4 Printf Function	224
6.6.5 Soft Oscilloscope Debugging	225
6.6.5.1 How to Plug Motion Variables	225
6.6.5.2 How to Plug PLC Variables	227
6.6.6 Compare PLC Programs	229
6.6.7 Variable Animation	231
6.6.7.1 Variable Monitoring	231
6.6.7.2 IEC 61131-3 Editor Debugging	234
6.6.7.3 Pipe Network Editor Debugging	234
6.7 Managing a Project	. 235
6.7.1 Print	236
6.7.1.1 Printable Elements	236
6.7.1.2 Page Setup	236
6.7.1.3 Print	236
6.7.1.4 Print Preview	237
6.7.1.5 Print Project	237
6.7.2 Use the Reference Folder	238
7 KAS IDE Reference	.239
7.1 Choose a Workspace Layout	239
7.1.1 Move Child Windows	239
7.1.2 Move Toolbox	239
7.1.2.1 Dock Window	239
7.2 Menus and Toolbar Overview	241
7.2.1 File Menu	241
7.2.2 Edit Menu	245
7.2.3 Tools Menu	246
7.2.3.1 User Options	246
7.2.4 Window Menu	. 248
7.2.5 Help Menu	. 249
7.2.6 Main Toolbar	250
7.2.7 Device Toolbar	
7.2.8 EtherCAT Toolbar	252
7.2.9 Online Change Toolbar	
7.2.10 Debug Toolbar	
7.3 Windows Standard Conventions	
7.3.1 Windows Manipulation	
7.3.2 Mouse Manipulation	
7.3.3 Table Manipulation	254
7.3.3.1 Sorting Items	
7.3.3.2 Selecting a Cell	
7.3.3.3 Selecting a Row	
7.3.3.4 Resizing a Column	
7.4 Keyboard Shortcuts	
7.4.1 Common Keyboard Shortcuts	
7.4.2 FBD Editor Keyboard Shortcuts	
7.4.2.1 FBD Editor (common)	
7.4.2.2 FBD Editor (when editing)	257

7.4.2.3 FBD Editor (during debug)	257
7.4.3 FFLD Editor Keyboard Shortcuts	258
7.4.3.1 FFLD Editor (when editing)	258
7.4.3.2 FFLD Editor (during debug)	260
7.4.4 SFC Editor Keyboard Shortcuts	261
7.4.5 ST Editor Keyboard Shortcuts	262
7.4.5.1 ST Editor (common)	262
7.4.5.2 ST Editor (when editing)	262
7.4.5.3 ST Editor (during debug)	262
7.4.6 Graphic Editor Keyboard Shortcuts	263
7.4.7 Table Keyboard Shortcuts	263
7.5 Bookmarks	264
7.6 Create Programs	265
7.6.1 Project Structure	265
7.6.2 IEC 61131-3 Editors	265
7.6.3 Some Tips	266
7.6.3.1 About Drag-and-Drop	266
7.6.3.2 About Autocompletion	266
7.6.3.3 About tooltip on variable	266
7.6.3.4 About Bookmarks	266
7.6.4 Select Function Blocks	267
7.6.5 Select Variables and Instances	267
7.6.6 Drag and Drop Programming	267
7.6.7 Autocompletion When Programming	268
7.6.8 Tooltips on Variables	
7.6.9 Selecting Function Blocks	
7.6.10 Selecting Variables and Instances	
7.6.11 Change Operators and Functions in FFLD and FBD Editors	
7.6.12 Change Function Blocks in FFLD and FBD Editors	
7.6.13 Sequential Function Chart (SFC) Editor	272
7.6.13.1 Using the SFC toolbar	273
7.6.13.2 Draw SFC divergences	274
7.6.13.3 View SFC charts	
7.6.13.4 Move or copy SFC charts	275
7.6.13.5 Enter SFC macro-steps	
7.6.13.6 Renumber steps and transitions	
7.6.13.7 Enter actions of a step	
7.6.13.8 Enter the condition of a transition	276
7.6.13.9 Enter notes for steps and transitions	277
7.6.14 Function Block Diagram (FBD) Editor	
7.6.14.1 Using the FBD toolbar	
7.6.14.2 Draw FBD connection lines	282
7.6.14.3 Select FBD variables and instances	283
7.6.14.4 View FBD diagrams	
7.6.14.5 Move or copy FBD objects	
7.6.14.6 Insert FBD objects on a line	
7.6.14.7 Resize FBD objects	

7.6.15 Structured Text (ST) / Instruction List (IL) Editor	
7.6.15.1 ST / IL Language selection	
7.6.15.2 ST / IL Syntax coloring	
7.6.15.3 Intellisense	
7.6.15.4 Auto-completion of words	
7.6.15.5 ST / IL Drag-and-drop features	
7.6.15.6 How to Read Output of a MC Function Block in ST	
7.6.16 Free Form Ladder Diagram (FFLD) Editor	
7.6.16.1 Using the FFLD Editor	294
7.6.16.2 Power rail and lines	
7.6.16.3 Contacts and coils	296
7.6.16.4 Function blocks	
7.6.16.5 Data In/Out	297
7.6.16.6 Jumps and RETURN	297
7.6.16.7 Selection grid	
7.6.16.8 Move and copy items	298
7.6.16.9 View FFLD diagrams	299
7.7 Create Variables	
7.7.1 Use the Dictionary	
7.7.2 Create Variables from the Editors	
7.7.3 Data Types	
7.7.4 Complex Structures	
7.7.5 Variable Editor	
7.7.6 Editing Variables as Text	
7.8 Create Functions and Function Blocks	
7.8.1 Declare Functions or Function Blocks	
7.8.2 Define Parameters and Private Variables	
7.8.3 Finalize Functions or Function Blocks	
7.8.4 Call Functions or Function Blocks	
7.9 Use the Defines List	
7.9.1 Internal Defines	
7.9.2 Global Defines	
7.9.3 Local definitions	
7.10 Map Input and Output to Variables	
7.10 Map Input and Output to Variables7.10.1 Map I/O from the Project Explorer	312 312
 7.10 Map Input and Output to Variables 7.10.1 Map I/O from the Project Explorer 7.10.2 PLC Variable Creation Wizard 	312 312 316
 7.10 Map Input and Output to Variables 7.10.1 Map I/O from the Project Explorer 7.10.2 PLC Variable Creation Wizard 7.10.3 PLC Variable Selector 	
 7.10 Map Input and Output to Variables 7.10.1 Map I/O from the Project Explorer 7.10.2 PLC Variable Creation Wizard 7.10.3 PLC Variable Selector 7.10.3.1 Create PLC Variable Button 	
 7.10 Map Input and Output to Variables 7.10.1 Map I/O from the Project Explorer 7.10.2 PLC Variable Creation Wizard 7.10.3 PLC Variable Selector 7.10.3.1 Create PLC Variable Button 7.10.3.2 Advanced Button 	
 7.10 Map Input and Output to Variables 7.10.1 Map I/O from the Project Explorer 7.10.2 PLC Variable Creation Wizard 7.10.3 PLC Variable Selector 7.10.3.1 Create PLC Variable Button 7.10.3.2 Advanced Button 7.10.4 Analog I/O Parameters 	312 312 316 317 317 317 317 318
 7.10 Map Input and Output to Variables 7.10.1 Map I/O from the Project Explorer 7.10.2 PLC Variable Creation Wizard 7.10.3 PLC Variable Selector 7.10.3.1 Create PLC Variable Button 7.10.3.2 Advanced Button 7.10.4 Analog I/O Parameters 7.10.4.1 AKT2G Input Terminals 	312 312 316 317 317 317 317 318 318
 7.10 Map Input and Output to Variables 7.10.1 Map I/O from the Project Explorer 7.10.2 PLC Variable Creation Wizard 7.10.3 PLC Variable Selector 7.10.3.1 Create PLC Variable Button 7.10.3.2 Advanced Button 7.10.4 Analog I/O Parameters 7.10.4.1 AKT2G Input Terminals 7.10.4.2 AKT2G Output Terminals 	312 312 316 317 317 317 317 318 318 318 318
 7.10 Map Input and Output to Variables 7.10.1 Map I/O from the Project Explorer 7.10.2 PLC Variable Creation Wizard 7.10.3 PLC Variable Selector 7.10.3.1 Create PLC Variable Button 7.10.3.2 Advanced Button 7.10.4 Analog I/O Parameters 7.10.4.1 AKT2G Input Terminals 7.10.4.3 AKT Input Terminals 	312 312 316 317 317 317 317 318 318 318 318 318
 7.10 Map Input and Output to Variables 7.10.1 Map I/O from the Project Explorer 7.10.2 PLC Variable Creation Wizard 7.10.3 PLC Variable Selector 7.10.3.1 Create PLC Variable Button 7.10.3.2 Advanced Button 7.10.4 Analog I/O Parameters 7.10.4.1 AKT2G Input Terminals 7.10.4.2 AKT2G Output Terminals 7.10.4.3 AKT Input Terminals 7.10.4.4 AKT Output Terminals 	312 312 316 317 317 317 317 318 318 318 318 318 318 318
 7.10 Map Input and Output to Variables 7.10.1 Map I/O from the Project Explorer 7.10.2 PLC Variable Creation Wizard 7.10.3 PLC Variable Selector 7.10.3.1 Create PLC Variable Button 7.10.3.2 Advanced Button 7.10.4 Analog I/O Parameters 7.10.4.1 AKT2G Input Terminals 7.10.4.2 AKT2G Output Terminals 7.10.4.3 AKT Input Terminals 7.10.4.4 AKT Output Terminals 7.10.4.4 AKT Output Terminals 7.11 Configuring EtherCAT 	312 312 316 317 317 317 317 318 318 318 318 318 319 320
 7.10 Map Input and Output to Variables 7.10.1 Map I/O from the Project Explorer 7.10.2 PLC Variable Creation Wizard 7.10.3 PLC Variable Selector 7.10.3.1 Create PLC Variable Button 7.10.3.2 Advanced Button 7.10.4 Analog I/O Parameters 7.10.4.1 AKT2G Input Terminals 7.10.4.2 AKT2G Output Terminals 7.10.4.3 AKT Input Terminals 7.10.4.4 AKT Output Terminals 	312 312 316 317 317 317 317 318 318 318 318 318 318 319 320 320

7.11.2 EtherCAT Master Settings Tab	
7.11.3 ENI File Tab	
7.11.4 ESI Files Tab	
7.11.5 FSoE / PDO Connections Tab	
7.11.6 ESI Files with References (MDP)	
7.11.6.1 EtherCAT Device Configuration - MDP Devices	
7.11.7 ESI File Management	
7.12 EtherCAT Devices	
7.12.1 EtherCAT Device General Properties Tab	
7.12.1.1 Information	
7.12.1.2 Topology	
7.12.2 Configuration Tab	
7.12.3 PDO Selection/Mapping Tab	
7.12.3.1 Map PLC Variable to PDO Object	
7.12.3.2 Select Input and Output PDOs	
7.12.3.3 Viewing the contents of a PDO	
7.12.3.4 How To Set Up a Variable to Monitor Safety Data	
7.12.4 PDO Editor Tab	
7.12.4.1 Editable PDOs	
7.12.4.2 PDO Content	
7.12.4.3 PDO Restrictions and Compiler Errors	
7.12.4.4 Redundant PDO Entries	
7.12.4.5 About Safety, PDOs, and Variables	
7.12.5 Distributed Clock tab	
7.12.5.1 Oversampling devices	
7.12.6 CoE Init Commands tab	
7.12.6.1 Adding CoE Commands	
7.12.7 CoE Object Dictionary Tab	
8 Understanding KAS	
8.1 IEC 61131-3	
8.1.1 Introduction	
8.1.2 Data Types	
8.1.2.1 Structures	
8.1.2.2 Arrays	
8.1.3 Variables	
8.1.3.1 About Retain Variables	
8.1.3.2 Working with Variables	
8.1.3.3 Retain Variables	
8.1.4 Constant Expressions	
8.1.4.1 Examples	
8.1.5 Program Organization Units	
8.1.5.1 Difference between Functions and Function Blocks	
8.1.5.2 Functions	
8.1.5.3 Function Blocks	
8.1.5.4 Programs	
8.1.5.5 User-Defined Function Blocks	
8.1.6 Programming Languages	

	Sequential Function Chart (SFC)	
8.1.6.2	Structured Text (ST)	
	Function Block Diagram (FBD)	
8.1.6.4	Free Form Ladder Diagram (FFLD)	
8.1.6.5	Instruction List (IL)	369
	Definitions	
8.1.8 Han	dling Exceptions	
8.1.8.1	Startup Exceptions	371
	Shutdown Exceptions	
	Division By Zero Exceptions	
	Concepts	
	oducing Motion	
	Motion Control Main Functions	
	Single and Multi-Axis Motion	
	Hardware Organization of Motion Functions	
8.2.1.4	Motion Profile	
8.2.1.5	KAS Touch Probe & Fast Input Overview	375
8.2.2 Pipe	Network or PLCopen	
8.2.2.1	Motion Engine Differences	
8.2.3 Pipe	Network Concept	
8.2.3.1	Pipe Network	
8.2.3.2	Pipe	
8.2.3.3	Pipe Block	
8.2.3.4	Axis Pipe Block	
8.2.3.5	Executing Motion	
8.2.3.6	Pipe Block Lifetime	
8.2.3.7	Motion State Machine	
8.2.3.8	Phase Execution in the Pipe Network	
8.2.3.9	Use Motion Function Block for Pipe Network	
8.2.3.10)Function - General rules	
8.2.4 PLC	open®	
8.2.4.1	PLCopen Function Blocks	
8.2.4.2	PLCopen Function Blocks - Overview	
8.2.4.3	PLCopen Function Blocks - General Rules	
8.2.4.4	PLCopen State Machine	426
8.2.4.5	Programming a Dual Core Controller	428
8.3 EtherC	AT Motion Bus Concepts	432
8.3.1 Fund	tional Principle	432
8.3.2 Ethe	rCAT Features	432
8.3.2.1	Protocol	
8.3.2.2	Topology	
8.3.2.3	Distributed Clock (Synchronization)	435
8.3.2.4	Performance	
8.3.2.5	Safety over EtherCAT	
8.3.2.6	Gateways	437
8.3.2.7	Device profiles	
8.3.2.8	File Access over EtherCAT (FoE)	438

8.3.3 EtherCAT Implementation	438
8.3.3.1 Master Configuration	438
8.3.3.2 Slave Configuration	440
8.3.3.3 EtherCAT State Machine	
8.3.3.4 PDOs for AKD, AKD2G, AKD-N, and S300/S700	442
8.3.4 CANopen	451
8.3.4.1 CANopen Status Machine	451
8.3.4.2 CANopen Control Word	454
8.3.4.3 CANopen Status Word	455
8.4 AKD Drive	456
8.4.1 AKD Drive Configuration	456
8.4.1.1 Connection Modes	456
8.4.1.2 AKD Configuration According to EtherCAT State	457
8.5 Tasking Model / Scheduling	459
8.5.1 Priority Between Motion and PLC	459
8.5.1.1 EtherCAT Processing Time	459
8.5.1.2 NVRAM Processing Time	
8.5.1.3 What happens when a PLC Program is overrunning the Cycle Duration	460
8.5.2 Priority Between PLC Programs	461
9 Using the KAS Simulator	462
9.1 Start KAS Simulator	462
9.1.1 Diagnosing What Software is Using Port 80	
9.1.2 KAS Runtime Log Window	463
9.2 Stop KAS Simulator	
9.3 KAS Simulator Axes Tab	
9.4 Describing KAS Simulator Graphical User Interface	
9.4.1 Windows Overview	
9.4.1.1 Main window	
9.4.1.2 KAS Simulator log window	
9.5 KAS Simulator Menus Description	
9.5.1 File Menu	
9.5.2 Help Menu	
10 Using the AKD PDMM or PCMM Controller	
10.1 Booting the PDMM or PCMM	
10.1.1 Boot Sequence	
10.1.2 Boot Startup Script	
10.1.3 Booting from the Recovery Image	
10.2 Working with the Hardware	
10.2.1 AKD PDMM or PCMM Memory	
10.2.2 PDMM or PCMM B3 Button Menu	
10.2.3 Display the IP Address of a PDMM or PCMM	
10.2.4 Booting from the Recovery Image	
10.2.5 Reset the Controller to Factory Settings	
10.2.6 SD Card Support	
10.2.7 Backup and Restore a PDMM or PCMM	4/ጋ

	10.2.7.1 About the data transfer	475
	10.2.7.2 Backup Using the B3 Button	
	10.2.7.3 Restore Using the B3 Button	476
	10.2.8 EtherCAT Devices Backup and Restore	477
	10.2.8.1 EtherCAT Devices Backup	478
	10.2.8.2 EtherCAT Devices Restore	480
	10.2.8.3 AKD Backup/Restore Compatibility	481
	10.2.8.4 Import / Export Backup from Webserver	482
	10.2.8.5 EtherCAT Devices Backup/Restore Limitations	482
	10.2.8.6 Troubleshooting EtherCAT Devices Backup/Restore	483
	10.2.9 Configure Controller Onboard I/O	484
	10.2.10 About Errors and Alarms	
	10.2.11 Using the KAS Web Server	
	10.2.12 Web Server Home Page	
	10.2.12.1 Security	
	10.2.12.2 User Authentication	
	10.2.13 KAS Application	
	10.2.13.1 Axis	
	10.2.13.2 Log Configuration	
	10.2.13.3 Log Data	
	10.2.13.4 User Data	
	10.2.13.5 Shared Directory	
	10.2.14 Web Server Settings	
	10.2.14.1 Firmware Tab (AKD PDMM and PCMM Only)	
	10.2.14.2 Network Tab (PDMM or PCMM Only)	
	10.2.14.3 File System Tab	
	10.2.14.4 SD Card Tab (PDMM or PCMM Only)	
	10.2.14.5 User Account	
	10.2.15 Backup & Restore from the Web Server	
	10.2.15.1 Backup Tab	
	10.2.15.2 Restore Tab 10.2.15.3 Import/Export	
	10.2.16 Diagnostic (PDMM and PCMM Only)	
	10.2.16.1 Errors and Alarms	
	10.2.16.2 Hardware Status	
	10.2.16.3 Crash Reports	
	10.2.16.4 EtherCAT Diagnostics	
1	0.3 Using SSH	
•	10.3.1 Logging In To A Controller Via SSH	
	10.3.2 Change Controller's Password Via SSH	
	10.3.3 How to Restore the Controller's Default Password and/or SSH Settings	
11		
	1.1 Pipe Network Editor	
•	11.1.1 Inserting Pipe Blocks or Comments	
	11.1.2 Inserting Connections	
	11.1.3 Connect a Comment to a Pipe Block	
	11.1.4 Pipe Network Editor Controls	

11.1.5 Plug/Unplug Channels	512
11.2 Cam Profile Editor	513
11.2.1 About the Cam Profile Editor	513
11.2.1.1 Windows Overview	513
11.2.2 Cam Profile Editor's Cam Table	514
11.2.2.1 Modifying an Element using the Cam Table	515
11.2.2.2 Modifying the Type of a Cam Element	
11.2.2.3 Cam Table Contextual Menu	516
11.2.2.4 Adding a Point to the Cam Table	516
11.2.2.5 Remove a Point from the Cam Table	517
11.2.3 Cam Profile Graph	518
11.2.3.1 Modifying an Element	518
11.2.3.2 Cam Profile Graph Contextual Menu	518
11.2.3.3 Zoom In and Out	519
11.2.3.4 Panning	519
11.2.3.5 Restoring Zoom and Pan	519
11.2.4 Cam Profile Segment Overview	520
11.2.5 Curve Selection and Color Table	524
11.2.5.1 How to change color	524
11.2.6 Curves Graph	525
11.2.7 Reload, Save, Auto Fit, and Properties Buttons	525
11.2.8 Import Cam Profile	525
11.2.8.1 About the Import	
11.2.8.2 When Displaying the Imported Cam Profile	
11.2.8.3 About Invalid Data	
11.3 Soft Oscilloscope	
11.3.1 Access the Soft Oscilloscope	
11.3.2 About OpenGL	
11.3.3 The Control Panel	
11.3.4 The Graphical Area	
11.3.5 Traces	
11.3.6 Plugging Probes, Mapping Variables	
11.3.6.1 Mapping a Variable to the Oscilloscope	
11.3.6.2 Plugging a probe from the Dictionary	
11.3.6.3 Plugging a probe from the Pipe Network	
11.3.7 Setting Scale	
11.3.8 Trace Zoom Feature	
11.3.9 How Do I Export the Oscilloscope Data?	
11.3.10 Practical Application: Using Trace Time To Measure CPU Load	
11.3.10.1 Collect some data by pressing the "Start" button	
11.3.10.2 Check the peak times	
11.3.10.3 Heavily Loaded CPU Example	
11.3.10.4 Over Loaded CPU Example	
11.4 Human-Machine Interface Editor	
11.4.1 Using Kollmorgen Visualization Builder	
11.4.1.1 Create A New KVB Controller	
11.4.1.2 Import Variables Into The Project	551

11.4.1.3	Design Your HMI Panel with KVB	553
11.4.1.4	Download the KVB Panel	554
11.4.2 Des	ign the Control Panel with the Internal Control Panel Editor	554
11.4.2.1	Create Control Panel	554
11.4.2.2	Use the Control Panel control library	554
11.4.2.3	Edit the Control panel	555
11.4.2.4	Mapping Variables to the Control Panel	556
11.4.2.5	Graphic Objects	557
11.4.2.6	Graphic Objects Properties	564
11.4.2.7	Operate the Control Panel	569
11.4.2.8	Exiting Simulation Mode	570
12 Advanced	l Topics	
12.1 Coordi	nated Motion	571
12.1.1 Ove	rview	571
12.1.1.1	Coordinated Motion Terminology	571
12.1.1.2	Group State Diagrams	573
12.1.1.3	Coordinate Systems	573
	ign Coordinated Motion	
	Create a Linear or Circular Coordinated Motion Application	
	Blending Between Moves	
	Blending with Transitions	
	Transition Between Moves	
	What Does MC_GrpHalt Do?	
	What Does MC_GrpStop Do?	
	Differences between MC_GrpHalt and MC_GrpStop	
	Handling Axis Errors	
	1 Techniques	
	Online Change	
	What is Online Change	
	How to Activate Online Change	
	What is the Revert button	
	Difference between Local and Controller versions	
	Pulse Limitations with Online Change	
	ig PLC Online Change	
	Set up an application	
	Enable Online Change	
	Using Online Change	
	Revert Online Change	
	it Are Fast Inputs?	
	Fast Inputs with Pipe Network Motion	
	Fast Inputs with PLCopen Motion	
	que Feed-forward	
	open Homing	
	PLCopen Homing Description	
	PLCopen Homing Methods	
	AKD Capture Engine Configuration	
12.2.6 Pipe	Network Homing	

	12.2.6.1 Fast Homing Example with the Pipe Network Motion Engine Axis Pipe Block .	614
	12.2.7 PLCopen Registration	616
	12.2.7.1 Deciding Which Function Blocks to Use for Registration	. 616
	12.2.7.2 Single-Axis Registration	617
	12.2.7.3 Master/Slave Registration	617
	12.2.7.4 Registration Application Guide	618
	12.2.7.5 Fast Homing Example with the PLCopen Motion Engine	. 620
	12.2.8 Pipe Network Registration and Fast Homing	621
	12.2.8.1 Fast Homing Example with the Pipe Network Motion Engine Axis Pipe Block	621
	12.2.8.2 Registration Position Capture Example with Pipe Network Trigger Block	622
	12.2.9 How To Interpret a Timestamp	624
	12.2.10 Tuning Controller Performance for EtherCAT Communication Latency	624
	12.2.11 Error Management	625
	12.2.12 Restarting Motion with PipeNetwork	625
	12.2.13 Superimposed Axes with PLCopen	626
	12.2.14 Overtravel Conditions	627
	12.2.14.1 PLCopen Behavior	627
	12.2.14.2 Coordinated Motion Behavior	627
	12.2.14.3 PipeNetwork Behavior	627
	12.2.14.4 Configuring Generation 1 AKD Drives	627
	12.2.15 How To Create a Digitizing Axis	
	12.2.15.1 How To Create a Digitizing Axis When EtherCAT Devices Are Defined	629
	12.2.15.2 How To Create a Digitizing Axis When EtherCAT Devices Are Not Connected	l 630
	12.2.15.3 How To Create a Digitizing Axis When Using an Imported ENI File	
	12.2.15.4 How To Create a Digitizing Axis From Code	632
	12.2.16 Working With A Digitizing Axis in PLCopen	633
	12.2.16.1 Digitizing Axis – Encoders Without a Z-Channel	. 634
	12.2.17 Practical Applications - Camming	635
	12.2.17.1 Positioning an Axis Before Starting Camming	635
	12.2.17.2 Resuming Camming After an E-Stop	
	12.2.17.3 Cam On The Fly	
12	2.3 Motion Bus and Fieldbuses	. 637
	12.3.1 EtherCAT	
	12.3.2 Ethernet/IP	
	12.3.3 Modbus & TCP/IP	
	12.3.4 Profinet	
	12.3.5 Add Unsupported EtherCAT Device	
	12.3.5.1 How to modify the EtherCAT image in cyclic mode	
	12.3.5.2 How to configure EtherCAT device	
	12.3.5.3 How to map PLC variables	
	12.3.6 Set Up FSoE Master and an AKD2G with SafeMotion Monitor	
	12.3.6.1 Create Safety Application	
	12.3.6.2 KAS IDE Project Set Up	
	12.3.7 AKD2G Safety Parametrization Using FSoE with SCU-1-EC and PCMM/AKD PDMM	
	12.3.7.1 Requirements	
	12.3.7.2 How-To	
	12.3.8 Fieldbus Editor	. 643

12.3.8.1 PCMM/AKD PDMM Ethernet/IP Data Communications	643
12.3.8.2 Ethernet/IP IO Scanner (Client)	643
12.3.8.3 Ethernet/IP Adapter (Server)	
12.3.8.4 Ethernet/IP Tag Client	
12.3.8.5 PDMM/PCMM EDS file for EtherNet/IP	
12.3.8.6 Ethernet/IP FlexIO / PointIO	
12.3.8.7 Modbus Slave	651
12.3.8.8 Profinet IO RT Controller Configuration	
12.3.8.9 Profinet IO RT Device Configuration	
12.3.9 Timeout error	671
12.3.10 Timeout error	
12.3.11 Performance Guidelines	
12.3.11.1 EtherNet/IP	
12.4 Project Structure Guidelines	679
12.4.1 Introduction	
12.4.2 External Files	679
12.4.3 Application Software Structure - Definitions	
12.4.3.1 Modules to build up the Structure	
12.4.3.2 State and Function Definitions	
12.4.4 Application Software Structure - Implementation	
12.4.4.1 SFC children building up the software	
12.4.4.2 Variables for the Interface	
12.4.4.3 Main module implementation description	
12.4.4.4 States and Errors	
12.4.4.5 Functions linked to states	688
12.5 Project Templates	690
12.5.1 Pipe Network 1-Axis Template with FFLD only	
12.5.2 Pipe Network 1-Axis Template with ST only	
12.5.3 Pipe Network 2-Axes Template with SFC, ST, FFLD, and FBD	
12.5.3.1 PLC Programs	
12.5.4 Motion	
12.5.5 Control Panel	
12.5.5.1 HMI	
12.5.6 Pipe Network 2-Axes Template with ST only	
12.5.6.1 PLC Programs	
12.5.7 Motion	698
12.5.8 Control Panel	
12.5.8.1 HMI	
12.5.9 Pipe Network 2-Axes Template with FFLD only	701
12.5.9.1 PLC Programs	
12.5.10 Motion	701
12.5.11 Control Panel	702
12.5.11.1 HMI	702
12.5.12 PLCopen 1-Axis Template with FFLD	
12.5.13 PLCopen 1-Axis Template with ST	
12.5.14 PLCopen 2-Axes Template with SFC and FFLD	
12.5.14.1 PLC Programs	705

12.5.14.2 Motion	706
12.5.14.3 Control Panel	707
12.5.14.4 HMI	708
12.5.15 PLCopen 2-Axes Template with ST	710
12.5.15.1 PLC Programs	710
12.5.15.2 Motion	
12.5.15.3 Control Panel	711
12.5.15.4 HMI	
12.5.16 PLCopen 2-Axes Template with FFLD	714
12.5.16.1 PLC Programs	714
12.5.16.2 Motion	
12.5.16.3 Control Panel	715
12.5.16.4 HMI	716
12.5.17 Coordinated Motion 2-Axis Template	718
12.5.17.1 Programs	718
12.5.17.2 Motion	718
12.5.17.3 Control Panel	718
12.5.18 Coordinated Motion 3-Axis Template	719
12.5.18.1 PLC Programs	719
12.5.18.2 Motion	719
12.5.18.3 Control Panel	720
12.5.19 Coordinated Motion 3-Axis (3D) Template	721
12.5.19.1 PLC Programs	
12.5.19.2 Motion	721
12.5.19.3 Control Panel	722
13 PDMM and PCMM Errors and Alarms	. 723
13.1 PCMM and PDMM Errors	
13.2 PCMM and PDMM Alarms	726
13.3 CPU Overload (E23)	727
14 KAS Component Manuals	729
14.1 HMI	
14.1.1 AKI2G-CDA-MOD-05T-000, 5" LCD Display	
14.1.2 AKI2G-CDA-MOD-07T-000, 7" LCD Display	
14.1.3 AKI2G-CDB-MOD-07T-000, 7" LCD Display	
14.1.4 AKI2G-CDB-MOD-12T-000, 12" LCD Display	737
14.1.5 AKI-CDC-MOD-12T-000	739
14.1.6 AKI-CDC-MOD-15T-000	741
14.1.7 AKI-CDC-MOD-21T-000	743
14.1.8 HMI Accessories	745
14.2 Controllers	745
14.2.1 PCMM Specifications	745
14.2.2 AKD PDMM Specifications	
14.2.3 NVRAM	
14.2.3.1 How can I check the NVRAM space is enough to store my retain variables?	747
14.3 BBH FSoE Master	748
14.4 Kollmorgen Safety Solutions	749
14.5 Mounting and Wiring of I/O Terminals	

14.5.1 Instructions for ESD Protection	750
14.5.2 Installation on mounting rails	751
14.5.2.1 Assembly	751
14.5.2.2 Disassembly	751
14.5.2.3 Connections within a bus terminal block	752
14.5.2.4 PE power contact	752
14.5.3 Installation instructions for enhanced mechanical load capacity	753
14.5.3.1 Additional checks	753
14.5.3.2 Additional installation instructions	753
14.5.4 Connection	754
14.5.4.1 Connection system	754
14.5.4.2 Wiring	754
14.5.4.3 Shielding	755
14.5.5 Installation positions	756
14.5.5.1 Optimum installation position (standard)	756
14.5.5.2 Other installation positions	756
14.5.6 Positioning of passive Terminals	757
14.5.6.1 Examples for positioning of passive terminals (highlighted)	757
14.5.7 UL Notice	
14.5.7.1 Basic principles	
14.5.8 Continuative documentation about explosion protection	
14.5.9 Safety Instructions	
14.5.9.1 Delivery state	
14.5.9.2 Operator's obligation to exercise diligence	
14.5.9.3 Description of safety symbols	
14.5.10 Safety Operation	
14.5.10.1 Environmental Conditions	
14.5.10.2 Safety Instructions	
14.5.10.3 Transport / Storage	
14.5.10.4 Mechanical Installation	
14.5.11 Safety Terminal Reaction Times	
14.5.11.1 Typical Reaction Time	
14.5.11.2 Worst-Case Reaction Time	
14.5.12 Safety I/O Maintenance	
14.5.13 Interference-Free EtherCAT Terminals	
14.6 Remote Input/Output Terminals 14.6.1 AKT2G-AN-240-000	
14.6.2 AKT2G-AN-240-000	
14.6.3 AKT2G-AN-430-000	
14.6.4 AKT2G-AT-410-000	
14.6.5 AKT2G-AT-410-000	
14.6.6 AKT2G-DN-002-000	
14.6.6.1 Up/down counter 24 VDC, 100 kHz, 32 bit counter depth	
14.6.6.2 Technical Data	
14.6.6.3 LEDs and Connection	
14.6.6.4 Basic Function Principles	
14.6.6.5 Operating Mode Selection	

14.6.6.6 Objects for Commissioning	784
14.6.6.7 Profile-specific objects (0x6000-0xFFFF)	
14.6.7 AKT2G-DN-008-000	
14.6.8 AKT2G-DNH-008-000	
14.6.9 AKT2G-DT-008-000	
14.6.10 AKT2G-ECT-000-000	
14.6.11 AKT2G-EM-000-000	
14.6.12 AKT2G-ENC-180-000	
14.6.12.1 Incremental Encoder Interface AKT2G-ENC-180	
14.6.12.2 AKT2G-ENC-180 Technical Data	
14.6.12.3 AKT2G-ENC-180-000 LEDs and Pin Assignment	
14.6.12.4 Commissioning AKT2G-ENC-180-000	
14.6.12.5 Frequency	
14.6.12.6 Frequency mode A	
14.6.12.7 Measurement sequence	
14.6.12.8 Frequency mode B	
14.6.12.9 Period calculation	
14.6.12.10 Gate function	
14.6.12.11 Latch	
14.6.12.12 Reset	
14.6.12.13 Up/down counter	
14.6.12.14 Micro-increments	
14.6.12.15 Digital filter	
14.6.13 AKT2G-ENC-190-000	
14.6.13.1 Interface terminal for incremental encoder	
14.6.13.2 Technology	
14.6.13.3 Technical Data	
14.6.13.4 AKT2G-ENC-190-000 LEDs and Connection	816
14.6.13.5 Commissioning AKT2G-ENC-190-000	818
14.6.13.6 Restore object	821
14.6.13.7 Configuration data	821
14.6.13.8 Input data	822
14.6.13.9 Output data	822
14.6.13.10 Frequency	825
14.6.13.11 Frequency measurement	825
14.6.13.12 Measurement sequence	826
14.6.13.13 Period calculation	
14.6.13.14 Latch	826
14.6.13.15 Reset	
14.6.13.16 Up/down counter	
14.6.13.17 Overflow/underflow	
14.6.13.18 Open circuit detection	
14.6.13.19 Micro-increments	
14.6.13.20 Configuration Data	
14.6.13.21 Information / Diagnostic Data (Channel Specific)	
14.6.14 AKT2G-PSF-024-000	
14.6.14.1 Connections	

14.6.14.2 LEDs	836
14.6.15 AKT2G-SDI-004-000	837
14.6.15.1 Safety terminal with 4 digital fail-safe inputs	837
14.6.15.2 Intended use	
14.6.15.3 Technical Data	839
14.6.15.4 Safety Parameters	841
14.6.15.5 Characteristic curve of the inputs	841
14.6.15.6 Dimensions	
14.6.15.7 Block Diagram of the AKT2G-SDI-004	842
14.6.15.8 AKT2G-SDI-004 Pin Assignment	843
14.6.15.9 Signal Cables	844
14.6.15.10 Address Settings on Safety Terminals with 65535 Possible Addresses	846
14.6.15.11 Safety Terminal Address and Parameters	846
14.6.15.12 AKT2G-SDI-004 Diagostic LEDs	848
14.6.15.13 AKT2G-SDI-004 Diagostic Objects	850
14.6.15.14 Certificates	851
14.6.16 AKT2G-SDO-004-000	853
14.6.16.1 Safety terminal with 4 digital fail-safe outputs	853
14.6.16.2 Intended use	855
14.6.16.3 Technical Data	856
14.6.16.4 Safety Parameters	858
14.6.16.5 Dimensions	859
14.6.16.6 Block Diagram of the AKT2G-SDO-004	859
14.6.16.7 AKT2G-SDO-004 Pin Assignment	860
14.6.16.8 Signal Cables	861
14.6.16.9 Address Settings on Safety Terminals with 1023 Possible Addresses	862
14.6.16.10 Safety Terminal Address and Parameters	863
14.6.16.11 AKT2G-SDO-004 Diagostic LEDs	864
14.6.16.12 Possible Causes of Diagnostic Messages - AKT2G-SDO-004	865
14.6.16.13 Certificates	867
14.6.17 About AKT-SM-Lxx Stepper Slices	868
14.6.18 AKT-SM-L15-000 Stepper motor terminal, 24 V DC, 1.5 A	869
14.6.18.1 General Specification	869
14.6.18.2 Electrical and Mechanical Specification	869
14.6.19 AKT-SM-L50-000 Stepper motor terminal, 50 V DC, 5 A, with incremental	
encoder	
14.6.19.1 General Specification	
14.6.19.2 Electrical and Mechanical Specification	
14.7 Drives	
15 Troubleshooting	
15.1 FAQs	
15.1.1 Basic	
15.1.2 Setup	
15.1.3 Computer / Controller	
15.1.4 Performance	
15.1.5 EtherCAT	
15.1.6 Programming	
15.2 Compiler Errors	883

15.3 CPU Load Reduction Techniques	885
15.4 EtherCAT Diagnostics & Errors8	886
15.4.1 EtherCAT Diagnostics	886
15.4.2 EtherCAT Error Messages8	886
15.4.2.1 Other Messages Linked to EtherCAT8	888
15.4.3 EtherCAT Communication Diagnosis Steps	888
15.4.3.1 Diagnosing EtherCAT Communication Errors	888
15.4.3.2 Code Examples for Diagnosing EtherCAT Communication Errors	889
15.5 EtherCAT Coupler Error Handling And Diagnosis	891
15.5.1 EtherCAT Diagnostic LEDs8	391
15.5.2 EtherCAT LED Power Supply Diagnosis8	391
15.5.3 EtherCAT LED Off Power Supply Diagnosis8	
15.5.4 LEDs for EtherCAT State Machine/PLC Diagnosis8	391
15.5.5 LEDs for EtherCAT Connection Diagnosis8	392
15.5.6 LEDs for EtherCAT Data Diagnosis8	392
15.6 Troubleshooting FSoE Safety Parameters8	393
15.7 Troubleshooting KAS IDE and SafePLC28	395
15.8 Connect Remotely8	
16 Annexes: Lists of Manuals, Content, and Support Information	aa
16.1 List of How-Tos	
16.1.1 PLC Code How-Tos8	399 399
	399 399
16.1.1 PLC Code How-Tos8	399 399 399
16.1.1 PLC Code How-Tos	899 899 899 899
16.1.1 PLC Code How-Tos	399 399 399 399 399
16.1.1 PLC Code How-Tos	399 399 399 399 399 399 399 399
16.1.1 PLC Code How-Tos	399 399 399 399 399 399 399 399
16.1.1 PLC Code How-Tos816.1.2 EtherCAT Fieldbus How-Tos816.1.3 Advanced Motion How-Tos816.1.4 Run the Application How-Tos816.1.5 Hardware How-Tos816.2 List of KAS Manuals816.2.1 KAS IDE916.2.2 KVB9	 399 390 300 300 300
16.1.1 PLC Code How-Tos	 399 390 300 300 300
16.1.1 PLC Code How-Tos816.1.2 EtherCAT Fieldbus How-Tos816.1.3 Advanced Motion How-Tos816.1.4 Run the Application How-Tos816.1.5 Hardware How-Tos816.2 List of KAS Manuals816.2.1 KAS IDE916.2.2 KVB9	399 399 399 399 399 399 399 399 399 300 900
16.1.1 PLC Code How-Tos 8 16.1.2 EtherCAT Fieldbus How-Tos 8 16.1.3 Advanced Motion How-Tos 8 16.1.4 Run the Application How-Tos 8 16.1.5 Hardware How-Tos 8 16.2 List of KAS Manuals 8 16.2.1 KAS IDE 9 16.2.3 HMI 9	 399 399 399 399 399 399 399 399 399 300 000 001 001 001
16.1.1 PLC Code How-Tos 8 16.1.2 EtherCAT Fieldbus How-Tos 8 16.1.3 Advanced Motion How-Tos 8 16.1.4 Run the Application How-Tos 8 16.1.5 Hardware How-Tos 8 16.2 List of KAS Manuals 8 16.2.1 KAS IDE 9 16.2.3 HMI 9 16.2.4 I/O 9	 399 399 399 399 399 399 399 399 399 300 000 001 001 001 002
16.1.1 PLC Code How-Tos 8 16.1.2 EtherCAT Fieldbus How-Tos 8 16.1.3 Advanced Motion How-Tos 8 16.1.4 Run the Application How-Tos 8 16.1.5 Hardware How-Tos 8 16.2 List of KAS Manuals 8 16.2.1 KAS IDE 9 16.2.3 HMI 9 16.2.4 I/O 9 16.2.5 AKD2G, AKD, & S300/S700 9	 399 399 399 399 399 399 399 399 399 300 <
16.1.1 PLC Code How-Tos .8 16.1.2 EtherCAT Fieldbus How-Tos .8 16.1.3 Advanced Motion How-Tos .8 16.1.4 Run the Application How-Tos .8 16.1.5 Hardware How-Tos .8 16.1.6 List of KAS Manuals .8 16.2.1 KAS IDE .9 16.2.3 HMI .9 16.2.4 I/O .9 16.2.5 AKD2G, AKD, & S300/S700 .9 16.2.6 PDMM / PCMM .9	 399 399 399 399 399 399 399 399 399 300 001 001
16.1.1 PLC Code How-Tos 8 16.1.2 Ether CAT Fieldbus How-Tos 8 16.1.3 Advanced Motion How-Tos 8 16.1.4 Run the Application How-Tos 8 16.1.5 Hardware How-Tos 8 16.1.5 Hardware How-Tos 8 16.2 List of KAS Manuals 8 16.2.1 KAS IDE 9 16.2.2 KVB 9 16.2.3 HMI 9 16.2.4 I/O 9 16.2.5 AKD2G, AKD, & S300/S700 9 16.2.6 PDMM / PCMM 9 17 Acronyms & Initialisms 9	399 399 399 399 399 399 399 399 399 390 391 392 393 394 395 396 397 398 399 390 391 392 393 394 394 394 394

2 Document History

Edition	Software Version	Date	Description
0.2		September 2010	First release for KAS R1
А	2.5	May 2012	Initial release of KAS IDE and AKD PDMM
В	2.6	December 2012	EtherCAT devices backup & restore; PLCopen Registration; Superimposed Axes; Cam on the fly; Ethernet/IP; Profinet; Reset to factory settings; Save project on controller; Password protection
С	2.7	June 2013	Initial doc
D	2.7	October 2013	Coordinated Motion; EtherCAT device configuration; KVB 2, softscope; generic Modbus configuration; Dictionary improvements; Control panel changes
E	2.8	May 2014	Coordinated Motion (3D linear w/ blending & transition; nD w/o blending & transition), HTTP variable interface, UDP socket interface, EtherCAT MDP management
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.
G	2.10	January 2016	PCMM support, Code generation types: Normal and Optimized, MC_TouchProbe: Time or Position-based capture, , MCFB_ StepLimitSwitchFastInput, HTML5 help.
н	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 FFLD & FBD, F_Seek function, configurable circle center point tolerance.
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
К	3.0	January 2018	AKD 1.17, AKD2G integration, MC_CreatePLCAxis, MLPNAxisCreate
L	3.01	December 2018	Multiple installed versions, updated UDFBs, Backup Import/Export on webserver, UI improvements, Softscope enhancements, security enhancements
М	3.02	June 2019	Map a digitizing axis, AKD2G, multiple instances of KAS IDE, ElasticSearch, reorder programs
N	3.03	December 2019	FSoE; T-Bot / H-Bot / Gantry robotics: Axes Groups, MC_KIN_REF structure, MC_SetKinTra; AKD2G Backup & Restore, MDP support
Ρ	3.04	June 2020	TCP/IP functions, eitadapter, file management functions, SafePLC support, WorkBench improvements, support AKT2G slices, DriveParamStrRead, multi-select PDO objects for mapping
R	3.05	December 2020	Integration of BBH SCU-1-EC with KAS IDE and BBH SafePLC2 software; Integration of E-bus AKT2G I/O devices; WorkBench connection improvements; MLAxisReadVel

KAS IDE User Guide | 2 Document History

---- / ----

3 Preface

Welcome to the Kollmorgen Automation Suite help and documentation! This book covers the use and programming of the KAS IDE, the AKD PDMM and PCMM, information about KAS components, and much more. This chapter explains how to use the online help provided with Kollmorgen Automation Suite™.

3.1 System Requirements for KAS IDE and KAS Simulator

KAS IDE and KAS Simulator are compatible with any PC having the minimum following hardware:

Element	Description
Operating System	Microsoft® Windows® 7 SP1 (32 or 64-bit), Microsoft® Windows® 10 (32 or 64-bit). For optimal performance, please be sure your operating system is fully updated with the latest patches.
Processor type	Intel® Pentium® M or equivalent processor at 1.5 GHz or greater.
Memory	1 GB RAM (for 32-bit) or 2 GB RAM (for 64-bit) or greater (which is recommended for complex applications).
Storage	16 GB (for 32-bit) or 20 GB (for 64-bit) of free space on hard disk.
Display	WXGA+ (1440 x 900) or higher-resolution monitor with 24-bit color. See Note #1 below.
Connectivity	1 Ethernet port, at either 100Mbits/s or 1Gbits/s. See Note #2 below.
Web Browser	A modern web browser is required to access the web server and online help. We recommend Microsoft Edge $\stackrel{\textbf{c}}{\textbf{c}}$, Google Chrome $\stackrel{\textbf{o}}{\textbf{o}}$, Mozilla FireFox 🕹, or Internet Explorer (IE9 or later, see Note #3) 🔁.

NOTE

- 1. Better results are achieved with OpenGL and 3D cards.
- 2. A 100Mb network is required in order to allow the IDE to Runtime communication to work in all conditions. The AKDWorkBench AutoTuner and Scope both require 100Mb of bandwidth to function properly.
- 3. IE9 should be considered a minimum. Later versions of the browser are more compliant with web standards and afford better performance and compatibility.

★ TIP

See the topic Connect Remotely for information about the ports used by the KAS IDE which may need to be opened to support connecting from an external network.

Table 1-1: Minimum System Requirements for the KAS IDE

3.2 Learning Kollmorgen Automation Suite

To learn Kollmorgen Automation Suite, you can either:

- Navigate this online help and choose chapters depending on your experience, or
- Read the printed materials

3.2.1 Where to Start

The KAS documentation includes information for readers from a variety of backgrounds. To get the most out of the documentation, we recommend that you start by reading the chapters that are most relevant to you. Within each chapter, read through the topics in sequence.

3.2.1.1 Beginner

- Find basic information about KAS in Introducing Kollmorgen Automation Suite
- If you are not familiar with the concepts behind KAS, read the Understanding KAS
- An overview of the KAS IDE User Interface is in Describing KAS Graphical User Interface
- To get information on how to run and debug the project, read Launch KAS Simulator and Testing and Debugging the Project

3.2.1.2 Advanced User

- In order to design and create a project, refer to the Getting Started
- Go to Tools if you need explanations about the tools used by the KAS IDE
- For in-depth information, refer to Advanced Topics

3.2.1.3 Online

- KDN: The Kollmorgen Developer Network ("KDN") has many articles, resources, and a user community where you may learn, ask questions, and contribute to our community. Visit KDN at https://www.kollmorgen.com/en-us/developer-network/
- YouTube: Kollmorgen has a channel dedicated to KAS. Check out the KAS Online Learning Collection.

3.2.2 Read KAS Manuals

If you prefer to read printed material, the following manuals (in PDF format) are available under the C:\Program Files\Kollmorgen\Kollmorgen Automation Suite\Help folder

Document Title	PDF	Description
Release Notes	PDF	The KAS version 3.05 Release Notes contain fixed limitations, known limitations, workarounds, and information on all hardware and software components that have been updated, changed or added in this release.
Getting Started	PDF	Covers the main steps to get your _{KAS} system up and running. <i>What does it contain?</i>
		 HW Installation (Connection and Wiring): Wiring & hardware details, connectors, system diagrams HW Configuration: Basic configuration and settings needed to start the HW components (HMI + Industrial PC + Fieldbus + I/O) SW Installation: KAS software setup

Document Title	PDF	Description
30 Minutes to Motion	PDF	Covers the main topics to help you start 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. What does it contain?
		 Key Features Explore the Workspace Build a motion project Almost every task that you perform in KAS falls under one of the following basic steps (which may not always be completed in the following order):
		 Start Projects - Create a project from scratch, or modify an existing project. Add Components - Add elements to build your project necessary to control the motion part of your system. Build Output - Select a device and generate the application that you will deliver to users. Run Output - Make the output accessible to your end-users.
IDE User Manual	PDF	Contains the content to help you with KAS IDE, except the topics included in the Reference Manuals.
Reference Manual - PLC Library		Contains Technical References on PLC Programming Languages and Library.
Reference Manual - Motion Library	PDF	Contains Technical References on Motion Library for Pipe Network and PLCopen.

Table 1-2: List of KAS Guides in PDF Format

★ TIP

The KAS IDE allows you to include references to external files such as the PDF files listed above. For more details, refer to Use the Reference Folder.

Additionally, you can add in the PDF your own comments, tips and tricks, provided that you have Adobe Acrobat®.

4 Introducing Kollmorgen Automation Suite

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

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

As can be seen, KAS can cover a wide variety of applications:

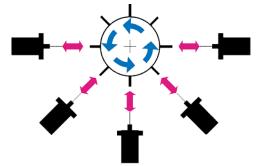


Figure 2-1: Synchronized Feeder

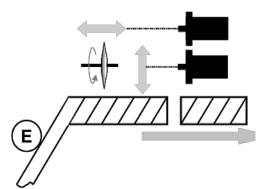


Figure 2-2: Spring Winding

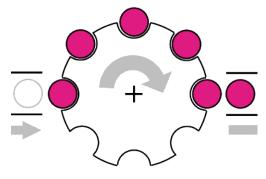


Figure 2-3: Synchronizer

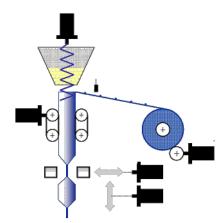


Figure 2-4: Form Fill Seal

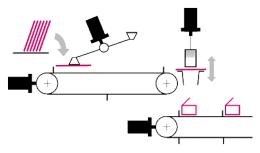
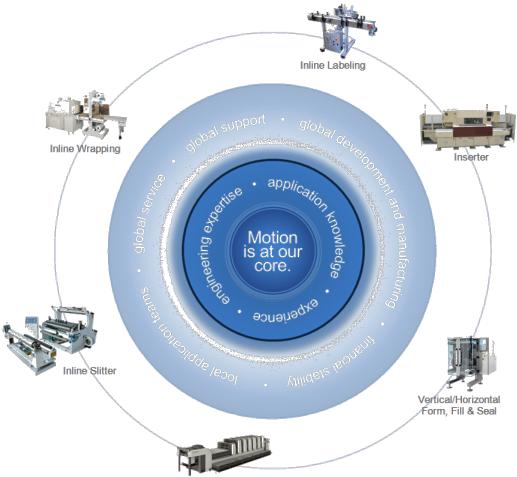


Figure 2-5: Carton Erector

4.1 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 can be encapsulated as follows:



Flexographic Printing

Kollmorgen Automation Suite (also known as 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 integrated hardware products such as Programmable Automation Controllers, Programmable Drives, AKD drive family, award winning components like the 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 goal by integrating in a **coherent**, **intuitive**, **flexible** way the three main functionalities of a machine:

- 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 two different software components:

- KAS IDE the Integrated Development Environment allowing the 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

4.1.1 Integrated Development Environment



KAS comes with a powerful Integrated Development Environment (IDE) (commonly named **KAS IDE**) which 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 (AKD PDMM, PCMM).

With the KAS IDE, system engineers can:

- Create new application projects using predefined or custom application templates
- Define the machine hardware architecture (motion bus, fieldbus, controllers (Controllers, PCMM), drives and motors) as well as 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 five IEC 61131-3 programming languages (ST, IL, FFLD, FBD and SFC), the IEC 61131-3 standard library and KASFunction Block libraries dedicated to motion, communication and monitoring
- Centrally manage all IEC 61131-3 variables with KASvariable dictionary
- Create and organize your own libraries of functions and function blocks
- Easily set up **HMI panels** by means of graphical objects that are 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 - also called 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 (for instance 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 **softscope** tool
- Access the full online documentation

4.1.2 KAS Runtime



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

The 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:

- Ranging 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
- Running on a single or multiple controllers
- Communicating via Ethernet, EtherCAT, Profinet, EtherNet/IP, or Modbus
- Using the high-performance Pipe Network or the standard PLCopen function blocks
- Controlling 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 the two different contexts:

- With a controller implementation (PCMM)
- With a master drive implementation (AKD PDMM)

See Different Implementations for more details.

4.1.3 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 for more information.

4.2 Looking at Kollmorgen Automation Suite

4.2.1 Physical View

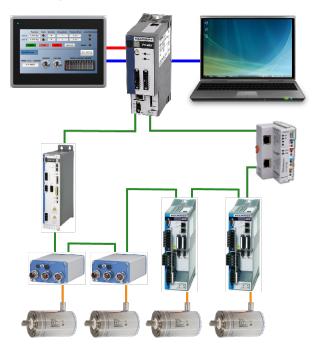


Figure 2-6: Example of Automation System

Key of Communication Types

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

4.2.2 Logical View

An automation system usually needs an organized hierarchy of controller systems to function and usually including the following items:

Item	Call out#	Description
HMI	1	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 a AKI panel.
Communication	2	HMI is linked to the middle layer via a non time critical communication system (e.g. Modbus TCP protocol on Ethernet).
PLC	3	Programmable Logic Controllers is a digital computer used for automation of industrial processes, such as 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.
Motion Engine	4	There are two Motion Engines available: Pipe Network and PLCopen. The Motion Engine implements different motion algorithms and functions to create, access and delete pipes, pipe blocks and axes. It also provides a set of Functions and Function Blocks that IEC 61131-3 applications can use to control the behavior of these algorithms.
Fieldbus	5	The fieldbus is the way to connect instruments in a plant design by linking the PLC to the external systems.
I/O	6	Input/Output refers to the communication between your automation system, and the outside world.
Drive	1	A Drive is an electronic device that provides power to a motor or servo.
Motor	8	At the bottom of the control chain is the motor which actually does the work.

Color Legend	Description
Blue Cell	Belongs to KAS
Green Cell	Kollmorgen products
White Cell	Third parties

SCADA	ERP	PLC	НМІ 1			
Communication (2						
Soft PLC 3						
I/O E	ngine	Motion Engine				
Operating System						
Fieldbus 6						
I/O (
Drives						
Motors (8						
Feedback						
Gear boxes						
Mechanical Elements						

4.2.3 Architectural View

The diagram shows an architectural view of what KAS is from a hardware and software perspective, or realtime and not-realtime 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 (seen in blue), EtherCAT (seen in green), Modbus (seen in red), or several other protocols.

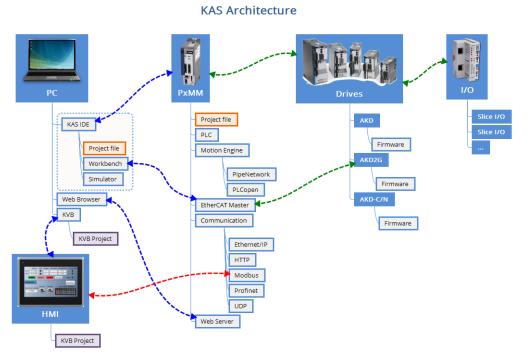


Figure 2-7: Architectural view of KAS components.

The non real-time part is composed of:

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

The realtime components, including that which run the KAS Runtime engine, consists of:

Item	Description
IEC 61131-3 virtual machine	Responsible for managing an IEC 61131-3 application with its programs and variables

Item	Description
Motion manager	Manages motion engines, axis objects and motion bus drivers. The KAS Runtime comes with two motion engines: Pipe Network and PLCopen. 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 also 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 custom function blocks

Interface between the realtime and non-realtime systems is done via shared memory buffers.

The 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.

4.3 KAS Breakdown

Domains	Concept (Technology)	Task (Tools)	Reference
HMMI		Using Kollmorgen Visualization Builder Add an HMI Device	HMI
Controllers – PCMM & DUARDA	PCMM and AKD PDMM (Programmable Drive Multi-axis Master)	Add the Controller Configure the Controller	Controllers
PLC RachineState > (157(P1)2 357(P1)2 457(P1)2 457(P1)2 457(P1)2 457(P1)2 457(P1)2 457(P1)2 557.	IEC 61131-3	ST editor IL editor FBD editor FFLD editor SFC editor Variable dictionary Softscope	STLanguage IL Language FBD Language FFLD Language SFC Language
Motion Engines	Motion Concept Pipe Network PLCopen 	Design Pipe Network Pipe Network Editor Create Cam Profile Cam Profile Editor Softscope	

Domains	Concept (Technology)	Task (Tools)	Reference
Fieldbus	EtherCAT	Configure EtherCAT Motion Bus	Motion Bus and Fieldbuses Cables
I/O Terminals	EtherCAT	Add and Configure I/O Slices Map Input and Output to Variables	Remote Input/Output Terminals
Drives	AKD AKD2G AKD-C AKD-N S300 S700	Add and Configure Drive AKD Drive Configuration Download AKD PDMM Drive Firmware AKD/AKD2G Firmware Update	Drives
Safety	Safety over EtherCAT	 Working with SafePLC2 Projects How To Set Up a Variable to Monitor Safety Data AKD2G Safety Parametrization Using FSoE with SCU-1-EC and PCMM/AKD PDMM Set Up FSoE Master and an AKD2G with SafeMotion Monitor 	BBH FSoE Master Drives AKT2G-SDI- 004-000 AKT2G-SDO- 004-000
Motors	Kollmorgen Servomotor		AKM2G AKM Cartridge DDR Housed DDR KBM TBM Direct Drive Linear
Mechanical			Linear Actuators Gearheads

Table 2-1: KAS - Technologies and Tools

4.3.1 Human-Machine Interface



Figure 2-8: Hardware to Display the Human-Machine Interface

PLCs interact 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 displays are available as well as graphics 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" on page 121). Variables defined in the HMI to describe the interface (see Map Variables to HMI) are passed to the AKD PDMM/PCMM this way. 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. As this data has a rather low priority and is interpreted by human feedback, it should never be noticed by the user.

NOTE

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

4.3.2 Controllers



Figure 2-9: High, medium and low voltage AKD PDMMs and the PCMM.

Kollmorgen Automation Suite controllers consist of the AKD PDMM (Programmable Drive Multi-Axis Master) and PCMM (Programmable Controller Multi-Axis Master).

4.3.2.1 AKD PDMM and PCMM Hardware

The AKD PDMM comprises of three printed circuit boards (PCB) while the PCMM comprises of one.

- Power board (AKD PDMM only)
- AKD control card (AKD PDMM only)
- AKD PDMM and PCMM option card, available in two variants:
 - NXP/Freescale QorIQ P1011 processor (800MHz)
 - NXP/Freescale QorIQ P2010 processor (1.2GHz)

РСММ

AKD PDMM

- NXP/Freescale QorIQ P2010 processor (1.2GHz)
- NXP/Freescale QorIQ P2020 processor (Dual Core 1.2GHz)

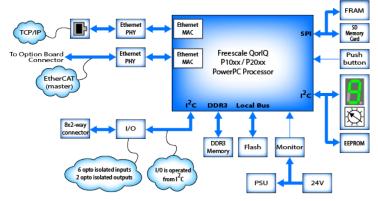


Figure 2-10: AKD PDMM and PCMM card

NOTE

42

Please note that any reference to AKD PDMM or PCMM refers to both the 800MHz and the singleand dual-core 1.2GHz variants, unless otherwise noted.

4.3.2.2 PDMM or PCMM Rotary Switch



The rotary switch on the AKD PDMM or PCMM can be set on a position from 0 to 9.

Switch Position	Description
Position 0	The drive tries to get an IP address from a DHCP server. If the DHCP fails, then the AKD PDMM or PCMM uses AutoIP to get a usable IP address.
Position 1	The default custom static IP address, 192.168.0.101 or a custom IP address.
Position 2-9	The AKD PDMM or PCMM is pre-configured with static IP addresses ranging from 192.168.0.102 (Position 2) to 192.168.0.109 (Position 9).

★ TIP

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

NOTE

The AKD PDMM or PCMM will not set (or show) an IP address without an attached network cable.

4.3.2.3 PDMM & PCMM Web Server

These controllers contain a web server that allows you to perform the following operations:

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



See Using the KAS Web Server for more information.

4.3.2.4 Controller Performance

Kollmorgen offers several controllers with varying performance. This section compares the relative performance to help you decide what is best for your needs.

4.3.2.4.1 Performance Graphs

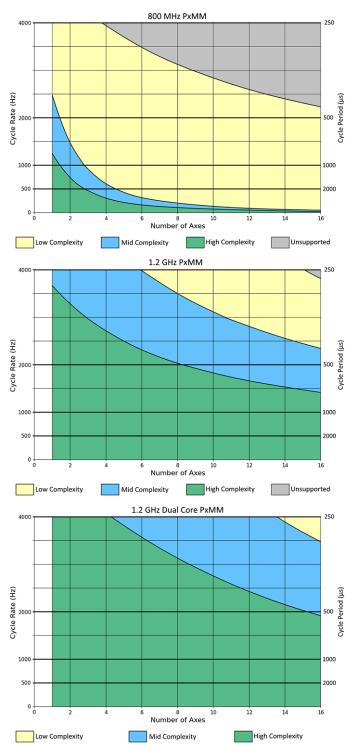


Figure 2-11: Graphs comparing the performance of 800 MHz single-core, 1.2 GHz single-core, and 1.2 GHz dual-core controllers.

- A *high-complexity* application is characterized by complex setups such as most axes being slave axes, and cammed or geared to a master axis, frequent moves, frequent reading and setting of external I/O points, and complex PLC logic.
- A *low-complexity* application is characterized by point-to-point and velocity moves, occasional reading of external I/O points, and simple PLC logic.
- A *middle-complexity* application lies in between and is usually characterized by most axes being point-to-point or velocity moves and some axes being cammed or geared to a master; PLC logic is split so that parts of a complex logic run at given times.

Generally speaking, Pipe Network applications use less of the controller's CPU. So for the purposes of using the above graphs, a Pipe Network application will tend to act like a slightly lower-complexity application while a PLCopen application will tend to act like a slightly higher-complexity application.

Should I use a single 800MHz, single-core 1.2 GHz, or dual-core 1.2 GHz controller?

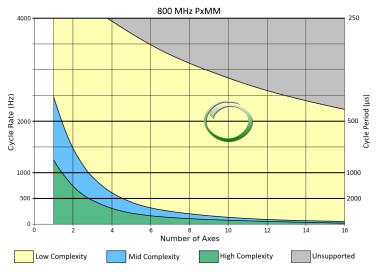
To determine which controller model to use for initial development of an application, the axis count and cycle rate of the application can be used to determine what type of application can be run on different controllers. For example, suppose you are planning a middle-performance application for 6 axes that runs at a 2 kHz (500us) cycle rate. Looking first at the 800 MHz PxMM, you can see that it can only handle a low-complexity application with these requirements. However, the single-core 1.2 GHz PxMM can handle a high-complexity application with these requirements. Since only a middle-completxity application is needed, the single-core 1.2 GHz PxMM should be sufficient for this application. If in doubt, start with a higher-performance controller model and after optimizing the application code, determine if a lower performance model will meet the present and future needs for the application.

NOTE

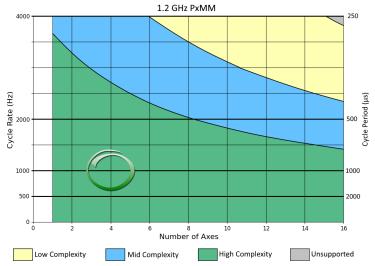
The graphs above are for guidance in initial controller selection for development. During development, the performance of an application should be measured to ensure that the selected controller will meet the performance requirements. Measurements can be done with KAS's digital oscilloscope using the trace times traces, the GetCtrlPerf function block, and by observing the CPU usage on the Hardware Status tab of the KAS controller's web page interface.

Examples

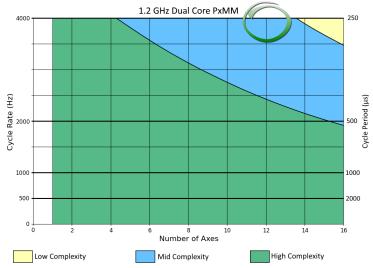
• A 10-axis low-complexity application running at 2 kHz (500us) can be handled by an 800 MHz PxMM.



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



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



How much improvement will I get in moving from a single-core 1.2 GHz PxMM to a dual-core PxMM?

The dual-core PxMM improves performance by running PLC code and motion code at the same time. Typical performance gain varies from 45% to 80% 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 (see Practical Application: Using Trace Time To Measure CPU Load), are roughly the same. If one of these two values dominate the execution time, then the performance gain will be on the smaller side.

Can I downgrade the controller used to run my application?

- If you are using a *dual-core* PxMM and the peak CPU usage (see Hardware Status) is under 40%, then the single-core 1.2 GHz PxMM model will likely meet your needs.
- If you are using a *dual-core* PxMM and the peak CPU usage is under 25%, then the 800 MHz PxMM model will likely meet your needs.
- If you are using a *single-core* PxMM and the peak CPU usage is under 50%, then the 800 MHz PxMM model will likely meet your needs.

KDN has an article that has a tool to help you pick a controller which matches your project needs.

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, such a 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 to be able to handle rigid jitter and timing demands including high data throughput and low latency.

4.3.3.2.1 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.

4.3.3.2.2 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, and outputs are the signals or data sent from it. Automation systems built with KAS are interrupt-driven. Typical interrupt uses include the following: system timers, disks I/O, power-off signals, and exceptions handling.



Figure 2-12: I/O Modules

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



Figure 2-13: Standard I/O Couplers and Slices

4.3.5 Drives

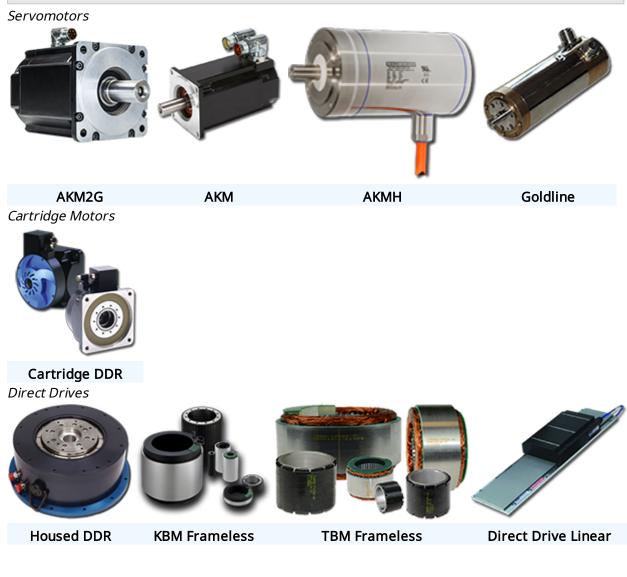


See Drives in Hardware Devices chapter for more details.

4.3.6 Kollmorgen Motors

NOTE

The following images are all hyperlinked to their respective product pages on Kollmorgen.com.



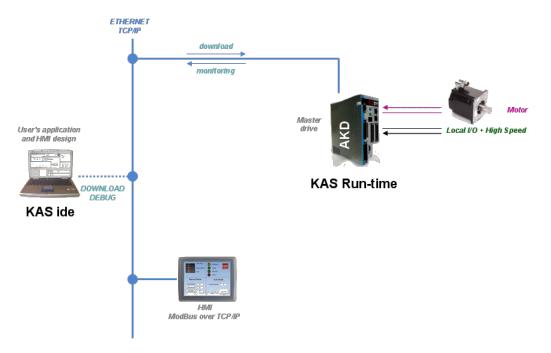
4.4 Different Implementations

KAS supports the following architectures:

- Single-Axis with PDMM (→ p. 49)
- Multi-Axis with PDMM or PCMM (→ p. 50)

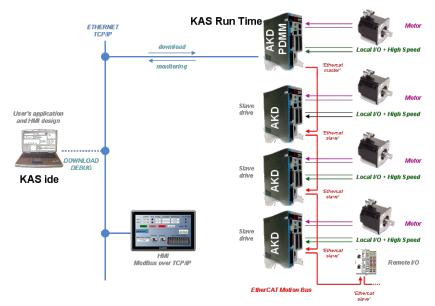
4.4.1 Single-Axis with PDMM

The scalable system architecture begins with an integrated programmable controller and drive package to control a single axis.



4.4.2 Multi-Axis with PDMM or PCMM

The AKD PDMM (integrated programmable controller and drive package) or PCMM (programmable controller) can be scaled to control additional drives w/ motors and I/O devices via EtherCAT. This configuration can manage up to 32 axes.



5 Describing KAS Graphical User Interface

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

NOTE

- For in-depth discussion of interface elements, refer to KAS IDE Reference
- For KAS Simulator GUI, refer to Using the KAS Simulator
- For AKD drive GUI View, refer to AKD Drive

5.1 Windows and Panels Overview

5.1.1 Main Window

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

E Kollmorgen Automation Suite Integrated Dev	elopment Environment - ReadAux.kas - [EtherCAT]
🛋 File Edit View Tools Window Help : 💽 🏫 💾 🗮 🕵 🖛 🥐 🗱 ն	
Project Explorer	Controller:Motion:PipeNetwork EtherCAT Main Control Panel
Name Type ^	Devices Cycle Time XML Configuration File
System Sinope Controller Drive Ba PLC PLC Main SFC	XML Configuration File 3a
Subprograms Subprograms Subprograms Subprograms Profiles	XML Options √Wite a unique ID to each EtherCAT device to prevent wiring mistakes. XML File Import / Export
Control Panel Panel PDMM Onboard I/O Control Panel PDMM Onboard I/O Control Panel Pomoto Panel Pomoto Panel Pomoto Panel	Import Configuration File Use imported configuration file Export Configuration File
Watch Window 🗗 🗶 Int	formation and Logs & X
• - + + 3b .	ocal log Controller log Find and Replace Compiler Output
Name	nd what: Match complete word Find
La	ook in: All documents
< → Watch 1	Device Location Text
Local: ReadAux8	Drives inactive 4 Not Connected

Figure 3-1: KAS IDEMain Window

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 capability to hide, enlarge or overlap windows in order to optimize visibility.

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

The KAS IDE main window contains the following items:

- Menu bar (see call out ⁽¹⁾)
- Toolbar ⁽²⁾ A toolbar is a little bar with icons which is usually located under the menu bar of a window.
- Workspace ⁽³⁾ which contains:
 - A specific area dedicated to displaying the workspace children windows (3a)
 - Several toolboxes ^(3b) A toolbox is a child window that provides you with some functions to perform specific tasks.
- Status bar at the bottom 4 displaying the current state of the target

5.1.1.1 About toolboxes

The available toolboxes include:

- 5.1.2 Project Explorer
- 5.1.3 Libraries
- 5.1.4 Dictionary
- 5.1.5 Information and Logs

★ TIP

You can hide/show each toolbox and toolbar directly from the contextual menus in any title bar (i.e. menu, toolbar or toolboxes).

5.1.2 Project Explorer

The Project Explorer toolbox is a window that displays machine application information in a treestructure representation. This window contains all the following items used to design, implement, test, and document the application.

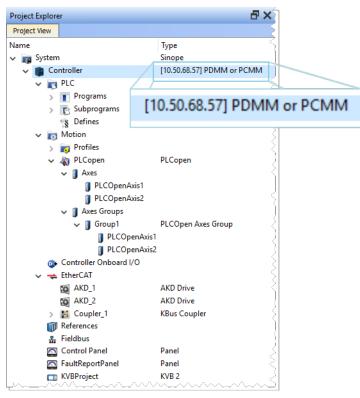


Figure 3-2: Project Explorer with Controller type and IP address.

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.
PLC (IEC 61131-3)	Programs that control the systemUser-defined Functions and Function Blocks
Motion	 Pipe Networks or PLCopen Axis objects Cam profiles

🗙 TIP

You can navigate in the project-tree by entering the item's initial letter, or by means of the arrow keys.

A project is made of several items that are:

- System
- Controller
 - PLC
 - Programs
 - Subprograms
 - Defines

- Motion
 - Profiles
 - Pipe Network or PLCopen
- Controller Onboard I/O
- EtherCAT
 - AKD Drive
 - AKT (K-Bus) I/O Coupler
 - Device
 Module
 - IVIO
- References
- Fieldbus
- Control Panel
- KVB Project
- SafePLC2 Project
- HMI Device
 - KVB Project

NOTE

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

5.1.2.1 System

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

Command	Description
Add HMI Device	Add a new HMI device with a KVB panel. For mode details see HMI Device below.

Table 3-1: System Node - Contextual Menu

5.1.2.2 Controller

This item displays the controller's IP address and controller hardware type. It is also used to Access the WebServer From the IDE. The webserver functionality may be used directly within the IDE. For more information on the webserver see Using the KAS Web Server.

NOTE

Please note that the IP address is shown as 127.0.0.1 if the system is in simulation mode.

Command	Description
Add Control Panel	Add a new contol panel to the controller. For more details see Control Panel.
Import Control Panel	Import a pre-configured control panel for use in the project. See Import a Control Panel for more information.
Add KVB Project	Add a new KVB panel which is embedded into the contoller. For more details see KVB Project.
Import KVB Project	Import a compressed ("zipped") KVB project, which may be created in _{KAS} or KVB. The system will validate the compressed KVB project and add the panel.
Add SafePLC2 Project	Add a node to the tree that allows you to access a SafePLC2 project directly from the KAS IDE. See Working with SafePLC2 Projects for more information.
Import SafePLC2	Select a . SPL2 file to import.
Project	There may only be a single instance of a SafePLC2 project in the KAS IDE. If none exists, a SafePLC2 Project will be added to the end of the Project tree. If there is a project you will be prompted to overwrite the project or cancel the import. Note that the import cannot occur if SafePLC2 is running.
Add Fieldbus	Add a node to access the Fieldbus Editor. See for more information.
Access Web Server	This command opens the web server interface in the GUI. See Access the WebServer From the IDE and Using the KAS Web Server for more information.

Command	Description
Properties	Open a dialog box to configure the controller. See <u>Configure the</u> Controller for a full description of this dialog box.
	Controller Properties
	Controller Configuration
	IP Address 10.50.68.30 Controller Type PDMM or PCMM
	Project Version
	Download project source to the controller Enable PLC variable remote access
	Modbus Configuration
	Other Modbus devices
	Figure 3-3: 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. Selecting an item in the list and pressing the 'Delete' key will clear the entry from the list. When you click to choose the simulation mode, this address is disabled.
	() IMPORTANT Do not use leading zeroes when entering an IP address. The connection to the controller will automatically fail if leading zeroes are used. For example, use 10.2.3.40 instead of 010.002.003.040 .
	• The Controller type can be a AKD PDMM or PCMM.
	 Different Implementations 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 will allow users to read/write variables using an HTTP connection. "Modbus Configuration" sets what the Modbus will connect to and the Cycle Time specifies the update rate.

Table 3-2: Controller Node - Contextual Menu

A controller is composed of a PLC item, a Motion item, control panels, an EtherCAT Motion Bus and some References. These items are described in the following sections.

5.1.2.3 PLC

This item contains all the PLC (Virtual Machine) part of the controller. The following items can be present in this item:

- Program items
- Subprogram items
- Some "Defines"

Command	Description
Libraries	Import new libraries

Table 3-3: PLC Node - Contextual Menu

5.1.2.4 Programs

Command	Description
New Program	Add new program items (SFC,ST,FBD, IL or FFLD)
Cycle	Configure the cycle of the virtual machine For mode details on Cycle, see "Define the PLC Cycle" on page 205
Import	Import a saved program

Table 3-4: Program Node - Contextual Menu

🗙 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.

Command	Description
Add Child SFC	Add a child program to this program. Note that this is reserved for the first SFC program only.
Import Child SFC	Import a saved SFC program to the current program. Note that this is reserved for the first SFC program only.
	How to import all children from one project to another?
	 Export each program one at a time from the existing project Save the program (specify a location and a name, avoiding spaces in the file name) Close the project Open the project to be updated Import each saved program in the project tree Rename the program if needed
	Please note that 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 for more details.

Table 3-5: Program Item - Contextual Menu

🗙 TIP

You can double-click to open the program in the workspace.

5.1.2.5 Subprograms

Command	Description
New Function (Subprogram)	Add a new subprogram item (ST,FBD, IL or FFLD)
New UDFB	Add a new UDFB item (ST,FBD, IL or FFLD)
Import	Import a saved program

Table 3-6: Subprogram Node - Contextual Menu

You can create your own functions as well as functional blocks that are called UDFBs (User-Defined Functional Blocks). For each of them, you can use the following commands:

Command	Description
Export	Save the selected subprogram onto your file server
Rename	Rename the selected subprogram
Create Unlocked Copy	Duplicate the selected, locked subprogram. The duplicate will not be locked.
Delete	Delete the selected subprogram
In/Out Parameters	Open the Program Properties dialog box to Declare Functions or Function Blocks. This item is disabled for locked UDFBs.

Table 3-7: Subprogram Item - Contextual Menu

5.1.2.6 Defines

This item contains all the global definitions in the scope of the corresponding device.

★ TIP

```
You can double-click a Define item to show these global definitions. A file of internal defines can be found in [the installation directory of the KAS IDE]\Astrolabe\Bin\HwDef\lib.eqv.
```

Use the Defines List

5.1.2.7 Motion

The motion item contains the motion-specific items (i.e. the Profiles and PipeNetwork items).

Command	Description
Motion Engines	Choose the motion engine for your application between PLCopen and PipeNetwork

5.1.2.8 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) For mode details, see "Adding Cam Profiles" on page 199

Command	Description
Import	Import already existing cam profiles to your project
Show compiled code	Show the code corresponding to the selected cam profile

Table 3-8: Profiles Node - Contextual Menu

Right-clicking on a cam profile provides additional commands.

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

For more information on cam profiles see Adding Cam Profiles and Cam Profile Editor.

5.1.2.9 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. You will be presented with a dialog box to locate the pre-saved file. The Pipe Network Editor will be 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**. Additionally, profiles assigned to Cam blocks are cleared at this time.

1. Double-click on **EtherCAT** in the Project View to open the <u>EtherCAT Devices tab</u> tab so you can reassign the axes.

2. Double-click on any Cam blocks, and set the Profile_Nam	e parameter.
CAM: CAM	
General Parameters	
Profile Name Output Modulo Position 360.0	
ОК Сапсе!	

5.1.2.10 PLCopen

Command	Description
New Axis	Add a new axis to your project For mode details, Create PLCopen Axis
Show compiled code	Show the code corresponding to the PLCopen

Table 3-9: PLCopen Node - Contextual Menu

5.1.2.10.1 Axes

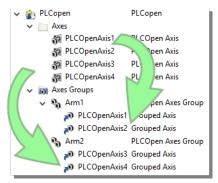
Each axis is listed here, whether it is found via a scan of the EtherCAT network, or added manually. To add an axis manually, right-click and select **New Axis**. For each PLCopen axis you can use the following commands:

Command	Description
Properties	Open a dialog box to configure the PLCopen axis data
Delete	Delete the selected axis

Table 3-10: Axis Item - Contextual Menu

5.1.2.10.2 Axes Groups

Axes Groups are collections of related axes defined by the data type AXIS_GROUP_REF, typically used by coordinated motion and function blocks such as MC_SetKinTra. Right-click on this section to create a new Axes Group. Groups are populated by dragging an existing axis from the above section into the group.



The following 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.

5.1.2.11 Control Panel

This item holds the Control Panel item used to provide a basic interface between you and the virtual machine.



- For more details, Design the Control Panel with the Internal Control Panel Editor
- For a more advanced tool to build HMI, KVB Project

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.

Table 3-11: HMI Control Panel Node - Contextual Menu

5.1.2.12 Controller Onboard I/O

Command	Description
Properties	Open the Properties dialog box to configure the local I/O for the PDMM or PCMM controller.
	Configure Controller Onboard I/O

Table 3-12: Controller Onboard I/O Item - Contextual Menu

5.1.2.13 EtherCAT

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

Command	Description
Add Device	Add a Kollmorgen drive, power supply, or coupler, pr a third-party EtherCAT device to the EtherCAT node in the Project view.
Scan Devices	TheKAS Runtime sends EtherCAT messages to discover the devices present in the network EtherCAT Devices tab
Enable/Disable Online Configuration Mode	Toggles Online Configuration Mode on and off. See Online Configuration Mode for more information.
Properties	Open the Properties dialog box. Configure EtherCAT Motion Bus

Table 3-13: EtherCAT Node - Contextual Menu

Add & Configure Third Party Devices

5.1.2.14 AKD2G Drive

You can double-click an AKD2G to set its parameters. Configure the AKD2G/AKD Drive

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.

Table 3-14: AKD Drive Item - Contextual Menu

5.1.2.15 AKD Drive

You can double-click an AKD to set its parameters. Configure the AKD2G/AKD Drive

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.

Table 3-15: AKD Drive Item - Contextual Menu

5.1.2.16 AKD-C Central Power Supply

You can double-click an AKD-C to set its parameters. Configure the AKD2G/AKD Drive

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.

Table 3-16: AKD-C Drive Item - Contextual Menu

5.1.2.17 AKD-N Drive

You can double-click an AKD-N to set its parameters. Configure the AKD2G/AKD Drive

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.

Table 3-17: AKD-N Drive Item - Contextual Menu

5.1.2.18 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.

Table 3-18: AKT2G I/O Coupler Node - Contextual Menu

Note that all those commands are disabled when the controller is running.

5.1.2.19 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.

Table 3-19: Standard I/O Coupler Node - Contextual Menu

Note that all those commands are disabled when the controller is running.

See EtherCAT Coupler Error Handling And Diagnosis in the Troubleshooting section for information about diagnosing the coupler LEDs.

5.1.2.20 I/O Slice

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

Command	Description
Rename	Rename the selected slice
Delete	Delete the selected slice
Properties	Open the Properties dialog box to configure the I/O slice
	Map Input and Output to Variables

Table 3-20: I/O Slice - Contextual Menu

5.1.2.21 Device

Double-clicking a Device accesses its EtherCAT device configuration views.

Command	Description
Rename	Rename the selected device
Delete	Delete the selected device
Add Module	Add a module to an MDP device. Add Modules to Third Party EtherCAT Devices.
Properties	Access the EtherCAT device's configuration views

Table 3-21: Device - Contextual Menu

5.1.2.21.1 Module

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

5.1.2.22 References

This item allows you to **insert references** into your project. Each reference is a user-defined reference that links any kind of deliverable to your project (for more details, refer to Use the Reference Folder)

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

Table 3-22: Reference Node - Contextual Menu

5.1.2.23 Fieldbus

This item holds the Fieldbus Editor to configure the Ethernet/IP or Profinet fieldbuses. For mode details,

5.1.2.24 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. For mode details, Add an HMI Device Note that this command is disabled when a KVB panel already exists
Import KVB Project	Import a compressed ("zipped") KVB project, which may be created in KAS or KVB. The system will validate the compressed KVB project and add the panel.
Rename	Rename the selected HMI device
Delete	Delete the selected HMI device

Table 3-23: HMI Device Node - Contextual Menu

5.1.2.25 KVB Project

Command	Description
Open KVB	Opens the associated KVB project.
Rename	Rename the selected KVB panel
Delete	Delete the selected KVB panel
Export	Save a copy of the panel in a compressed (.zip) file.

Table 3-24: KVB Panel Node - Contextual Menu

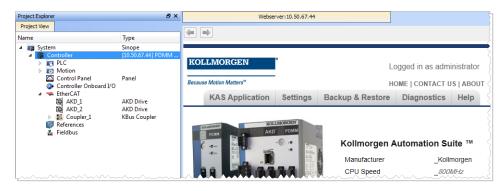
5.1.2.26 SafePLC2 Project

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 Working with SafePLC2 Projects for more information.		
Rename	Rename the SafePLC2 project		

Table 3-25: SafePLC2 Project Node - Contextual Menu

5.1.2.26.1 Access the WebServer From the IDE

Double-clicking **Controller** will both expand/collapse the Controller's components as well as open the web server and automatically log into the *administrator* account. For more information on using the webserver see Using the KAS Web Server.



The web server can also be accessed by right-clicking the Controller node and selecting **Access webserver**.

🗙 TIP

IDE users are considered administrators and therefore are automatically logged into the webserver. You may change the password but there is no logout function.

By default the localhost (127.0.0.1) will be opened. To set the IP address of the controller, right click and select Properties. Enter the proper **Address** and **Controller type** then click **OK**. The page is automatically refreshed.

If an invalid or wrong IP address is entered, the following error will be displayed.

	Webserver:255.255.255
1	This program cannot display the webpage
	Most likely causes:
	You are not connected to the Internet.
	The website is encountering problems.
	 There might be a typing error in the address.
	What you can try:
	• Check your Internet connection. Try visiting another website to make sure you are connected.
	Retype the address.
	 Go back to the previous page.
	⊘ More information

5.1.2.26.1.1 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**.

	Name	Туре	1	
	 ▲ System ▲ Controller ▲ PLC ▶ Programs ⊗ Subprograms ⊗ Defines ▲ Motion ➡ Profiles 	Sinope [10.50.67.44] PDMM or PCMM		
2.	Control Panel PDMM Onboard I/O	Rename Delete Export		
	Export to			х
	CO V KAS	Projects 🕨 👻 🍫	Search KAS Projects	م
	File name: Control Panel.gra Save as type: *.gra	3		•
	Save as type			
	Browse Folders		Save Cancel	

3. Click OK to confirm the export.

See also: Import a Control Panel

5.1.2.26.1.2 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.



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

Choose a HMI panel to import		×
◯◯ ▽ 🌗 « My Documents → KAS Projects → 👻 🍫	Search KAS Projects	٩
Organize 🔻 New folder	** -	0
CD Navigator KAS PDMM WebServer PDF Comparisons PDMM PDMM PDMM Iterations Registration documentation Control Panel.gra		
File name: Control Panel.gra 👻	*.gra Open ▼ Can	▼ cel

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

5.1.3 Libraries

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

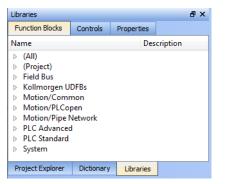


Figure 3-4: Libraries Toolbox

5.1.3.1 Function Blocks

This tab displays all the available libraries shown in a tree-structure representation and gathered by categories. You can expand a library to access all its functions. A short description of each function is also 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, refer to the following libraries description:

- PLC Standard
- PLC Advanced
- Motion/Pipe Network
- Motion/PLCopen
- Field Bus
- System
- Kollmorgen UDFBs

★ 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 will be seen in the editor.

5.1.3.2 Controls

This tab displays all the controls available for the HMI design. For more details, refer to the Graphic Objects description.

5.1.3.3 Properties

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

More information about setting the properties of an HMI widget can be found in Graphic Objects Properties.

5.1.4 Dictionary

The Dictionary toolbox is used to show all the identifiers (variables, data types, sub-routines, etc.) defined within the project. There are three tabs within the Dictionary, the Variables tab, the Enum Tab, and the Bit Fields Tab.

5.1.4.1 Variables tab

The Variables tab is used to show all the variables defined within the project. All the variable details are displayed in order to show the variable types, dimensions, attributes, etc.

Dictionary							₽×
Controller:PLC		_	📃 Track Select	ion			
Variables Enums Bit Fields							
🝸 Name	Туре	Dim. Attrib.	Init value	User Gro	НМІ	Tag	Description
🕀 📭 Global variables	(1)						
E 💾 Retain variables	\mathbf{U}						
CntIn	UINT	IN			\boxtimes		
CntOut	UINT	OUT					
Actions							
Main							
🔁 📖 EtherCATCode							
RewStructure1	(2)						
RewStructure							
🖽 🛃 PNCode							
🖽 🛃 ProfilesCode							
🖽 🛃 PLCopenCode							
<	Ш						>

Figure 3-5: Dictionary Toolbox

The list of variables is split into two parts:

- All the Variables at the top (1)
- All the Structures at the bottom 2

NOTE

For more information about the procedure to create an instance of a structure, see "Call Functions or Function Blocks" on page 308

🗙 TIP

70

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

About the Dictionary's contextual menu.

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

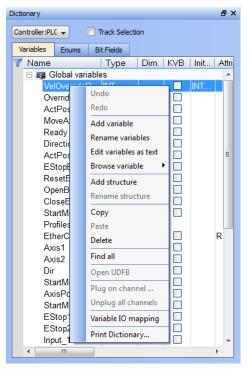


Figure 3-6: Dictionary Contextual Menu

This menu allows you to perform the following 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 (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 (for more details, see "Define Parameters and Private Variables" on page 307)			
Rename variables	This function can either replace a section of matching variables or append text to the variables' names. ^{See} Rename Variables			
Edit variables as text	This function will open a text editing dialog, allowing you to edit the variables found in the group as text using IEC 61131-3 syntax. ^{See} Editing Variables as Text for more information.			
Browse variable	This function allows you to browse all instances of a variable. See Browse Variable Tab for more information.			
Add Structure	Used to have a new complex type. A structure named NewStructure is created and variables can be dragged into it (for more details, see "Complex Structures" on page 302)			
Rename Structure	Rename the selected structure			
Сору	Copy a variable			
Paste	Paste the copied variable to the selected level			
Delete	Delete the selected variable. A deletion can also be performed by pressing the Delete key on the keyboard			
Find all	This function will find all instances of the specified variable and open the results in the Find and Replace tab.			

Command	Description					
Open UDFB	Open the selected UDFB instance (for more details, see "Monitoring UDFBs" on page 231)					
Plug On Channel 	Plug the selected variable on a channel. This command opens a dialog used to configure the variable plug operation.					
	the _{type} of varia	ble is eligible NT, UDINT, L	e for th JLINT,	ne softsco BYTE, WC	pe (i.e. BOOl)RD, DWORD	ted and running, and if ., INT, SINT, DINT, 9, LWORD, TIME and
Unplug All Channels	Unplug all plugge	ed probes fr	om th	e softscop	be	
Variable I/O mapping	Connect a variab	le to an I/O.				
Print Dictionary 	Print all the varia columns display t					ed by programs. The , and Attributes
	MasterDeltaPos MachineSpeed Axis1Status MachineState bMasterAbs bMasterRel bEstop	LREAL LREAL LREAL DINT DINT DOL BOOL BOOL BOOL BOOL ProfilesCode EtherCATCode PNCode LREAL LREAL LREAL LREAL LREAL DINT	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	FALSE FALSE FALSE	R	

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 your project variables will be displayed. This is the default setting.
- **Selected**: The variables in the Dictionary are filtered to display only those that are relevant to the PLC item currently selected in the project tree. Along with the Global, retains and variables related to the selected program or UDFB, structure definitions will be displayed. The dictionary content will change accordingly if another PLC program is selected in the project tree.

About the two editing modes for editing a variable.

There are two available modes when editing a variable in the Dictionary:

• Cell: only the selected cell is active

Dictionary								Β×
Controller:PLC	~	Track Selection						
14 Name	Туре	Dim. Attrib.	Init value	User Gro	HMI	Tag	Description	
🗆 💽 INCR								^
CntIn	UINT	IN			\boxtimes			
CntOut	UINT	OUT						
Inst_UntitledFBD	AnUFBD							

• **Row**: all the row is active

Dictionary							8
Controller:PLC	*	Track Selection					
14 Name	Туре	Dim. Attrib.	Init value	User Gro HM	II Tag	Description	
🗆 💽 INCR							^
Cntln	UINT	IN		٥	3		
CntOut	UINT	OUT					
Inst UntitledFBD	AnUFBD			C]		

Press Spacebar to toggle the selection mode from cell to row (Table Keyboard Shortcuts).

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 which allows you to modify the table's appearance. These settings will persist until you change them. Please note that the Visible box for the Value field cannot be changed as the Value column is automatically shown in debug and online modes but hidden in edit mode.

Columns	Visible	Width Filter	ОК
Name	×	100	
Туре	~	80	Default
Dim.	~	40	
КУВ	~	75	Cancel
Init value	~	40	
Attrib.	~	60	Move Up
User Group	~	50	INDAE Ob
Tag	~	50	Move Dow
Description	~	500	
Properties		100	
Value		100	Reset Filte

- You can sort the list of variables in the table as follows:
- Ensure you are in **cell** editing mode (press the **Spacebar** to toggle from one mode to the other)
- Click the header of the column you want to use as the key sort order

How do I modify parameters of a variable?

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

Mode	Description
One Parameter	Assuming you are in the cell edition mode, double-click on the parameter
All the parameters are at the same time	Assuming you are in the row edition mode, double-click in any parameter to open the dialog box for variable configuration as shown below.
	For more details on parameters, see "Variables" on page 74-

NOTE

It is not possible to modify a variable when the KAS IDE is connected to the controller.

5.1.4.1.1 Variables

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

Variable	Description
Global variables	List all global variables that are used and accessible throughout the entire program
Retain Variables	List all variables that are to be retained when the system is powered down
Program variables	List the variables related to your specific selected program

For each variable, the Dictionary toolbox allows you to set the following parameters:

Field	Description
Name	The variable name
Value	All the variables in the Dictionary are animated with real-time values 1
	Note that this column is only displayed when your application is running
	For more details, _{see} "Variable Monitoring" on page 231
Туре	The variable type (which can also be UDFB or complex structure)
Dim.	To declare an <mark>array</mark> , you can specify dimension(s) for an internal variable
Attrib.	The variable attributes (Read Only, External, IN, OUT) as defined:
	 Read Only: a variable set as Read Only is a constant (it cannot be modified in your PLC code, but it can be forced manually). Read Only variables can be mapped to Outputs but not to Inputs. This is because Inputs can change state and a Read Only variable would not be able to 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
Init value	The variable initial value when you start your application (see more details here)
User Group	The variable user group (used for sorting variables)
HMI	Select variables to be used in HMI (see Map Variables to HMI)

¹To 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.

Field	Description
Тад	The variable tag is a short comment, that can be displayed together with the variable name in graphical editors.
	Edit the variable parameters
	SetOverrideBusy1
	Name: SetOverrideBusy1 OK
	Type: BOOL Cancel Cancel
	Init value:
	Tag: SetOverride FB
	Description: Indicates the setoverride FB is executing
	External E Syb.
	Add the variable to your FBD program
	ST/IL ● FBD ● FFLD
	Resize the rectangle to make the Tag and Description visible
	ST/IL ● FBD Image: state st
Description	The variable description is a long comment text that describes the variable
Syb.	reserved

5.1.4.1.2 Structures

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

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
	Project Explorer Project View
	Name Type
	Sinope Controller Con
	Control Panel Panel EtherCAT Coupler_1 KBus Coupler
	Dictionary
	Controller:PLC Track Selection
	If Name Type Dim. Attrib. Init value Image: Straight of the s
	AKD_1 INT Read Only INT#0 Coupler_1 INT Read Only INT#0
	• You can use them in your PLC programs with a drag- and-drop operation 3
	MyF8D*
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

5.1.4.1.3 Variable editor

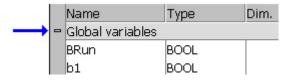
Variables are declared in the Dictionary of the KAS IDE main window.

The variable editor is a table that enables you to declare all variables of the application. Variables in the editor are sorted by groups:

- 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).

Please refer to the description of variables 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 can be used to expand or collapse the group:



See how to:

- Create New Variables
- Use the Variable Table List
- Define Structures
- Set Bookmarks

5.1.4.1.3.1 Create new variables

Press 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. You can rename a new variable or change its attribute by using the Variable Editor.

🗙 TIP

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

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

- adding an "IN" (input) parameter
- adding an "OUT" (output) parameter
- adding a private variable

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

5.1.4.1.4 Variable Table List

There are two available modes when editing a variable in the Dictionary:

• Cell: only the selected cell is active

Dictionary							8,
Controller:PLC	*	Track Selection					
1A Name	Туре	Dim. Attrib.	Init value	User Gro HMI	Tag	Description	
🗆 💽 INCR							^
CntIn	UINT	IN					
CntOut	UINT	OUT					
Inst_UntitledFBD	AnUFBD						

• **Row**: all the row is active

Dictionary								Ð
Controller:PLC	*	Track Selection						
14 Name	Туре	Dim. Attrib.	Init value	User Gro	HMI	Tag	Description	
🗉 🕞 INCR								^
Cntln	UINT	IN			\times			
CntOut	UINT	OUT						
Inst UntitledFBD	AnUFBD							

Press Spacebar to toggle the selection mode from cell to row (Table Keyboard Shortcuts).

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 the ENTER key to open the dialog box.

At any time you can drag with the mouse the column separators in the main grid header for resizing columns.

Press the following keys for browsing groups of variables:

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 Tables manipulation, see also Windows Standard Conventions

5.1.4.1.5 Sort variables

At any moment you can sort variables of a group according to their name, type or dimension. To do this, you simply need to:

- 1. Move the cursor to the header of the group
- 2. Click on the name of the column you want to sort

The KAS IDE always keeps the original order of declared variables, to allow safe online change. Each time you insert a new variable or expand/collapse a group, the original sorting is re-applied.

5.1.4.1.6 Define structures

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

For more details of the full procedure, refer to 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 you enter variables 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.

5.1.4.1.7 Name a variable

To change the name of the variable, do as follows:

- 1. In the Name column of the table, select the cell you want 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. The variable name cannot be a reserved keyword of the programming languages and cannot 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 userdefined Function Block.

The name of a variable 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 variable name. Naming is case-insensitive. Two names with different cases are considered as the same.

5.1.4.1.7.1 Naming Physical I/Os

Each I/O channel has a predefined symbol that reflects its physical location. This symbol begins with "**%I**" for an input and "**%Q**" for an output, followed by a letter identifying the physical size of the data. Refer to the description of variables for more details.

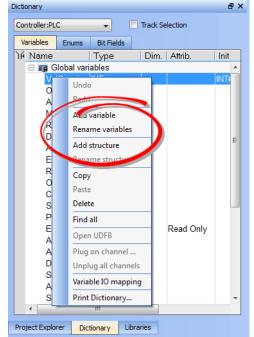
You cannot change the "%..." name of an I/O variable. This name is directly allocated according to the I/O devices defined in the I/O device list. But you can give an alias (a readable name) to each I/O channel. In that case, either the "%" name or the alias can be used in programs. The alias must fit to the same rules as a variable name.

When an alias is defined for a variable, both "%..." name and alias are displayed in the "name" column of the grid.

5.1.4.1.8 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.

1. Right-click on any variable in the Dictionary and select **Rename Variables**.



- 2. Enter the phrase to be replaced in What.
- 3. Enter the new phrase in **Replace**.
- 4. Set the replacement method using the **Where** menu.
- 5. Click OK to make the changes shown in the **Results** frame.
- 6. The **Replace in files** dialog box opens if there are matches within programs. Select which programs to modify.

🔳 Rename Variat	oles			x
What:				
Replace:				
Where:	Prefix			-
Match Case				
Results		 	 	
				~
		ОК	Cancel	

Item	What it means
What	Enter the variable name/text which is to be replaced
Replace	New text which will replace current variable name text

Item	What it mea	ns			
Where	How the replace will occurs				
	Prefix	The search/replace starts from the first character in the variable name.			
	Suffix	The "Replace" text will be appended to the "What" text, completely replacing any matches. Rename Variables What: VarStatus Where: Suffix Where: Suffix WarStatus -> VarStatusMotor Match Case Results VarStatus -> MachineLogic/VarStatusMoto Match Case Results VarStatus -> MachineLogic/VarStatusMoto Please be aware that this search method completely replaces the search string. Use this method with caution. Rename Variables What: us Us us What: us What: us </th			
	Everywhere	Searches the entire variable name.			
Match Case	The search wi	ill be case sensitive			
Results	The search re	sults and proposed changes are displayed.			

When matches exist within programs the **Replace in Files** dialog box displays the matches and lets you choose which programs to modify. Once **OK** is clicked, the programs will be modified in the order

they appear in the list.

Replace in files	3	×
Find what: Wheel Replace with: Motor Replace in files:		OK Cancel
 ✓ ^Automation ✓ Industrial ✓ MachineLogic ✓ STPrg 		

5.1.4.1.9 Initial Value of a Variable

A variable can have an initial value. The value must be a valid constant expression that fits to the data type 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.

You can change the initial value of a variable by using the Variable Editor.

5.1.4.1.10 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.

To change the tag or description of a variable, enable the modification mode to Row and move the cursor to the corresponding cell. Then press ENTER to enter the new text.

SetOverrideBu	sy1	×
Name:	SetOverrideBusy1	OK
Туре:	BOOL	Cancel
	🗖 Read Only Dim.:	
Init value:		
Tag:	SetOverride FB	
Description:	Indicates the setoverride FB is executing	
	☐ External □ Syb.	

5.1.4.1.11 Editing variables as text using IEC 61131-3 syntax

Using IEC61131-3 syntax, variables are declared within structured blocks. Each blocks begins with "VAR", "VAR_INPUT", "VAR_OUTPUT" or "VAR_EXTERNAL" keyword and ending with "END_VAR" keyword (with no semicolon after). Below is the meaning of each keyword:

Keyword	Meaning
VAR	Memory variables. Can be global, local or retain 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.
VAR_EXTERNAL	External variables. Can be global or local depending on the edited group

5.1.4.1.11.1 Basic syntax for declaring a variable:

To declare a variable, simply enter its symbol, followed by ":" and its data type. If the data type is STRING, it must be followed the maximum length between parentheses. Example:

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

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

```
CONSTANT VarName : DataType;
```

To declare an array, the data type must be preceeded by "ARRAY [dimensions] OF". There are at most 3 dimensions, separated by commas. Each dimension is specified as "0 .. MaxBound". Below are examples:

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

Additionally, 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. Examples:

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

5.1.4.1.11.2 Additional information and description texts:

As a variable may have additional properties and comment texts in the KAS IDE, we use special directives entered as IEC comments AFTER the declaration of the variable, to specify additional info. The following directives are available:

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

You can also use "//" single line comments to enter the directives:

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

5.1.4.2 Enum Tab

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.

Controller:PL	.C 👻	Track Selection
Variables	Enums	Bit Fields
Туре		Value
Season_et		Winter,Spring,Summer,Fall
DayOfWeek_et		Mon,Tue,Wed,Thu,Fri,Sat,Sun
TimeOfDay_et		Morning,Afternoon,Evening,Night
	/	

5.1.4.2.1 Adding Enums

An enum may be created in one of two 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.

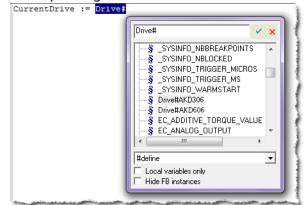
Controller - Tri	Dictionary	
Variables Enums	Controller 👻	Track Selection
E Drive	Variables	Enums
التصريب والمستعلى والمحم	Туре	Value
	Drive	AKD306,AKD606
	معتمون محمد	

NOTE

- An enum must contain two or more values. If only one value is defined, it will be not be considered and the previous value will be retained.
- An enum should not contain special characters such as #, @, etc.. If a special character is defined in the value field, it will be not be considered and the previous value will be retained.

5.1.4.2.2 To Use 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 your program.

CurrentDrive := Drive#AKD306

5.1.4.2.3 Declaring Enums

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.

CurrentDriv	e	23
This symbo	I does not exist. Do you want to:	
C Renam	e the variable	
 Declare 	e a new variable	
Type:	Drive	–
Where:	UntitledST	-
Where:	UntitledST	

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

UntitledST
 TravelSpeedInd...
 LREAL
 CurrentDrive
 Drive

5.1.4.3 Bit Fields Tab

A bit field packs multiple pieces of data together in one variable. Each field represents one piece of data. Each piece of data should have no dependency upon other fields. Bit Fields are used to define custom variable types and values.

5.1.4.3.1 Adding Bit Fields

A Bit Field may be created in one of three ways:

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

This creates a default Bit Field labelled as "BitFieldn" with a Value type "INT".

• The Bit Field type **Value** may be modified by double-clicking on value, allowing a selection from a list.



Once the Type is defined you may populate the bits.

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

IMPORTANT

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

NOTE

86

Bit Fields are supported over MODBUS. Below is a variable with a custom Type defined in the Bit Fields list which is flagged for KVB. It will be exported to MODBUS.

Variables En	ums Bit Fiel	lds					
16 Name		Туре 🔺	Dim.	Attrib.	Init val	Usey K	VB Ta
🗆 🧊 Globa	al variables						
Motor_	State	AxisState					
bMaste	erRel	BOOL			FALSE		
bMaste	erAbs	BOOL			FALSE		
bLedSt	tatus	BOOL	[03]				
bESto	р	BOOL			FALSE		
Machir	neState	DINT			0		
And in the second	and the second second	man and in the same	- and a	والمعاد والتعادي	And the second division of	and the second second	and the second second

Dictionary			5 ×
Controller 👻)	Track Selection	1
Variables	Enums	Bit Fields	
Туре	Va	alue	
🗄 BitField	11 IN	Г	
- 0			
- 1			
- 2			
- 3			
- 4			
- 5			
6			
- 7			
8			
- 9			
- 10			
- 11			
- 12			
- 13			
- 14			
- 15			

5.1.4.3.2 Using Bit Fields

Following is an example of setting up and testing a Bit Fields entry.

1. Define the Bit Field Type and Values.

Here we have defined a bit called **AxisState** with four values.

ctionary Controller:Pl		Track Select	ion
Variables	Enums	Bit Fields	
Туре		Value	
⊟ AxisStat	ie II	νT	
- 0 .		oq	
-1 F		Running	
- 2	5	Stopped	
- 3		Irror	
- 4			
		-	والمحكمين والمستحلين والمستحد

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

Here we have defined a variable called **Motor_State**. We selected the **AxisState** type from the dropdown menu.

Create Variable			×
Name: Type:	Motor_State AxisState		OK Cancel
1990.	E Read Only	Dim.:	
Init value:			
User Group:		•	
Tag:			
Description:			
	🗆 External		
	🗆 Syb.		

3. To use the bit enter VariableName.BitName in any editor.

Here we are using the Structured Text editor and typed <code>"motor_state."</code> When the dot was typed the selector opened, showing us the available bits for the variable.

Control Panel	UntitledST*
	notor_State.
	الاسالاني الطلائبين الكافلي ووعادهما التعلى والاسالاه

5.1.5 Information and Logs

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 different tabs that provide:

- Log messages (Local or Controller) including Log Messages Settings
- A system search function
- A list of breakpoints
- A state report on the program compiler

5.1.5.1 Log Messages

★ TIP

Log messages are an important source of information when you are troubleshooting with KAS IDE. When reporting an issue to Support, copy/paste the logs in your report.

nforma	tion and Logs						8>
Local lo	og Controller log Find an	d Replace	Compiler	Output		\sim	
ilter	1 * 3				~	Config	uration
	Time 🗸	Source	Level			\bigcirc	Messa 🛆
2 🕄	16-Mar-10 12:17:25 PM (384)	IDE	ERROR	Can not connect to addre		Disable	
з 🖸	16-Mar-10 12:17:25 PM (385)	IDE	ERROR	Execute command withou		Disable	
4 🕄	16-Mar-10 12:17:25 PM (385)	IDE	ERROR	Could not get SinopeRTE		Clear	
5 🖸	16-Mar-10 12:17:25 PM (388)	IDE	ERROR	Execute command withou		Сору	
6 🚺	16-Mar-10 12:17:25 PM (388)	EtherCAT	INFO	Could not get ECatDiscov		Save As	
7 🔥	16-Mar-10 12:17:59 PM (422)	IDE	WARN	No devices mapped to ph			~
1						Select All	

Figure 3-7: Log Messages

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 $\binom{2}{}$), only messages that are recorded and that match the filter $\binom{1}{1}$ are displayed.

★ TIP

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

Every log message in the table widget 3 has the following 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.

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

Table 3-26: Log Messages - List of Fields

The table contains a contextual menu (see call out 4) with the following 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
Сору	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

Table 3-27: Log Messages - List of Buttons

5.1.5.2 Log Messages Settings

The KAS IDE manages all messages according to the two following gates:

- Configuration settings define what is recorded in the database
- Filtering defines which messages are displayed in the table widget

5.1.5.2.1 Configuration Settings

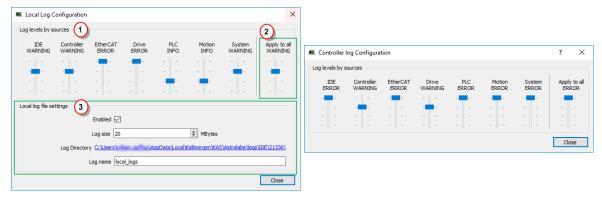


Figure 3-8: Configuration of the Local and Controller log messages

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 What are the sources of the log messages?, What are the log message levels?, and How do I choose the appropriate level for a log?.
2	You can set or reset all the sliders with the same level value
3	Logs can be recorded on the local machine as circular files.
	Note that on the controller, the recording of the logs is enabled by default. For more details, see About Local Log File Settings.

What are the **sources** of the log messages?

Source	Apply to
IDE	Win32 applications: the KAS IDE and the KAS Runtime Server (also called the KAS Runtime Front-end)
Controller	For the KAS Runtime items: Drivers, IOEngine, SinopEngine
EtherCAT	For all kinds of EtherCAT items: Motion bus, I/Os
Drive	Messages from AKD drive
PLC	For application engineers to create custom log within the PLC programs (similar to printf)
Motion	Messages coming from the Motion engines: PLCopen, Pipe network or VM
System	For common API and libraries. Also includes messages issued from the operating system.

What are the log message **levels**?

Level	Icon	Description
CRITICAL	÷	Application crashes or becomes unstable. Data is corrupted. At that point, the application behavior can be unpredictable.
ERROR	8	The application does not behave as expected but the processes remain stable.
WARNING	<u> </u>	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	1	Information status of the current process. You can ignore this log.
DEBUG	ø	Any information logged for development purpose. You can ignore this log.

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

About 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

You can import the log files into Microsoft Excel using drag-and-drop.

Field	Description
Enabled Log size	The Log File Settings has to be enabled to record all the logs. 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 will store 20 files. When
	a new file is created, an old one is deleted, maintaining the 20 file threshold. In the example below, the size on disk is limited to 100 KBytes (see call out 1).
	CircularLogFilePMO - Tue Jan 25 15h27m24 2011 CircularLogFilePMO - Tue Jan 25 15h27m32 2011 CircularLogFilePMO - Tue Jan 25 15h27m32 2011 CircularLogFilePMO - Tue Jan 25 15h27m32 2011 CircularLogFilePMO - Tue Jan 25 15h27m42 2011 CircularLogFilePMO - Tue Jan 25 15h28m42
	Image: CircularLogFilePMO - Tue Jan 25 15h29m09 2011 Size on disk: 100 KB (102,400 bytes) Image: CircularLogFilePMO - Tue Jan 25 15h29m43 2011 Size on disk: 100 KB (102,400 bytes) Image: CircularLogFilePMO - Tue Jan 25 15h30m27 2011 Attributes: Read-only Image: CircularLogFilePMO - Tue Jan 25 15h31m27 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h31m32 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h31m32 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h31m32 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h31m32 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h31m45 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h31m45 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h31m45 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h31m45 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h31m45 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h31m45 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h31m45 2011 Hidden Image: CircularLogFilePMO - Tue Jan 25 15h32m50 2011 Hidden
	CircularLogFilePMO - Tue Jan 25 15h32m51 2011 CircularLogFilePMO - Tue Jan 25 15h45m43 2011 CircularLogFilePMO - Tue Jan 25 15h46m20 2011 Note that the number of rows can vary for each file, depending on what is in the backlog when KAS creates the log files.

Field	Description
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. For more information see Where are the log files stored?
Log name	You can define the filename prefix to be used on the local machine.(on the controller, the filename prefix is: controller logs).
	The suffix to create the complete filename contains a timestamp with the following 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.

How do I 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. In other words, drag the level control slider **Up to reduce** the verbosity, **Down to increase** it. When the configuration leads to lower verbosity, the treatment during the filtering is quicker.

For 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). In other words, 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).

Where are the log files stored?

• 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

• The AKD PDMM or PCMM logs are accessed via the web server page by browsing to:

KAS Application > Log Data.

★ TIP

AKD PDMM/PCMM generated logs may be configured through the webpage. See PDMM and PCMM Log Files for more information on those controllers' logs.

		KA	AS Applicati	on	Settings	Dia	agnostics	He	elp
KOLLMORGEN		Kolmongen	Automation S	uite	KAS	Applicatio	on		
KA	S		ERFORMA N & PLC E		Versio	n of KAS Ap	р <i>Р</i>	N_4axis_HC:	:296
	197	"Because Mot			Status	of KAS App	_s	opped	
	Ť	Kollmorgan Automatic	in Suite High Performance Mot ind PLC software into a powerf	ion and PLC Engine	Start/S	itop	St	art Cold St	tart Stop
		to deliver superior per any of the five IEC 6113	formance for your automation 11 languages and combine the sen or the high-performance P	applications. Run m with a Motion	Option	s		Auto-start	
-							A	oply	
	0						A	рру	
	0						A	орту	
A	kis	Log	Configuratio	n	Log Dat	a	User		
A: og Config		Log	Configuratio	n	Log Dat	a			
			Configuratio EtherCAT	n Drive	Log Dat PLC	a Motion			
	uration						User	Data	
og Config	uration IDE	Controller	EtherCAT	Drive	PLC	Motion	User I System	Data	
og Config ebug	IDE	Controller	EtherCAT	Drive	PLC ©	Motion	User System	Data All ©	
og Config ebug fo	IDE O	Controller	EtherCAT	Drive ©	PLC ©	Motion ©	User	Data All ©	

NOTE

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

5.1.5.2.2 Filtering

You can narrow the list of recorded messages by specifying a filter. 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.

Information	and Logs			
Local log	Controller log	Find and Replace	Compiler Output	
Filter	*			Y
	* Info			45
1 🖸 19	Info OR Warning)		d
2 🖸 19	Warning AND ID	E		
3 🕄 19	Warning NOT en Warning AND ID	ror E NOT Controller		d

Figure 3-9: Filtering the Messages

For example, filtering with **Warning NOT error** means that only the lines including the word "warning" but not the word "error" are listed.

Filtering Rules

The following rules apply when you work with filters:

- You can combine several strings by including one of the three following 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

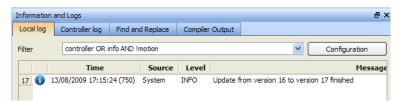


Figure 3-10: Filtering the Messages - Example

NOTE

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**.

About Scrolling

94

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

All the upcoming recorded messages are added at the end of the list, but your 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.

5.1.5.3 Find and Replace Tab

Informa	ion and Log	ļs		6 ×
Local lo	g Contr	oller log	Find and Replace	Compiler Output
Find wh	at: Machir	neSpeed		✓ Match complete word Find
Look in:	All doc	uments	(1	Find Unused Variables Replace
	All doc Dev Diction	uments Iary	2	3 Text 4
1 Cor	troller:PLC	Machinel	Logic: GS4-P1(4)	MLMstRun(PipeNetwork.Master, MachineSpeed);
2 Cor	troller:PLC	Machinel	Logic: GS4-N,NW1(0,2)) MachineSpeed >-
3 Cor	troller:PLC	Machinel	Logic: GS4-N,NW1(0,3)) lastMachineSpeed >-
4 Cor	troller:PLC	Machinel	Logic: GS4-N,NW1(2,3)) MachineSpeed >-
5 Cor	troller:PLC	Machinel	Logic: GS4-N,NW2(0,2)) MachineSpeed >-
6 Cor	troller:PLC	Machinel	Logic: GS4-N,NW2(2,2)) -> lastMachineSpeed 📃
7 Cor	troller:PLC	Dictionar	y:(Global)	(MachineSpeed,LREAL,LREAL#0,,)
8 Cor	troller:PLC	Dictionar	y:MachineLogic	(lastMachineSpeed,LREAL,LREAL#0,,)

Figure 3-11: Find and Replace

This tab enables you to search for all the instances of a string of characters (search is **not** case sensitive) within the entire environment, and replace it if desired.

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:
	Dictionary: search the dictionaryAll documents: search all files of the project and the dictionary
2	The search string can be with 'complete word' only. When selected, only the instances that match the complete words defined in the "Find what" field are selected (for example, a search for "MyVar" returns "MyVar" but not "MyVariable"). When this option is selected, the search is case sensitive.
3	To optimize your project, the Find Unused Variables option allows you to perform a search in the dictionary to locate variables not used ¹ in any program. Double- click a variable in the list within the table widget to open it in the Dictionary.
4	With the Replace button, you can replace in all documents a string with another one (or re-use one of the last ten entries) Replace in files Find what: MachineSpeed Replace with: OK
	The replace function is limited: it is not supported in the Dictionary, Pipe Network and HMI. For those, you have to make the change manually.

Once the search is done, the results appear in the table widget at the bottom of this tab. If a replace has been performed, the Text column provides more information about the replacement.

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

How to Understand the Location Details – SFC Programs

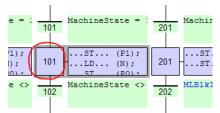
SFC Location details

¹A variable is **not used** when there is no effective usage of it in your entire project.

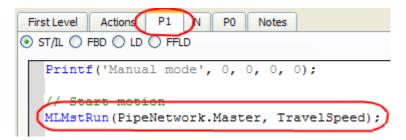
It can still be the case even when a value is assigned to a variable (e.g. MyVar := 100. * Axis1.Velocity ;). The variable MyVar becomes **used** when it is affected as an input argument (e.g. ^{Velocity} := MyVar ;).

2 controller:PLC MachineLogic: GS101-P1(4) MLMstRun(PipeNetwork.Master, TravelSpeed);

- 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

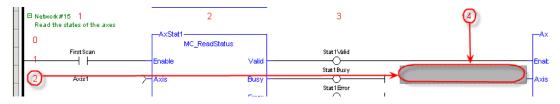


How to Understand the Location Details – FFLD Programs

FFLD Location details



- Controller : PLC 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



How to Understand the Location Details - ST Programs

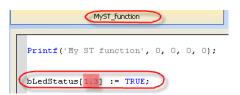
ST Location details

96

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



For more details, see "Find and Replace Operations" on page 97.

5.1.5.4 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 (ST/IL, FBD, FFLD)

5.1.5.4.1 Information and Logs

For more details, refer to the Information and Logs toolbox.

5.1.5.4.2 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 toolbox.

NOTE

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

5.1.5.4.3 Editor

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

🔳 Find and I	×	
Find what	MLGearInit	Find next
Replace with	MLGearInit2	Replace
		Replace all

Figure 3-12: 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 (for example, a search for $MyVar$ returns "MyVar" but not "MyVar1").
Find next	Allows you to select in the current editor the next instance of the matched string.
Replace next	Allows you to replace the next instance of the matched string.
Replace all	Allows you 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.

🗙 TIP

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

	MLBlkCreate('GEAR1'	, 'GEAR')	
MLGearI	Undo	Ctrl+Z	, false,
CNV1 :=	Redo	Ctrl+Y	');
MLCNVIn	Сору	Ctrl+C	
	Paste	Ctrl+V	
AXIS1 :	Cut	Ctrl+X	ATDriver'
MLAxisI	Find	Ctrl+F	60.0, SHI
GEAR2	Find All	Ctrl+Shift+F	.
MLGearI	Find Next	F3	, false,
	Add/Remove Breakpoint	F9	,,
CNV2 :=	Enable/Disable Breakpoint	Ctrl+F9	');
MLCNVInit	(CNV2, 1);		

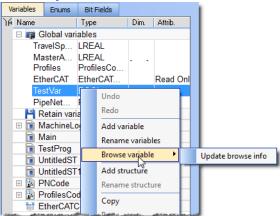
5.1.5.5 Browse Variable Tab

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 for more information.



• Right-click on a variable in the Dictionary and select **Browse Variable**. See Using the Dictionary's Browse Variable menu item for more information.



5.1.5.5.1 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.

Γ	× >	¢
6	Variables: (all)	-
	실 Axis1Status	
	실 Axis2Status	
-	실 bEStop	
-	실 bLedStatus[]	
-	실 bMasterAbs	
ŀ	실 bMasterRel	
-	🕒 EtherCAT	
ŀ	🕒 MachineLogic/lastMachineSpeed	
-	🕒 MachineLogic/lastTravelSpeed	
ŀ	la MachineSpeed	
-	실 MachineState	
	실 MasterAbsPos	
-	실 MasterDeltaPos	
÷.	실 PipeNetwork	
Þ	실 Profiles	
L.	🍓 TravelSpeed	
r Və	riables: (all) 🔻	1
<u> </u>		1
	Local variables only	
	Hide FB instances	

- 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.

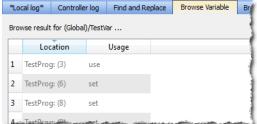
5.1.5.5.2 Using the Dictionary's Browse Variable menu item

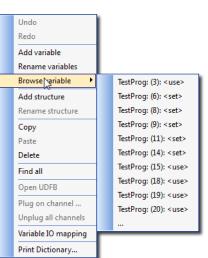
The first time this menu is accessed (per project instance) you are presented with **Update browse info**. This updates the browse data. Further selections of any variable result in the location and usage being shown directly in the menu.

 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.

• Selecting the ellipsis (...) brings you to the Browse Variable tab, which will be populated with the variable locations and usages.





5.1.5.6 Breakpoints tab

The Breakpoints tab lists all of the breakpoints in the PLC program, including their position and status. Double-clicking on an entry will take you to that location in the editor.

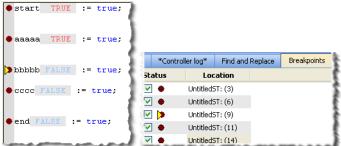
Breakpoints may be enabled and disabled singly by clicking the **Status** checkboxes. Buttons on the left of the tab provide the ability to remove single breakpoints, enable and disable all breakpoints, and remove all breakpoints.

Inform	nation an	nd Logs				
Loc	al log	*Controlle	er log*	Find and Replace	Breakpoints	Compiler Output
×	9	Status	Lo	ocation		
•	E	• • •	Untitled	iST: (1)		
_		• • •	Untitled	fST: (3)		
00		✓ ♦	Untitled	IST: (5)		
•		• • •	Untitled	fST: (7)		
~						

Button	Action Performed
×	Delete selected breakpoint from the list and editor.
••	Enable all breakpoints. This will make all breakpoints "active"
00	Disable all breakpoints. This will make all breakpoints "inactive"
•	Remove all breakpoints.

Right-clicking on 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 "hit" or 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.



🗙 TIP

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

As breakpoints set in SFC programs cannot be enabled or disabled, entries in the Breakpoints widget do not have a checkbox to perform these actions.

Information	n and Logs								
Local log	Controller lo	g	Find and Replace	Breakpoints	Compiler Output				
X	Status	Location							
	•	Main: GT2							
_	V 🔶	Up_Timer_UDFB_FFLD: NW1(0,0)							
00									
•									

Figure 3-13: Example of a breakpoint (Main: GT2) set in an SFC program.

For more information on breakpoints, see About Breakpoints and Setting, Removing, Enabling, and Disabling Breakpoints.

5.1.5.7 Compiler Output

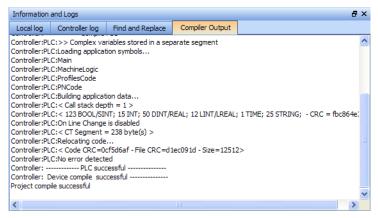


Figure 3-14: Compiler Output

This tab displays information about the last project compilation. It shows information messages as well as Warnings and Errors (highlighted in red). Successful and unsuccessful output is reported within this tab to help identify and troubleshoot problem areas of the program development.

mormation a	iu cogs					
Local log	Controller log	Find and Replace	Browse Variable	Breakpoints	Compiler Output	
Compile Projec Controller:	t Compile De	vice				
Controller:KVB Controller:KVB Controller:KVB Controller:KVB Controller:KVB Controller:KVB Controller:Controller:Controller:PLC Controller:PLC	: Warning : (Global : Warning : (Global : Warning : (Global : Warning : Untitlec : Warning : Untitlec : Warning : Untitlec : Compile PLC :Compile PLC :Compile V13,5,10):Sample_Inst contain):Inst_NewStructure 1):Inst_NewStructure 1 IST:Inst_NewStructur Modbus Files success 	g a member NewVar s variables which are is having a member contains variables w e is having a member e contains variables ful	(String variable) e not supported b Inst_NewStructu hich are not sup r NewVar (String	length is more than s by KVB ure.NewVar (String va ported by KVB variable) length is mo	upported Modbus string length of 2401 riable) length is more than supported Modbus string length of 2401 re than supported Modbus string length of 2401

★ 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 the image above, the first warning is for Sample_Inst. Double clicking that item opens that item's location in the Dictionary.

Name	Туре	Dim.	Attrib.	Init value	User	KVB
🗆 📷 Global variables						
TravelSpeed	LREAL			LREAL		
MasterAbsPos	LREAL			LREAL		
MasterDeltaPos	LREAL			LREAL		
and a second distance of the second	a strict and the street			and the second second		
PipeNetwork	PNCode					
Sample_Inst	NewStructure					
Inst_NewStructure1	NewStructure					 Image: A set of the set of the

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?

To clean-up your application, do as follows:

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

NOTE

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, the following messages are displayed:

- 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 does not have the expected type in a multiplication or division operation. Typically, REAL is the default precision for floating points, so you have to explicitly declare your long real constants with the LREAL# prefix when required.

5.1.5.7.1 How to Understand the Details of Location?

Same explanations contained in previous section **Find and Replace** are also applicable here.

5.1.6 Watch Window

This toolbox enables you to add variables to a dedicated watch window to display its value in real time.



Figure 3-15: 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 will be 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 or locked/unlocked (see Force a Variable), parameters may not be. AKD parameters in the Watch Window are read only.

5.1.6.1 Multiple Watch Windows

The KAS IDE allows you to group several variables in a single watch window, and to have up to 10 different watch windows. Each of them is displayed as a tab with its own label.

Explanation for each icon:

Icon	Description
+	Add a variable with the PLC Variable Selector
-	Remove a variable
1	Move up the selected variable
+	Move down the selected variable
	Add a new watch window. Each window is displayed as a tab with its own label. (Note that there is a limit of 10 tabs)
	Remove the selected watch window

Table 3-28: Watch Window - List of Icons

🗙 TIP

You can also add an existing variable in the watch window directly by using **drag-and-drop** from the Dictionary or the PLC editors.

Each variable in the table widget has the following information:

Field	Description
Name	Lists the variables as well as structure, arrays and expressions. You can double-click a variable (or press F2 key when it is selected) to edit its name (except for structure and array members)
Value	When the application is running, displays the variable or expression's value. You can 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

5.1.6.2 Access Structure and Arrays

Watch Window		đ×
\$ = 🛉 🖡 🔁 💪		
Name	Value	
🖃 PLCopen		~
.cmdID		
.actualState		≡
🖃 Main/ArrayTest		
ArrayTest[0]		
ArrayTest[1]		~
<		>
Watch 1		

Figure 3-16: 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.

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

5.1.6.3 Add Variable

Double-click the nodes ((Global), Main...) to expand their related variables
 TIP

Expand AKD node if you want to add AKD parameters to the Watch Window

- Select one from the list
- Click OK

Q PLC Variable S	Selector	? 🛛
Select a variable belo	ow.	
Name	Туре	
(Global) (Retain) (Retain) (Retain) (Retain) (Retain) (Retain) (Retain) (Retain)]	
·		
Expression:		🗌 Ins
Ok		Cancel

Figure 3-17: Watch Window - Selecting PLC Variable

This variable is then added to the current watch window tab.

5.1.6.4 Add an Expression

You can enter variable strings as an expression.

For example, if you want to add together two integer variables, follow these steps:

- 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)

PLC Variable Selector PLC Variable below.			
Name	Туре		
Global) Global) Getain) Getain) Getain) WerrorID1 WerrorID2 Machine			
Expression: Main/	WErrorID 1		🗸 Ins
Ok		Cancel	

Figure 3-18: Watch Window - Creating Expression

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

Reference PLC Variable Selector				
Select a variable below.				
Name	Туре			
Global) Global) Main Wain WErrorID1 WErrorID2 Machine				
	rrorID1 + Main/WErrorID2 V Ins			
Ok	Cancel			

• Then the expression is displayed into the watch window

۷	Vatch Window		đΧ
	+ - + + 🔒 🖬		
	Name	Value	
	Main/WErrorID1 + Main/WErrorID2		
	🖃 bLedStatus		
	bLedStatus[0]		
	bLedStatus[1]		
	bLedStatus[2]		
	bLedStatus[3]		

Figure 3-19: Watch Window - Displaying Expression

What you can include in a complex expression:

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

Please note that the DIVIDE SIGN (/) is not interpreted as an operator when used with prefixed variables (e.g. MachineLogic/lastTravelSpeed)

5.1.6.5 Force a Variable

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

You can double-click on the value of a variable (or press the **ENTER** key when it is selected) to open a pop-up window that allows you to:

• 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.

LastMachineState	8
2	
Force	
Lock	
Unlock	
31 24 23	16

Figure 3-20: Forcing a variable

• 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.



Test_bVar		
Name Value	Inst_NewStru	ucture 🛛 🔤
🖻 Test_bVar	Name	Value
Test_bVar[FALSE	🗆 Inst New9	Struc
Test_bVar[FALSE		ar ([TRUE]] (FALSE)
Test_bVar[[[TRUE]]	NewVa	
Test_bVar[FALSE		

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:

1To 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

Variables	Enums	Bit Fields			
Name		Value	Туре	Dim.	K
🗄 📭 G	lobal vari	ables			
💾 R	etain vari	ables			
🗆 🗎 N	lachineLo	igic			
las	tTrave	0.0	IREAL	_	
1 1	astTravelS	peed		×	
		· · · · · ·	~~~~		

If disabled (default), the Lock and Unlock buttons are not accessible:

7	Name		\mathcal{L}_{-}	Value	Туре	
	- 📷 (Global variable	es			
	A	xis1Status		185	DINT	
	A	xis2Status		185	DINT	
	bE	EStop		FALSE	BOOL	
	bL	edStatus			BOOL	0
	bl	MasterAbs		FALSE	BOOL	
	bl	MasterRel		FALSE	BOOL	
	Et	therCAT			EtherCAT	
	M	achineSpeed		0.0	IREAL	
	1	MachineSpeed			2	
			F	orce Lock)
	+ 1		U	nlock		
	🕀 🛃 🛛	PNCode ProfilesCode EtherCATCode	Ð		_	
m						

() 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.

• Unlock:

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

5.1.7 AKD Drive

First generation AKD drives support one axis per drive, while AKD2G drives support two axes per drive. When configuring drives in the EtherCAT Devices tab, you will need to designate which motion axis applies to each "drive axis", which is the actual axis the drive controls.

In addition to the different views, the WorkBench provides a toolbar and a status bar to display some extra information.

5.1.7.1 WorkBench Toolbar

The toolbar provides access to the following:

- Enable / Disable the drive (software enable)
- Start / Stop the Service Motion
- Mode:position / torque / velocity
- Disable & Clear Faults: Click this button to clear the fault, then click the Enable button to enable the drive again

💿 💿 🕼 Enable Stop 2 - Position Mode 🗸 Disable & Clear Faults Save To Drive							
To-name (Online) Settings Performance Servo Tuner Slider Tuning Exuke and Magnitude	You are connected to a drive and the communication is working						

Figure 3-21: AKD Toolbar

<u>★ TIP</u>

To stop all the AKD drives at the same time, click on the Stop button 🦳 in the Device Toolbar.

5.1.7.2 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 You can also set the Log message to get more details on the drive messages
- The drive status: active / inactive
- The software (SW) enable status
- The hardware (HW) enable status

No Faults Drive Inactive SW HW

Not Connected



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 (for example when SW or HW is not enabled: status is not green because a motion could not happen, and not red because it is not an error)

5.1.8 Status Bar

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

- Local version
- Controller version (application version located in the controller)
- Drives state
- Controller state (stopped/running)
- Connection state

🕞 Local: myUDFB2:25 🐼 Controller: UnnamedProject:3 Drives inactive Stopped Connected 🔇 🤮

Figure 3-23: Status Bar Labels

An icon *per between the Local and Controller versions allows to show any differences (for more details, Compare PLC Programs).*

The space on the left of the status bar is reserved for messages.

5.1.8.1 Local Version

This label provides information about the version locally present in the KAS IDE. There are three different states:

- Nothing displayed (for instance when no project is loaded)
- Version information (when available)
- Compilation error (background in **red**)

🗙 TIP

You can position the mouse over the text field to display a tooltip with the detailed version information.

5.1.8.2 Controller Version

This label provides information about the version present in the controller. There are three 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.

5.1.8.3 Drives State

There are three different states:

- Drives inactive (drives are disabled or your application is not connected to the target)
- Drives active (at least one drive is active)
- Drives error (at least one drive is in error)

5.1.8.4 Controller State

The Controller state label lets you know if the Controller is running or stopped. There are three different states:

- Nothing displayed (the label is empty when the KAS IDE is not connected to the target)
- Controller is stopped
- Controller is running

5.1.8.5 Connection State

The Connection label displays the Connection state between the KAS IDE and the Controller. There are five different states:

- Not connected
- Connecting
- Connected (background in green)
- Connection Error (background in red)
- Unexpected Disconnection (background in red)

🗙 TIP

You can hold the mouse over the text field to display a tooltip with some detailed information about the Error, and the Controller address when connected.

5.1.8.6 Color Codes

The Local and Controller version labels has an **orange** background in case of version mismatch between the 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.

List of use cases for the labels of the status bar

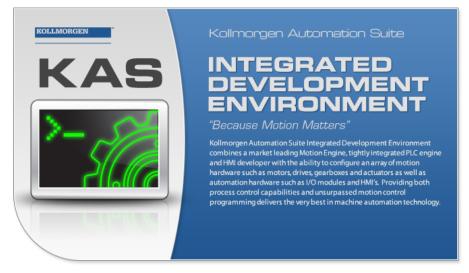
The following table summarizes all cases for the labels of the status bar.

Connection state	Local version	Controller version	Controller status	Connection status
Disconnected				Not Connected
Disconnected	Version A			Not Connected
Connecting	Version A			Connecting
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
Disconnected	Compile error			Not Connected
Connected	Compile error	Version A	Stopped or running	Connected
Comm. error	Version A			Connection Error
Disconnected	Version A			Unexpected disconnection

Table 3-29: Connection Status

112

6 Getting Started



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

6.1 KAS IDE to Runtime Compatibility

The KAS software includes two main software components, the IDE and the Runtime. The IDE resides on your PC and the Runtime resides on your programmable automation controller or programmable drive. For optimum operation, the IDE you install and run on your PC must be compatible with the Runtime you install on your controller or programmable drive.

The KAS installer and AKD PDMM or PCMM Runtime contain the version information in the filename:

```
KAS-Setup-3.05.0.60001.exe
KAS-PCMM-M-M1EC-3.05.0.60001.img
KAS-PCMM-M-MCEC-3.05.0.60001.img
KAS-AKD PDMM-M-M1EC-3.05.0.60001.img
KAS-AKD PDMM-M-MCEC-3.05.0.60001.img
```

NOTE

The version in the text above is only an example of the syntax. The firmware file will have a different version number.

The version can be determined by the following:

Version Source	Found From	Example		
IDE, Controller or Simulator	Help menu > About	Product details: General Version: 2. 5 Build 2.5.0.30540		
PDMM	Web server Settings Firmware tab	Firmware	Network	File System
		Firmware Information	ı	1
		Firmware version	_2.8.1.54839	
		Recommended file name	_KAS-PDMM-M-M	CEC-{version}.img
		المتحديد المتحصير ووسعا والمعاد متتعلى سادا	ورواف المتحلي المتحاطين الم	Contraction and Contraction

The compatibility between the IDE and the Runtime is defined by matching the digits in the versions. The IDE and Runtime use the same version scheme:

major . minor . micro . revision

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

Examples

IDE	Runtime	Compatible?	IDE to Runtime Connection
3.05 .xx.xxxx	1.2.xx.xxxx	NO, No connection.	The IDE displays an error message.
3.05 .0.xxxx	3.05 .1.xxxx	YES*	Connection possible. The IDE displays warning and requires user to press "ok" to continue.
3.05 .0.xxxx	3.05 .0.xxxx	YES	Normal connection.
3.05 .0.30540	3.05 .0.30540	YES	Normal connection.

* Not a recommended configuration. The IDE will operate, but there can be different features available between the IDE and the Runtime. It is recommended to upgrade either the IDE or the Runtime to matching versions.

6.2 KAS PLC Code to Runtime Compatibility

It is possible that after a Controller Runtime upgrade that the Runtime and the compiled project versions major.minor.micro may be different. The Runtime will not start if there is an incompatibility between the versions.

AKD PDMM or PCMM	An E24 error will be displayed and an error will be logged if compiled project version is incompatible with the Runtime version.
PAC	An error will be logged if the compiled project version is incompatible with the Runtime version.

To correct the incompatible versions, either re-compile and download the KAS project file using the same IDE version as the Runtime version OR change the Controller's Runtime version to match the compiled project's version.

KAS IDE User Guide |

6.3 KAS IDE to Project File Compatibility

This section helps describe forwards and backward compatibility between the KAS IDE and project files.

About Backward Compatibility

The KAS IDE is backward compatible, meaning the KAS IDE can open project files made with older versions of the software. There are a few things to note.

- An "older" project file will be converted to take advantage of features in the newer version.
- As there may be multiple conversions needed to update the file, if any step is not backward compatible you will be warned with a pop-up message and the process will continue with the next conversion step.
- You can exit without saving the project conversion.
- You may use the "Save As" function to make a copy of the newly converted project file, preserving the original.

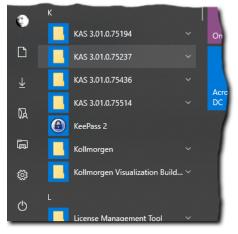
About Forward Compatibility

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 will simply open the file.

6.4 Starting the KAS IDE

There are multiple ways to start the KAS IDE.

- Double-click on a desktop shortcut.
- Double-click on a project file to open it with the most recently installed version of the KAS IDE..
- Windows 10: From the Start menu, under "K".
 - All installed versions may be 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? for more information.



• *Windows 7*: Open **All Programs** and start the **KAS IDE** application located under the **Kollmorgen** folder.

★ TIP

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

See the FAQs How do I launch my project with a particular version of the KAS IDE? and How do I choose the default version of the KAS IDE to use? for more options.

6.4.1 Find KAS IDE Version

The version of the KAS IDE or KAS Simulator being used is visible in the title bar of the application.



For more detailed information you may use the **About** command in the **Help** menu.

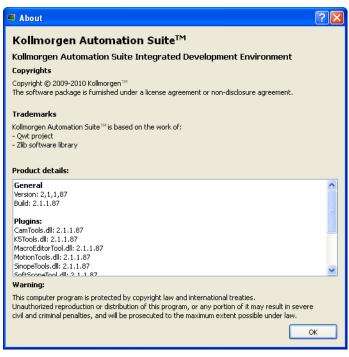


Figure 5-1: About Window

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 Help System

You can access the online help using the **Documentation** command in the **Help** menu.

See also "Use the Context-Sensitive Help provided in "

6.4.3 KAS GUI

For a better understanding of **KAS** menus, toolbar and workspace items (description and manipulation), refer to Describing KAS Graphical User Interface

6.4.4 KAS IDE Log Window

6.4.4.1 Log Information

The KAS log window (see Log Messages) provides a running display of activity related to the execution of the application. Items displayed include application startup and initialization information.

Infor	matio	n and Logs					5 ×
Loci	al log	Controller log	Find an	id Replace	Compiler	r Output	
Filte	r	*					Configuration
		Time		Source	Level		Me
1	0	13/08/2009 14:44	05 (679)	KMS IDE	INFO	Successf	ully installed plugin datas from Danaher M
2	0	13/08/2009 14:44	05 (910)	KMS IDE	INFO	Load libra	ary C:/Program Files/DanaherMotionSA/A
3	0	13/08/2009 14:44	05 (911)	KMS IDE	INFO	Load libra	ary C:/Program Files/DanaherMotionSA/A
4	0	13/08/2009 14:44	05 (912)	KMS IDE	INFO	Load libra	ary C:/Program Files/DanaherMotionSA/A
5	0	13/08/2009 14:44	05 (912)	KMS IDE	INFO	Load libra	ary C:/Program Files/DanaherMotionSA/A
6	0	13/08/2009 14:44	05 (914)	KMS IDE	INFO	Load libra	ary C:/Program Files/DanaherMotionSA/A
7	0	13/08/2009 14:44	05 (914)	KMS IDE	INFO	Load libra	ary C:/Program Files/DanaherMotionSA/A
8	0	13/08/2009 14:44	05 (915)	KMS IDE	INFO	Load libra	ary C:/Program Files/DanaherMotionSA/A
9	0	13/08/2009 14:44	05 (916)	KMS IDE	INFO	Load libra	ary C:/Program Files/DanaherMotionSA/A
10	0	13/08/2009 14:44	05 (916)	KMS IDE	INFO	Load libra	ary C:/Program Files/DanaherMotionSA/A
11	i	13/08/2009 14:44	05 (917)	KMS IDE	INFO	Load libra	ary C:/Program Files/DanaherMotionSA/A
<							>

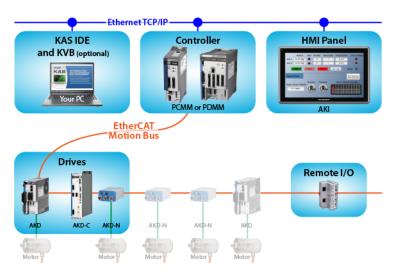
Figure 5-2: Log Messages

Information and Logs

6.4.5 Create a Project

There are six 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



6.4.6 Add and Configure a Controller

6.4.6.1 Add the Controller

To add a controller to your project:

- Click the New command in the File menu tso start the Controller Creation Wizard
- Choose the motion engine option (Pipe Network or PLCopen) and select the application template (see list below)

🔍 Add new	/ controller	×
Choose	an application template	
Template	Туре	
PipeNe	twork 🔿 Library	
	n	
O Coordin	nated Motion	
Name	Description	
	Simple initialization, relative position move	
1 Axis ST		
	Position and velocity moves, gearing, and drive status with optimized performance	
	Position and velocity moves, gearing, and drive status	
2 Axes ST	Position and velocity moves, gearing, and drive status	
	Finish Cancel	

Figure 5-3: Select an Application Template

- Click the Finish button
- Click the Save As command in the File menu
- Define the Project Name and its Location
- Click OK

List of available application templates

Template Type	Template name	Description
PipeNetwork	1 Axis FFLD	Simple initialization, relative position move
	1 Axis ST	Simple initialization, relative position move
	2 Axes FFLD	Position and velocity moves, gearing, camming, and drive status
	2 Axes ST	Position and velocity moves, gearing, camming, and drive status
	2 Axes SFC	Position and velocity moves, gearing, camming, and drive status with optimized performance.
PLCopen	1 Axis FFLD	Simple initialization, relative position move
	1 Axis ST	Simple initialization, relative position move
	2 Axes FFLD	Position and velocity moves, gearing, camming, and drive status
	2 Axes ST	Position and velocity moves, gearing, camming, and drive status
	2 Axes SFC	Position and velocity moves, gearing, camming, and drive status with optimized performance.

Template Type	Template name	Description
Coordinated Motion	2 Axes - Linear / Circular	Raster Scan Motion Path, 2 PLCopen axes
	3 Axes - Linear / Circular	Raster Scan Motion Path, 2 PLCopen axes and 1 PipeNetwork axis
	3 Axes - Linear (3D)	Diamond/Square Motion Path, 3 PLCopen axes
KAS Runtime	Library	Allows you to create a custom library (Create and Use Custom Libraries)

6.4.6.2 Configure the Controller

The controller is configured using the Controller Properties dialog box.

Controller Properties	×
X Co	ntroller Configuration
IP Address	10.50.68.30 👻
Controller Type	
PDMM or PCMM	
Project	
Version	
Download project sou	rce to the controller
Enable PLC variable re	emote access
Modbus Configuration	
KVB Panel	Cycle Time (20ms - 1000ms)
Other Modbus devices	s 200
	Ok Cancel

Figure 5-4: Configure the Controller Properties

To set-up the controller:

- 1. In the Project Explorer, right-click on the new controller to open the contextual menu
- 2. Select the **Properties** command
- 3. Define the IP Address

A note about addressing

- For the KAS Runtime Simulator, enter the localhost IP address: 127.0.0.1
- For the AKD PDMM or PCMM enter the IP address of the controller (e.g. 10.155.100.150). The default is 192.168.0.1, which is the same as the default IP address on the PCMM or AKD PDMM device.

NOTE

You must ensure that controller is accessible by the KAS IDE machine (see FAQ section for IT issues)

IMPORTANT

Do not use leading zeroes when entering an IP address. The connection to the controller will automatically fail if leading zeroes are used. For example, use **10.2.3.40** instead of **010.002.003.040**.

4. (Optional) Specify a version number (the string can be composed of any character)

Versioning can be useful when you make improvements to your application and need a version control system (Use a Version Control System). The version is saved in your project file.

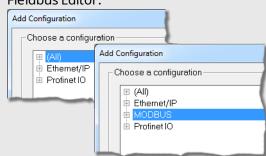
- 5. (Optional) Choose whether the project's source code should be downloaded to the controller. This is enabled by default and your preference is saved with the project. Disabling this option means a comparison of source on the controller and in the IDE will not be available.
- 6. (Optional) Choose whether PLC variables may be 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

Be sure to recompile the application before downloading it to the controller.

7. Select whether Modbus will go to a KVB Panel or be handled by another Modbus device, and set the **Cycle Time** (the communication update frequency, see "Modbus Slave" on page 651 for more information).

NOTE

• If KVB Panel is selected there will be no node to add a Modbus configuration in the Fieldbus Editor.



• If Other Modbus devices is selected then the KVB checkbox for Dictionary items will be hidden.. Motion configuration will need to be added from the Fieldbus Editor.

)ĥ	Name		Δ_{-}	Туре	Dim.	Attri	b.	Init value	U	KVB	Tag	Des	cription	
	🗆 🧊 GI	oba	l vari	ables										
	Axis	s1S	ta	DINT										
	Axis	32S	ta	DINT										
	bES	Stop)	BOOL				FALSE						
	bLe	dSt	at	BOOL	[03]									
	bM)Æ	Nam	ne 🔺	Туре		Dim.	Attrib.	Init	/alue	U	KVB	Tag	Description
	bM		- 📭	Global vari	ables									
	Eth			Axis1Sta	DINT									
	Ma			Axis2Sta	DINT									
-				bEStop	BOOL				FALS	SE				
				bLedStat	BOOL		[03]							
				bMaster	BOOL				FALS	SE				
				bMaster	BOOL				FALS	SE				
				EtherCAT	EtherCA	Τ		Read Only						
				Machine	LREAL				LRE	AL#0				
	(-	_	10	-WIT	-	-		-	-	_	_	-	

8. Click OK

6.4.7 Configure EtherCAT Motion Bus

★ TIP

Before configuring your EtherCAT settings, you may wish to add third party EtherCAT devices. See Add & Configure Third Party Devices for more information.

Double-click the **EtherCAT** node in the Project Explorer to open the EtherCAT properties dialog in the workspace. This window is composed of following tabs:

Tab	Description
EtherCAT Devices tab	Displays all the E-Bus devices present in the project tree
EtherCAT Master Settings Tab	Allows you to configure the EtherCAT bus master
ENI File Tab	Allows you to use an external configuration file
ESI Files Tab	Display, add, and remove available ESI files

KAS includes an integrated tool to configure the EtherCAT master and start up the fieldbus operation.

The configuration tool enables you to:

- Describe your motion topology as a configuration tree (see procedure in EtherCAT Devices tab)
- Associate variables to the I/O channels of devices (see procedure in Map Input and Output to Variables)

About Slave devices

Slave devices can support several PDOs (for the list, PDOs for AKD, AKD2G, AKD-N, and S300/S700). Some of them are mandatory; others are optional.

One of the main tasks of the EtherCAT configuration is to select the PDOs used by each slave (see also EtherCAT Devices Summary Form) and group them all in the EtherCAT image.

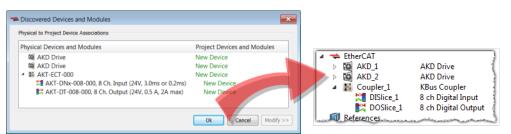
NOTE

PDOs contain real-time cyclic data which is deterministic. Non-cyclic data is not deterministic and is defined by Service Data Objects (SDO).

As explained in the introduction, input and output parameters are grouped in predefined blocks called PDOs.

6.4.7.1 EtherCAT Scan 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 will persist through subsequent scans unless you change them. The **Scan Devices** is accessed by doubleclicking the EtherCAT node in the project explorer, then pressing the **Scan Devices** button in the **Devices** tab.



Changing the network's physical topology (by adding moving, or removing devices) or changing the associations between physical devices and project devices will require that **Scan Devices** be run

again. The **Discovered Devices and Modules** window will display any changes and the **Modify** button will be enabled. When Modify is clicked the dialog box expands to show Available Project Devices and Modules. Click on a device and drag it to the corresponding **New Device** to create an association. Repeat as necessary for slices and modules.

nysical to Project Dev	ice Associatio	ons		Availab	le
hysical Devices ar	d Modules		Project Devices and Modules	Proje	ct Devices and Modules
 AKD Drive AKD Drive AKT-ECT-000 AKT-DNx 		Ch. Input (24V, 3.0ms or 0.2ms)	AKD_1 AKD_2 New Device New Device		Coppler_1
elected Device Detail	;				
Product Name	AKT-ECT	-000			
/endor ID	Kollmorg	en			
	Konnorg				
Revision Number	0x100000		-		
Revision Number Device Description Use Discover	0x100000 AKT-ECT) -000 EtherCAT Fieldbus coupler and Modules	1		Available
Revision Number Device Description Use Discover Physical to Physical 1	0x100000 AKT-ECT ed Devices Project Devic) -000 EtherCAT Fieldbus coupler and Modules e Associations	Project Devices and M	odules	Available Project Devices and Module
Vevision Number Device Description Use Discover Physical to Physical 1 @ Ak # Kk	Ox100000 AKT-ECT ed Devices Project Devic Devices and D Drive D Drive T-ECT-000) -000 EtherCAT Fieldbus coupler and Modules e Associations	AKD_1 AKD_2 Coupler_1	odules	
Use Discover Physical to Physical to Market AK	Ox100000 AKT-ECT ed Devices Project Devic Devices and D Drive D Drive T-ECT-000) -000 EtherCAT Fieldbus coupler and Modules e Associations I Modules	AKD_1 AKD_2 Coupler_1	odules	Project Devices and Module
Use Discover Physical to Physical to Market AK	0x100000 AKT-ECT- ed Devices Project Devic Devices and D Drive D Drive D Drive T-ECT-000 AKT-DNX-f) -000 EtherCAT Fieldbus coupler and Modules e Associations I Modules	AKD_1 AKD_2 Coupler_1	odules	Project Devices and Module
Vevision Number Device Description Use Physical to Physical 1 Physical 1 R AK Selected D	0x100000 AKT-ECT Project Devices Project Device Devices and D Drive D Drive D Drive T-ECT-000 AKT-DNx-0 wice Details) -000 EtherCAT Fieldbus coupler and Modules e Associations d Modules D08-000, 8 Ch. Input (24V, 3.0ms of	AKD_1 AKD_2 Coupler_1	odules	Project Devices and Module
Vevision Number Device Description Use Discover Physical to Physical 1 @ AK @ AK @ AK @ AK % Selected Dr Product 1	0x10000C AKT-ECT ed Devices Project Devic Devices and D Drive D Drive T-ECT-000 AKT-DNX-C wice Details) -000 EtherCAT Fieldbus coupler and Modules e Associations I Modules 	AKD_1 AKD_2 Coupler_1	odules	Project Devices and Module

Once the associations are complete, click **Ok**. See the following procedures for more details.

- New projects with no project devices: Scan and Associate Network Devices. This creates default configurations for the discovered physical devices/modules and couplers/slices.
- Adding, removing, or moving devices: Re-Scan and Change Device Associations.
- Adding, removing, or moving couplers with slices: Re-Scan and Change Coupler/Slice Associations.
- Adding, removing, or moving devices with modules: Re-Scan and Change Device/Module Associations.

The physical network topology will be compared to the project's expected network topology after the EtherCAT devices are associated and the PLC application is started. An error will be reported if the topologies do not match and the PLC program will not run.

NOTE

The physical topology must match the project's expected topology for the PLC application to operate the devices properly.

6.4.7.1.1 Scan and Associate Network Devices

To discover the physical EtherCAT devices, follow these steps:

- 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. <u>In the **Devices**</u> 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.

)evici	es 🚺	Ethe	erCAT Devic							
sh	now only	/ drives						Online Configuration Mo	de 🖉 Scan Devices	
s	Name	Model	HW Information	Mapped to Axis	Simulated	PDO Selection			S S	can Devices
s			s to NV memory	Save the paramete	r values of se	ected AKD drives	into the drive	s non-volatile memory.		-
			irmware	Update the firmwar						

• If physical devices are discovered during the scan, they will appear under the Physical Devices column.

Physical Devices and Modules Project Devices and Modules Image: MAD Drive New Device Image: MAD Drive New Device		
Image: MAXD Drive New Device Image: MAXT-ECT-000 New Device Image: MAXT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms) New Device		Project Devices and Modules
▲ MKT-ECT-000 New Device ☆ AKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms) New Device	🚳 AKD Drive	New Device
XI AKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms) New Device	AKD Drive	New Device
	4 🖺 AKT-ECT-000	New Device
AKT-DT-008-000 8 Cb Output (24V, 0.5 A 2A max) New Device	AKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms)	New Device
ART DT 000 000, 0 CH. Output (247, 0.5 A, 2A Hax) Thew Device	🔀 AKT-DT-008-000, 8 Ch. Output (24V, 0.5 A, 2A max)	New Device

- 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.

Filysica	I Devices and Modules	Project Devices and Modules
Q /	AKD Drive	AKD_1
10 A	AKD Drive	AKD_2
4 🔡 A	AKT-ECT-000	Coupler_1
14 A A A A A A A A A A A A A A A A A A A	AKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms)	DISlice_1
	AKT-DT-008-000, 8 Ch. Output (24V, 0.5 A, 2A max)	DOSlice_1

To change the device associations see Re-Scan and Change Device Associations.

To troubleshoot an error due to a missing ESI file, please see Unknown – Missing ESI File.

6.4.7.1.1.1 Unknown - Missing ESI File

The **Discovered Devices and Modules** view will identify a device as "Unknown" if a corresponding ESI file is not found during the Scan for physical EtherCAT devices. For example, the last three devices are missing ESI files:

hysical to Project Device Associations	
Physical Devices and Modules	Project Devices and Modules
🐼 AKD Drive	New Device
🚳 AKD Drive	New Device
AKT-ECT-000	New Device
🕅 AKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms)	New Device
🛤 AKT-DT-008-000, 8 Ch. Output (24V, 0.5 A, 2A max)	New Device
😮 Unknown - Missing ESI File	New Device
Onknown - Missing ESI File	New Device
Onknown - Missing ESI File	New Device

Click on the **Ok** button and the Unknown devices will be added to the Project View:

N 155 A	CAT KD 1	AKD Drive
	-	
	KD_2	AKD Drive
> 🔡 C	oupler_1	KBus Coupler
🕝 D	evice_1	Unknown
🕜 D	evice_2	Unknown
🔹 D	evice_3	Unknown

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.

General Properties	PDO Selection/Mapping	Distributed Clock	CoE Init	4
	ice_1 (Unknown)	General Pr	opertie	s
Information				
Name:	Device 1			
Description:	Unknown			
Vendor:	Beckhoff A	utomation GmbH		
Product Code:	0x44C2C5	2 (72100946)		
Revision Number:	0x120000	(1179648)		
URL:	Not Availat	ble		
ESI File:	ESI file mis	sing Im	port ESI File	
EtherCAT Address:	1004			
Topology				
Missing Port Informa	ation			

3. 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.

4. Import the ESI file using the **Import ESI File** button and the device specific information will be automatically updated in the Project View and General Properties.

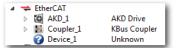
🔺 幸 EtherCAT	and a state of the second s
KD_1	AKD Drive
KD_2	AKD Drive
b 🔡 Coupler_1	KBus Coupler
Device_1	EK1100
2 Device_2	Unknown
Pevice_3	Unknown
and the second states and the second states and the	No. or and the second se

6.4.7.1.1.2 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, then the **Discovered Devices and Modules** view will identify the device as "Unknown" and will not be able to discover any modules under the device. For example, the last device is missing an ESI file:

Physical Devices and Modules	Project Devices and Modules
🗐 AKD Drive	New Device
AKD Drive	New Device
AKT-ECT-000	New Device
X AKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms)	New Device
🔀 AKT-DT-008-000, 8 Ch. Output (24V, 0.5 A, 2A max)	New Device
Onknown - Missing ESI File	New Device

Click on the **Ok** button and the **Unknown** device will be 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.

Information		
Name:	Device_1	
Description:	Unknown	
Vendor:	WAGO Kontakttechnik	GmbH & Co. KG
Product Code:	0x7500354 (12268424	14)
Revision Number:	0x0002 (2)	
JRL:	Not Available	
ESI File:	ESI file missing	Import ESI File
EtherCAT Address:	1004	

- 3. 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. A pop-up message will indicate that this a MDP device and the EtherCAT network must be scanned to re-discover the modules:

Ether	rCAT Project Device Added	23
<u> </u>	A MDP project device has been added from an imported or updated ESI file project's network topology does not match the physical network topology. Scan Devices to re-discover the network before re-compiling the project.	
	Do you want to Scan Devices now?	
	Tes	

After the scan and association of devices/modules, the MDP device and its modules will be added to the Project View:

⊿	+	EtherCAT	
	\triangleright	AKD_1	AKD Drive
	\triangleright	🔯 AKD_2	AKD Drive
	\triangleright	Coupler_1	KBus Coupler
	4	Device_1	750-354
		Module_1	750-5xx / 750-15xx (4b out)
		Module_2	750-5xx / 750-15xx (4b out)
		↑ Module_3	750-4xx / 750-14xx (4b in)

6.4.7.1.2 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.

ysical to Project Device Associations	
Physical Devices and Modules	Project Devices and Modules
🗐 AKD Drive	AKD_1
AKD Drive	New Device
AKT-ECT-000	Coupler_1
AKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms)	DISlice_1
🔀 AKT-DT-008-000, 8 Ch. Output (24V, 0.5 A, 2A max)	DOSlice_1

3. Click the **Modify** button to open the expanded view.

Physical Devices and Modules	Project Devices and Modules	Project Devices and Modules
 20 AKD Drive 20 AKD Drive 20 AKT-ECT-000 21 AKT-DNx-008-000, 8 Ch. Input (24V, 3.0 21 AKT-DT-008-000, 8 Ch. Output (24V, 0.5 		
elected Device Details		
Product Name		

4. Drag-and-drop *AKD_2* from **Available** to the second physical AKD drive. This associates the project device to the physical device.

Discov	vered Devices and Modules		
Physica	al to Project Device Associations		Available
Physi	ical Devices and Modules	Project Devices and Modules	Project Devices and Modules
	 [AKD Drive] AKD Drive] AKT-ECT-000 [21] AKT-DNk-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms) [22] AKT-DVB-000, 8 Ch. Output (24V, 0.5 A, 2A max) 	AKD_1 New Device Coupler_1 DISIce_1 DOSlice_1	₩ AKD_2
~ 0	iscovered Devices and Modules		×
P	hysical to Project Device Associations		Available
1	Physical Devices and Modules	Project Devices and Modules	Project Devices and Modules
	AKD Drive	AKD_1	AKD_2
	AKD Drive	New Device 🔯 AKD_2	
	▲ 🛃 AKT-ECT-000	Coupler_1	
	AKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms) AKT-DT-008-000, 8 Ch. Output (24V, 0.5 A, 2A max)	DISlice_1 DOSlice 1	
6	Discovered Devices and Modules	bosice_r	
	 Discovered Devices and Modules 		
	Physical to Project Device Associations		Available
	Physical Devices and Modules	Project Devices and Modules	Project Devices and Modules
	AKD Drive	AKD_1	
	AKD Drive	AKD_2	
	▲ MKT-ECT-000	Coupler_1 DISlice 1	
	AKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms) AKT-DT-008-000, 8 Ch. Output (24V, 0.5 A, 2A max)	DOSlice 1	

- 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.

Project View		Project View	
Name	Туре	Name	Туре
System System Controller Solution Define System Define S	Sinope Panel Controller	System System Ontroller PLC Solution	Sinope Panel Controller
Control Panel	Panel AKD Drive KBus Coupler AKD Drive	Control Panel	Panel AKD Drive AKD Drive KBus Coupler

6.4.7.1.3 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.

① 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.

6.4.7.1.3.1 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.

Project Explorer	8×			E	therCAT				
Project View									
Name	Туре	Der	rices Maste	r ENI File	ESI Files				
System Gontroller System System Softroller Softroller Softroller Motion Control Panel OP PDMM Onboard I/O	Sinope [10.50.67.95] Drive Panel		Et	herCAT D	vevices				Scan Devices
EtherCAT ID AKD 1	AKD Drive	-	Name	Vendor	Product	Model	FW Version	Mapped to Axis	Simulated
AKD_2	AKD Drive KBus Coupler	1	AKD_1 AKD_2		AKD EtherCAT Drive (CoE) AKD EtherCAT Drive (CoE)	AKD-P00306-NAEC-0000 AKD-P00306-NAEC-0000	01-11-00-002 01-11-00-002		N/A N/A

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.

Physical Devices and	d Modules	Project Devices and Modules	Project Devices and Modules
AKD Drive AKD Drive AKT-ECT-000	008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms)	AKD_1 AKD_2 New Device	Coupler_1
elected Device Details			
Product Name	AKT-ECT-000		
	AKT-ECT-000 Kollmorgen		
Product Name			

- 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).

sical Devices and Mo	dules	Project Devices and Modules	Project Devices and Modules
AKD Drive		AKD_1	Coupler_1
AKD Drive		AKD_2 New Device	A DISlice_1
	000, 8 Ch. Input (24V, 3.0ms or 0.2ms)	New Device	
Discovered Devices	and Modules	-	
Physical to Project Devi	ce Associations		Available
Physical Devices an	d Modules	Project Devices and Modules	Project Devices and Modules
		AKD 1	▲ L Coupler_1
AKD Drive		AND_1	- m couplet_1
AKD Drive		AKD_2	DISlice_1
AKD Drive		AKD_2 Coupler_1	
AKD Drive	008-000, 8 Ch. Input (24V, 3.0ms or 0.2	AKD_2 Coupler_1	
AKD Drive	008-000, 8 Ch. Input (24V, 3.0ms or 0.2	AKD_2 Coupler_1	
AKD Drive AKT-ECT-000	008-000, 8 Ch. Input (24V, 3.0ms or 0.2	AKD_2 Coupler_1	
AKD Drive	008-000, 8 Ch. Input (24V, 3.0ms or 0.2	AKD_2 Coupler_1	
AKD Drive AKD Drive AKT-ECT-000 AKT-DNx- Selected Device Details Product Name	008-000, 8 Ch. Input (24V, 3.0ms or 0.2 AKT-ECT-000	AKD_2 Coupler_1	

2. Drag-and-drop DISlice_1 from Available to the physical I/O slice (AKT-DNx-008-000).

Physical Devices and Moc	lules	Project Devices and Modules	Project Devices and Modules
AKD Drive	ares	AKD 1	A B Coupler 1
AKD Drive		AKD_2	DISlice_1
AKT-ECT-000		Coupler_1	
X AKT-DNx-008-0	00, 8 Ch. Input (24V, 3.0ms or 0.2ms)	New Device	
= Discovered Devices	and Modules		
Physical to Project Devi	ce Associations		Available
Physical Devices an	d Modules	Project Devices and Modules	Project Devices and Modules
AKD Drive		AKD_1	
AKD Drive		AKD_2	
AKT-ECT-000	008-000, 8 Ch. Input (24V, 3.0ms or 0	Coupler_1 (.2ms) DISlice 1	
AKTONA	008-000, 8 cm input (244, 5.0ms of c	Jane 1	
Selected Device Details			
Product Name	AKT-ECT-000		
Vendor ID	Kollmorgen		
Revision Number	0x100000		
Revision Number	AKT-ECT-000 EtherCAT Fieldbus co		

5. Click OK to confirm the modified associations.

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.



6.4.7.1.3.2 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.

Project Explorer	8×
Project View	
Name	Туре
 Imp System Imp Controller Imp PLC 	Sinope [10.50.67.95] Dri
Motion Motion Control PDMM Control A Contro	
	Scan Devices
AK Referen	Enable Online Config Mode Properties
	And the second se

- 2. The Discovered Devices and Modules dialog box appears after a successful scan.
- 3. Click the Modify button to open the expanded view.

Physical Devices and	d Modules	Project Devices and Modules	Project Devices and Modules
AKD Drive AKD Drive AKD Drive		AKD_1 AKD_2 New Device New Device	▲ Haraka Coupler_1 Main DISlice_1
	AKT-ECT-000		
Product Name	AKT-ECT-000 Kollmorgen		
Product Name Vendor ID Revision Number			

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).

sical to Project Device Associations		Available
ysical Devices and Modules	Project Devices and Modules	Project Devices and Modules
📾 AKD Drive	AKD_1	Coupler_1
AKD Drive	AKD_2	II DISlice_1
AKT-ECT-000	New Device	
AKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0.2ms)		
KT-DT-008-000, 8 Ch. Output (24V, 0.5 A, 2A max)	New Device	
Discovered Devices and Modules		
 Discovered Devices and Modules 		
Physical to Project Device Associations		Available
Physical Devices and Modules	Project Devices and Modules	Project Devices and Modules
M AKD Drive	AKD_1	▲ I Coupler_1
AKD Drive	AKD_2	DISlice_1
4 🗄 AKT-ECT-000	Coupler_1	_
MKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms or 0	0.2ms) New Device	
KT-DT-008-000, 8 Ch. Output (24V, 0.5 A, 2A	max) New Device	
Selected Device Details		
Product Name		
Vendor ID		
Revision Number		

2. Drag-and-drop DISlice_1 from Available to the physical I/O slice (AKT-DNx-008-000).

		Available
ysical Devices and Modules	Project Devices and Modules	Project Devices and Modules
AKD Drive	AKD_1	Coupler_1
AKD Drive AKT-FCT-000	AKD_2	DISlice_1
AKT-DNx-008-000, 8 Ch. Input (24V, 3.0ms	Coupler_1 or 0.2ms) New Device	
AKT-DT-008-000, 8 Ch. Output (24V, 0.5 A,		
		· · · · · · · · · · · · · · · · · · ·
Discovered Devices and Modules		
Physical to Project Device Associations		Available
Physical Devices and Modules	Project Devices and Modules	Project Devices and Modules
AKD Drive	AKD_1	
AKD Drive	AKD_2	
AKT-ECT-000	Coupler_1	
AKT-DNx-008-000, 8 Ch. Input (24V,		
KT-DT-008-000, 8 Ch. Output (24V,	0.5 A, 2A max) New Device	
Selected Device Details		
Product Name		
Vendor ID		
Vendor ID Revision Number		

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:

Project Explorer		₽×
Project View		
Name		Туре
4 📷 System	ı	Sinope
4 💼 Co	ntroller	[10.50.67.44] PDMM
Þ 💽	PLC	
Þ 💽	Motion	
	Control Panel	Panel
	PDMM Onboard I/O	
⊿ 关	EtherCAT	
	AKD_1	AKD Drive
	AKD_2	AKD Drive
4	Coupler_1	KBus Coupler
	DISlice_1	8 ch Digital Input
	DOSlice_1	8 ch Digital Output
	References	inter and in strength

6.4.7.1.4 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.

① 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.

6.4.7.1.4.1 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.

Project Explorer	8 ×			EtherCAT					
Project View		_							
Name	Туре	De	vices Mast	er ENI File ESI Files					
■ System ■ Controller ■ Controller ■ PLC ■ Motion Control Panel	Sinope [10.50.67.95] Drive Based C Panel		et	herCAT Devices					
PDMM Onboard I/O	Panel		Show only dri	ves				,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	Scan Devices
EtherCAT	750-354		Name	Vendor	Product	Model	FW Version	Mapped to Axis	Simulated
References	750-5xx / 750-15xx (4b out)	1	Device_1	WAGO Kontakttechnik GmbH & Co. KG	750-354 EtherCat fieldbus couple	r N/A	(Unknown)	N/A	N/A

• Another module needs to be added to match the device on the physical network.

To update the network topology:

 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.

hysical to Project Device Associations		Available	
Physical Devices and Modules	Project Devices and Modules	Project Devices and Modules	
▲ 1 750-354 ↓ 750-5xx / 750-15xx (8b out)		▲ I Device_1 ↓II Module_1	
1 750-4xx / 750-14xx (8b in)	New Device		
ielected Device Details Product Name			
elected Device Details	New Device		
elected Device Details			

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).

sical Devices and Modules	Project Devices and Modules	
750-354 1 750-5xx / 750-15xx (8b)	New Device	Device_1 ↓ ■ Module 1
1 750-4xx / 750-14xx (8b		Va Wodure_1
Discovered Devices and M	odules	
Physical to Project Device Assoc	iations	Available
Physical Devices and Modu	lles Project Devices and M	odules Project Devices and Modules
750-354	Device_1	Device_1
🐙 750-5xx / 750-15		Module_1
1 750-4xx / 750-14x	ox (8b in) New Device	
Selected Device Details		
Product Name		
Vendor ID		
Revision Number		
Device Description		

2. Drag-and-drop Module_1 from Available to the physical module (750-5xx).

1	ical to Project Device Associations		
Phy	sical Devices and Modules	Project Devices and Modules	Project Devices and Modules
4	9 750-354	Device_1	Device_1
	1 750-5xx / 750-15xx (8b out)	New Device	Module_1
_	1 750-4xx / 750-14xx (8b in)	New Device	
C			
	Discovered Devices and Module	S	
	Physical to Project Device Associations		Available
	Physical Devices and Modules	Project Devices and Me	odules Project Devices and Modules
	750-354	Device_1	
	VII 750-5xx / 750-15xx (8b		
	1 750-4xx / 750-14xx (8b	in) New Device	
	Selected Device Details		
	Product Name		
	Vendor ID		
	Revision Number		
	Device Description		

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.

Project Explorer	₽×
Project View	
Name	Туре
Gontroller Controller PLC Motion	Sinope [10.50.67.44] PDMM or PCMN
Control Panel PDMM Onboard I/O Control PDMM Onboard I/O	Panel
 Device 1 Module 1 Module 2 References 	750-354 750-5xx / 750-15xx (4b out) 750-4xx / 750-14xx (4b in)
And a second sec	And a second data and a second data and a second data

6.4.7.1.4.2 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.

hysical to Project Device Associations		Available
Physical Devices and Modules	Project Devices and Modules	Project Devices and Modules
▲ 🗄 750-354 ↓≣ 750-5xx / 750-15xx (8b out)	New Device New Device	▲ B Device_1 ↓ ■ Module_1 ① ■ Module_2
Product Name		
Product Name Vendor ID Revision Number		

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).

to Project Device Associations		Available
ical Devices and Modules	Project Devices and Modules Create New Confe	Project Devices and Modules
↓ 750-5xx / 750-15xx (4b out)	Create New Con	↓ ↓ ↓ ↓ ↓ Module_1 ↓ ↓ Module_2
Discovered Devices and Modules		
Physical to Project Device Associations		Available
Physical Devices and Modules 4	Project Devices and Modules Device 1	Project Devices and Modules
Ja 750-5∞ / 750-15∞ (4b out)	Create New Configuration	Bevice_1 Module_1 Module_2
Selected Device Details		
Product Name		
Vendor ID		
Revision Number		
Revision Number		

2. Drag-and-drop Module_1 from Available to the physical module (750-5xx).

	ociations		
Physical Devices and Mod	ules	Project Devices and Modules	Project Devices and Modules
■ 1750-354 ↓ 750-5xx / 750-1	5xx (4b out)	Device_1 Create New Configuration	Image: Constraint of the second se
Discovered Devices a Physical to Project Dev			Avaiable
Physical Devices and 750-354 Physical Devices and 750-5xx	d Modules / 750-15xx (4b out)	Project Devices and Modules Device_1 Module_1	Project Devices and Modules Project Device_1 Module_2
Selected Device Details			
Selected Device Details Product Name			
Product Name			

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.

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.

Project Explorer	₽×
Project View	
Name	Туре
 ▲ System ▲ Controller ▷ PLC ▷ Motion △ Control Panel ⊙ PDMM Onboard I/O 	Sinope [10.50.67.44] PDMM or PCMN Panel
 EtherCAT Device_1 Module_1 References 	750-354 750-5xx / 750-15xx (4b out)

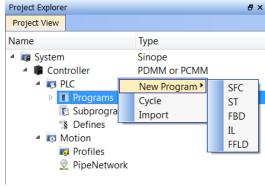
6.4.8 Overview: Create Programs

Once the project has been created and the EtherCAT devices have been defined, the program may be created.

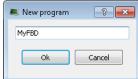
🗙 TIP

The KAS IDE has several program languages you may select from.

- Sequential Function Chart (SFC)
- Function Block Diagram (FBD)
- Free Form Ladder Diagram (FFLD)
- Structure Text (ST) and Instruction List (IL)
- 1. Right-click on the Programs item in the project tree and select the language you wish to use.



2. Name the program.



3. Click Ok to create the program and open the language-specific editor.

Project Explorer	₽×	MyFBD
Project View		
Name	Туре	│
🔺 📭 System	Sinope	
Controller	PDMM or PCMM	
🔺 💽 PLC		
Programs		
🖬 Main	SFC	💾
Actions	ST	
MyFBD	FBD	
C Subprograms		» · · · · · · · · · · · · · · · · · · ·
S Defines		
Motion		
Profiles		I
PipeNetwork		•••••••••••••••••••••••••••••••••••••••
		🖸
		۲ <u>۰</u>
Project Explorer Dictionary	Libraries	

The empty program is ready for you to add functions and function blocks. Other information you may need to complete your program includes:

- Overview: Create Variables
- Overview: Create and/or Call Functions & Function Blocks

- Overview: Using Constants
- Overview: Map Input & Output to Variables

For more in-depth information on the programming languages, see Create Programs in the KAS IDE Reference section.

6.4.8.1 Overview: Create Variables

Variables may be created in either of two ways:

• From the Variables tab of the Dictionary. Clicking Add Variable opens the Create Variable dialog box.

Dictionary	Ð
Controlle 👻	Track Selection
Variables E	nums Bit Fields
🝸 Name	Type Dim. KVB
■ Relation of the second seco	Redo Add variable Rename variables
•	A secondary
Project Explorer	Dictionary Libraries

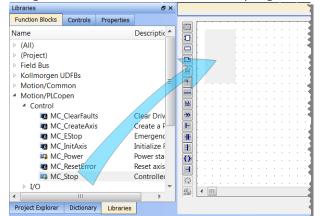
reate Variable		×
Name:	MasterAbsPos	ОК
Туре:	LREAL 🔻	Cancel
	Read Only Dim.:	
Init value:	LREAL#0	
User Group:	•	
Tag:		
Description:		
	🗌 External	
	🔲 Syb.	

• 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.

MyFBD	MyFFLD*		
OK × Image: wide of the state	MC_Power Enable Status Axis Busy Enable Positive Error Enable Negative ErrorD BufferMode	OK X Variables: (all) bPower bPun bPun Variables: (all) Image: Comparison of the pump of the	

6.4.8.2 Overview: Create and/or Call Functions & Function Blocks To call a function or function block:

1. Drag an item from the Libraries into a program

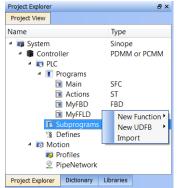


2. Name the new instance of the function or function block

See Call Functions or Function Blocks for more in-depth information.

To create 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 Create Functions and Function Blocks for more in-depth information.

See Also: Differences Between Functions and Function Blocks

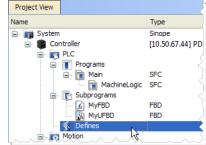
6.4.8.3 Overview: Using Constants

Constants, or Defines, exist at Internal, Global, and a Local level.

Level	Scope
Internal Defines	All the projects present on your machine
Global Defines	All the programs within your project. These are user-defined.
Local definitions	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 the following syntax.

#define Identifier Equivalence (* comments *)

See Use the Defines List for more in-depth information.

6.4.8.4 Overview: Map Input & 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 will vary 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 for Kollmorgen devices.
 - Directly map/unmap the inputs and outputs using the PLC Variable Selector.

See Map Input and Output to Variables for more in-depth information.

6.4.9 Adding Motion

There are two ways to create motion, depending on the motion engine:

- For Pipe Network, refer to Design Motion with Pipe Network
- For PLCopen, refer to Design Motion with PLCopen Axis
- If you are not sure which engine to use, refer to Pipe Network or PLCopen

For high-level discussions about motion, refer to the sections within Motion Concepts.

6.4.9.1 Design Motion with Pipe Network

The contents of this section detail how to create and modify a Pipe Network.

6.4.9.1.1 Create the Pipe Network

To create the Pipe Network, do as follows:

1. In the Project Explorer, double-click the **PipeNetwork** button to open the graphical Pipe Network Editor

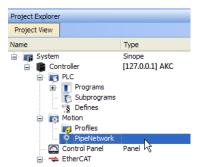


Figure 5-5: Pipe Network - Open Editor

NOTE

If you have created a project from a template (for instance the standard two-axis template) there is already a Pipe Network in the editor.

- 2. To add a new Pipe Block, right-click on the editor's background and select the **Add Pipeblock** command in the menu
- 3. Choose in the drop-down menu the type of Pipe Block you want to add

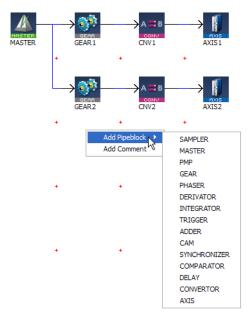


Figure 5-6: Pipe Network - Add Pipeblock

4. To link the newly created Pipe Block, move the arrow to the corresponding Pipe Block with a drag-and-drop operation

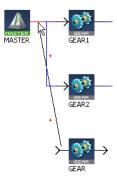


Figure 5-7: Pipe Network - Create a Link

How to 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.

How to change a link?

1. Select the link so that it becomes Red



Figure 5-8: Pipe Network - Edit a Link

You can either:

• Right-click and select the **Delete** command if you want to remove the link

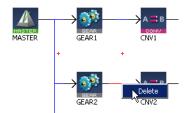


Figure 5-9: Pipe Network - Delete a Link

• Move the arrow to another Pipe Block with a drag-and-drop operation

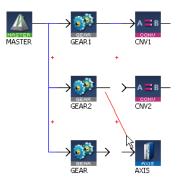


Figure 5-10: Pipe Network - Move a Link

See also §O.3: Application Notes for application examples

6.4.9.1.2 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

-				
	_	\rightarrow	Properties	Alt+Return
MASTER			Delete	Del
	+	*	Cut	Ctrl+X
			Сору	Ctrl+C
		→ 🐼	Plug on channel	
		GEF	Unplug all channe	ls
		GEARZ	CNV2	AXI52

Figure 5-11: 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 .

See example

In this example, the selected name "AngleAdvance" would be used in the PLC application program for this Pipe Network block.

PHASER :	AngleAdvance	? 🗙
General	Parameters	
	Name: AngleAdvance Type PHASER 🕶	
	OK Cancel	

6.4.9.1.3 Map the Axis to the Drive

To link the axis to an EtherCAT drive, you have to do the mapping as described in Mapped to Axis.

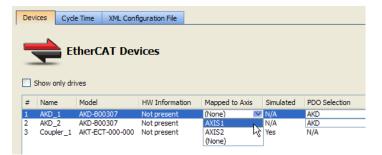


Figure 5-12: Pipe Network - Mapping Axis to Drive

6.4.9.1.4 Add Comments to 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

Axis 1 - Linear	Axis 1 - Linear

Figure 5-13: Pipe Network comments: editing, and deselected.

Click and drag the gray bar to move the comment.

6.4.9.1.5 Set the Axis Block Position Units

You can set up the position units in the parameter screen of the Axis block.

AXIS : AXIS1		×
General Parameters		
Position Unit	Degrees (motor shaft) ~	
User Units per Revolution		
Unit Label	deg	
Travel Speed	1000.0	
Acceleration	10000.0	
Decleration	10000.0	
Mode	Modulo ~	
Rollover Position	360.0	
OK	c	Cancel

Figure 5-14: Setting Axis Units

146

Some guidelines for suitable settings are given below:

- The user unit is adapted for the machine
- The user unit should be meaningful for the user

- The same user unit should be used for all related axes, for simplicity
- The user unit should 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)

Speed is defined in User Units for position / second, e.g. cm/s **Acceleration** in User Units for position / second2, e.g. cm/s2

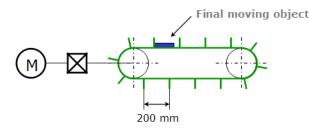


Figure 5-15: Setting the Units - Example

In the example above, 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 would correspond to 1200mm/revolution. Because the diagram specifies millimeters, one should select "Custom" from the Position Unit drop down box, enter "1200.0" into the User Units Per Revolution text box, and "mm" box into the Unit label text box.

6.4.9.1.6 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 5-16: 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.

6.4.9.1.7 Pipe Network Functions for the PLC

After creating the Pipe Network, the complete project has to be compiled before you can use the Pipe Network in your PLC Programs. Compiling creates a list of Functions that can be 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	Function Blocks included (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);
MLPN_POWER_OFF :	MLAxisPowerOFF(AXIS1); MLAxisPowerOFF(AXIS2);

Pipe Network Function	Function Blocks included (for 2 axis system)
MLPN_DEACTIVATE :	MLPipeDeact(PipeAXIS1); MLPipeDeact(PipeAXIS2);

For more details on all constant definitions related to Pipe Network, Use the Defines List

★ TIP

To see how these functions are used, open a project, go to the **Project Explorer**, right-click on PipeNetwork and select the **Show compiled Code** command

6.4.9.1.8 Initialize and Start up 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.

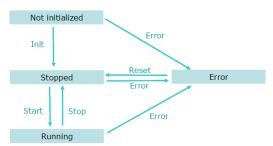


Figure 5-17: Motion State Machine

Each arrow represents a transition from one State to another one.

To start-up a Pipe Network in your IEC 61131-3 application program, you have to perform the following steps with their respective functions:

Step	ML function blocks	Description
Motion Init	MLMotionInit	Initialization of the Motion is done with this dedicated function Set the Motion engine update rate. Wait for acknowledgement: MLMotionStatus() = MLSTATUS_INITIALISED to continue program operation
Create Cam Profiles	Profiles(MLPR_ CREATE_ PROFILES);	Create Cam Profiles from cam files
Create Pipe Network	PipeNetwork (MLPN_CREATE_ OBJECTS);	
Motion Start	MLMotionStart	Starts the motion engine, motion bus driver, and initializes EtherCAT network to operational mode, then waits for acknowledgement: MLMotionStatus() = MLSTATUS_RUNNING to continue program operation
Power on all axes	PipeNetwork (MLPN_POWER_ ON);	
Activate the pipes	PipeNetwork (MLPN_ ACTIVATE);	

Step	ML function blocks	Description
Connect the axes to the pipes	PipeNetwork (MLPN_ CONNECT);	For example: in the following Pipe Network this function connects the Converter blocks (CNV1, CNV2 and CNV3) to the Axis blocks
		A B B AXIS MASTER GEAR1 CNV1 AXIS1 + + + + + + GEAR2 CAM2 CNV2 AXIS2 + + + + + +
		GEAR3

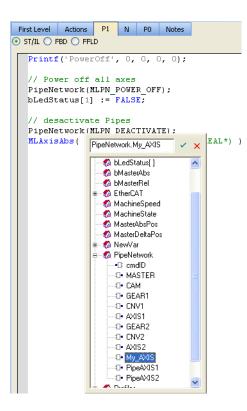
6.4.9.1.9 How the Pipe Network interacts with PLC programs

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 as well in the Variable Editor.

- Add the FB into your program (see procedure here)
- Select the variable to update



- Press CTRL+SPACE to open the Variable Editor
- Expand the PipeNetwork node and select the name of the Pipe Block in the list (all the Pipe Blocks created in the Pipe Network are listed)



Then your ST instruction is updated

// desactivate Pipes
PipeNetwork(MLPN_DEACTIVATE);
MLAxisAbs(PipeNetwork.My_AXIS, Position(*LREAL*))

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.

• Click the

icon to update your code

6.4.9.2 Design Motion with PLCopen Axis

This section explains how to modify an existing PLCopen Axis, and how to create a new one.

6.4.9.2.1 Create PLCopen Axis

To create a new PLCopen axis, follow these steps:

- 1. In the Project Explorer, right-click on the Axes folder under PLCopen and select the **New Axis** command in the menu
- 2. Give the new axis a name the Create PLCopen Axis dialog.

Create PLCopen Axis X		
Name:		
MyAxis1		
ОК	Cancel	

- 3. Fill in the axis properties by right-clicking on the new axis. See Modify PLCopen Axis for details.
- 4. Compile the project to generate the code for the PLCopen axis reference(s).

Each axis will show in the variable type PLCopenCode, which is used by the PLCopen Global variable.

Dictionary				
Controller:PLC V Trac	k Selection			
Variables Enums Bit Fiel				
Name Name	Type 🔺 Dim.S			
🗆 📭 Global variables	5			
CloseButton	BOOL			
EStopButton	BOOL			
OpenButton	BOOL			
Ready	BOOL			
ResetButton	BOOL			
StartMove	BOOL			
ActPos1	LREAL			
ActPos2	LREAL			
PLCopen	PLCopenCode			
Profiles	ProfesCode			
Retain variables				
MCFB_AKDFaultL MCFB_AKDFaultL MCFB_AKDFault				
MCFB_ARDFault_	v3			
B PLCopenCode				
actualState	DINT			
cmdID	DINT			
MyAxis1 ref	lib:AXIS REF			
InivAxis_Ter InivAxis_REF				
EtherCATCode				

5. The PLCopen.XXXXXX_ref (MyAxis1 in the example below) is an instance of an that can be used in your PLC programs.

<pre>IF bMotionEngineStarted THEN //Whe Inst_MC_ReadActPos(TRUE, PLCop </pre>	
-	THEN //When motion engine is ru TRUE, PLCopen.MyAxisl_ref)

★ TIP

The PLCopen axis Name and Number in the PLCopen Axis Data dialog, are automatically applied to the PLCopen instance under Global variables in the Dictionary when the project is compiled.

6.4.9.2.2 Modify PLCopen Axis

A PLCopen axis can be modified by using the PLCopen Axis Data dialog. To display this dialog you can:

- Double-click on a PLCopen axis in the Project Explorer
- Right-click the PLCopen axis in the project manager and select **Properties** in the menu as shown below

Pr	ojec	t Vie	w		6
Na	me				Туре
~		Sys	stem	n	Sinope
	~		Co	ntroller	[10.50.67.76] PDMM or PCMM
		>	5	PLC	
		~	-	Motion	
				📷 Profiles	
			~	💼 PLCopen	PLCopen
				✓ Axes	
				🙀 PLCOpenAxis1	Properties
				🙀 PLCOpenAxis2	Delete
				📷 Axes Groups	
			•	PDMM Onboard I/O	
		>	+	EtherCAT	le l
	~		M	References	

Figure 5-18: PLCopen Axis Context Menu

The PLCopen Axis Data dialog is displayed as follows:

🔳 PLCopen Axis Data				×
Axis Name: PLCOpenAxis	1]	Type:	Servo	~
Number: 1				
Axis Data Axis Lim	its			
Position Unit:	Degree (motor s	haft)	~	
User Units:	360.0		deg	/ revolution
Update Rate:	250 usec	~		
Rollover Position:	0		deg	
			OK	Cancel

Figure 5-19: PLCopen Axis Data Dialog

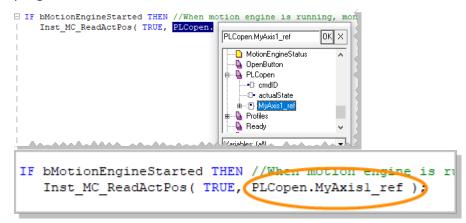
KAS IDE User Guide |

6.4.9.2.2.1 About Axis Name and Number

The PLCopen Axis Name and Number in the PLCopen Axis Data dialog, are automatically applied to the PLCopen instance of an AXIS_REF structure under Global variables in the Dictionary when the project is compiled.

Dictionary					
Controller:PLC V Track Selection					
Variables Enums	Bit Fields				
7 Name	Type 🔺 Dim.S				
🗆 🧊 Global varia					
CloseButton	BOOL				
EStopButtor	BOOL				
OpenButton	BOOL				
Ready	BOOL				
ResetButton	BOOL				
StartMove	BOOL				
ActPos1	LREAL				
ActPos2	LREAL				
PLCopen	PLCopenCode				
Profiles	Pri lesCode				
Retain varia					
E FaultMonit Main					
⊞ Main ⊞	Faulti askup				
⊞ MCFB Jog	aut_v3				
B A PLCopenC	de				
actualState	DINT				
cmdID	DINT				
MyAxis1 ref	lib:AXIS REF				
🗄 🛃 ProfilesCoo	e				
🗄 📅 EtherCATC	ode 🤇				

The PLCopen.XXXXXX_ref (MyAxis1 in the example below) instance can be used in your PLC programs.



154

6.4.9.2.2.2 Common Axis Parameters

Three types of axes are available: *Servo*, *Digitizing* and *Virtual Servo*. All types have common parameters related to an axis.

Name:	PLCOpenAxis1	Type:	Servo 👻
			Servo
Number: 1	1		Digitizing
			Virtual Servo

Figure 5-20: PLCopen Axis Parameters

Parameter	Description
Name	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.
Туре	A <i>Servo</i> axis is closed loop: commands are sent to the axis and feedback is read from the axis. After selecting this option you will need to define the Axis Data Parameters and Axis Limits Parameters.
	A <i>Digitizing</i> axis is read-only, open loop. Secondary feedback can be read by reading the actual position of the axis assigned to the secondary feedback. Digitizing axes can use extra feedback from the Drive or a third-party EtherCAT feedback device. Selecting a third-party EtherCAT feedback device will require you to enter EtherCAT Bus data (see Digitizing Axis Parameters) which will vary depending upon your EtherCAT devices. Also see How To Create a Digitizing Axis for specifics on different methods.
	A <i>Virtual Servo</i> is a servo axis with no feedback or drive hardware. The feedback for a virtual servo 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.

6.4.9.2.2.3 Digitizing Axis Parameters

When creating a digitizing axis you must set the Common Axis Parameters (name and number), the EtherCAT Bus parameters, and the Axis Data Parameters. The Digitizing axis type has some additional parameters to define for third-party EtherCAT feedback devices.

EtherCAT E	US	••				
Device:	AKD2G_1 V Address: 1002					
Feedback S	Feedback Settings					
CoE Obj	ect: Position actual value - Axis 1	\sim				
CoE Obj	ect Index: (hex) 0x 6064 CoE Object Subindex: 0	*				
	Encoder Type: Singleturn Absolute $$					
Single-turr	position bit count: 2					

Figure 5-21: PLCopen Axis - Bus Parameters

The EtherCAT device parameters are:

Parameter	Description		
Device	Select the device that will provide the digitizing axis feedback. The list is populated by the devices defined in the EtherCAT section of the 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 follows.		
	 first device: 1001 second: 1002 third: 1003 etc. 		
CoE Select the item that will be used to read the feedback. Object			
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	Typically 12, 16, 20, or 32-bits. A value of "0" indicates that rollover is not applicable.		

0)	<1A01 Contents		Siz	e: 8 bytes
	Index	Subindex	Object Name	Size [bit
	0x60E4	1	1st additional position act	32
	0x60E4	2	2nd additional position ac	22

• Feedback units will be configured on the AKD2G device based on the resolution specified for the digitizing axis. This is done using init commands during project compilation.

🗙 TIP

For more information see Working With A Digitizing Axis in PLCopen.

6.4.9.2.2.4 Axis Data Parameters

If a Servo axis is selected, two tabs are available: Axis Data and Axis Limits. If a Digitizing axis is selected, only the Axis Data tab is available. By default, new axes will be defined using degrees as the measurement unit.

Servo Axis		
Axis Data Axis L	imits	
Position Unit	Degree (motor shaft)	\sim
User Units	360.0 deg	/revolution
Update Rate	250 usec 🗸 🗸	
Rollover Position	0 deg	

Figure 5-22: Servo Axis - Axis Data

The Servo Axis - Axis Data parameters are:

Parameter	Description
Position Unit	This field sets the units (see UNIT Parameters) 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 will be 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.
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.

Parameter	Description
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 PLCopen code 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.
Rollover Position	The value at which the axis position rollovers to zero. Rollover Position is specified in User Units.
	For example:
	If the rollover position is 1000, 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.

6.4.9.2.2.5 Axis Limits Parameters



Figure 5-23: Servo Axis - Axis Limits

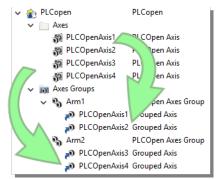
The Servo Axis - Axis Limits parameters are:

ParameterDescriptionIn PositionThe 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.

6.4.9.2.3 Create PLCopen Axes Groups

If your project utilizes coordinated motion and/or robotic transformations, you will need to create one or more Axes Groups from within the Project Explorer.

Axes Groups are collections of related axes defined by the data type AXIS_GROUP_REF, typically used by coordinated motion and function blocks such as MC_SetKinTra. Right-click on this section to create a new Axes Group. Groups are populated by dragging an existing axis from the above section into the group.



Axes Groups are easily modified by dragging the axes the axes between or within groups. Rightclicking on a group lets you **Rename** or **Delete** the group.

See Also:

- MC_KIN_REF Structure
- MC_SetKinTra
- What are Axes Groups?

6.4.9.2.4 Initialize and Start PLCopen

See Motion State Machine

The Motion State Machine is driven by the IEC 61131-3 application with the help of dedicated function blocks.

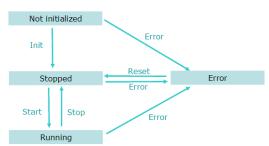


Figure 5-24: Motion State Machine

Each arrow represents a transition from one State to another one.

To start the PLCopen engine in your IEC 61131-3 application program, you have to perform the following steps with their respective functions:

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 acknowledgement: MLMotionStatus() = MLSTATUS_INITIALISED to continue program operation
Create Cam Profiles	Profiles(MLPR_ CREATE_ PROFILES);	Create Cam Profiles from cam files
Create/Initialize PLCopen Axes	PLCopen (0);	Create and initialize PLCopen axes from the PLCOpenCode attached to the PLCopen node in the Project tree. To view the PLCOpenCode, right click on the PLCopen node and select Show Compiled Code .
Motion Start	MLMotionStart	Starts the motion engine, motion bus driver, and initializes EtherCAT network to operational mode, then waits for acknowledgement: MLMotionStatus() = MLSTATUS_RUNNING to continue program operation

6.4.9.3 Design Coordinated Motion

This section discusses how to create a coordinated motion application, including adding coordinated motion to existing applications.

For more information on Coordinate Motion an the associated functions and function blocks see:

- Coordinated Motion in the Advanced Topics section
- Coordinated Motion Function Blocks

6.4.9.3.1 Create a Linear or Circular Coordinated Motion Application

A Coordinated Motion application can be created in one of two 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. More information about Axes Groups can be found in the section What are Axes Groups?.

NOTE

Coordinated motion can only be performed with PLCopen axes. Pipe Network axes do not support this feature, although 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.

Related axes are "grouped" in an axes group. Coordinated motion is then performed on an axes group. For more information see What are Axes Groups?.

🗙 TIP

There are two vital concepts to remember when making interpolated motion.

MC_MoveAdditive MC_MoveRelative

MC_MoveSuperimp

MC_MoveVelocity

MC_RemSuperAxis MC_SetOverride

Profile

- 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, and non-group coordinate system positions are only affected by non-group motion.

Group motion comman	ds Non-Group motion commands
 Coordinated Motion Group control Info Motion MC_AxisSetDefaults MC_GrpHalt MC_GrpSetOverride MC_MoveCircAbs 	 Motion/PLCopen ▷ Control ▷ I/O ▷ Info > PLCopenMotion ○ MC_AddSuperAxis ○ MC_Halt ○ MC_MoveAbsolute

NOTE

162

MC_MoveCircRel

MC_MoveDirRel

MC_MoveLinAbs

MC MoveLinRel

Typically 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.

Typically, the following set of 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 needs to be called between MLMotionInit() and MLMotionStart().

Inst MC CreateAxesGrp(TRUE, 'GROUP1', 6, 2, Group1 ref);

In the example above, 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' will be 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 the example above, the kinematic limits for axes group 'Group1_ref' will be set. The velocity limit will be set to 100.0 user units/second, acceleration and deceleration limits will be set to 300.0 user units/second2 and jerk will be set to 1000.0 user units per second3 (Jerk will be supported in a future release).

4. Call MC_CreatePLCAxis (AxisName, BusInterface, BusAddress, AxisNumber, AxisType, UserUnits, FeedbackUnits, Rollover, UpdateRate) to create a Coordinated Motion Axis. This function needs to 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 the example above:

- 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, which 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 be added to the group.

```
Inst_MC_AddAxisToGrp(TRUE, Group1_ref, CoordAxis1_ref, 0);
Inst_MC_AddAxisToGrp(TRUE, Group1_ref, CoordAxis2_ref, 1);
```

In the example above, 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. Note

that 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 the example above, drives CoordAxis1_ref and CoordAxis2_ref will be enabled and the position loop will be closed. Note that parameters 'TRUE, TRUE, 0' are place holders for future use and are not currently used.

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 the example above, 'Group1_ref' state will be changed from GroupDisabled to GroupStandby. The group must be in GroupStandby in order to perform motion.

 For the examples that follow, we want to set the current location of the axes in the group to 0, 0. This can be 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 the example above, the axis positions of 'Group1_ref' will be 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. The buffer mode, '0', is a placeholder for future use and is not currently used.

NOTE

No motion will be performed when this function block is executed.

- 10. *Optional:* To Add more axes to the group, modify the above code in the following 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 that will added to the group.
 - In Step 6: Add the additional axes to the group.
 - In Step 7: Power on the additional axes.
 - In Step 9: You will need to increase the size of the PosAbs array so it matches the number you used in step 2, and set the position of the additional axes to zero.

After the above function calls have been made, we can start coordinated motion moves.

Performing a Linear Move

Performing a Circular Move

164

6.4.9.3.1.1 What are Axes Groups?

Related axes are grouped in an Axes Group to support interpolation. Axes Groups are accessed via the type AXES_GROUP_REF. The following image shows the relationships between the different CSs and groups.

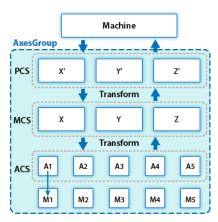


Figure 5-25: Overview of AxesGroup

The Axes Group, shown in blue above, 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.

6.4.9.3.1.2 Performing a Linear Move

Linear moves can be programmed using absolute or relative positions using the following function blocks:

- MC_MoveLinAbs which commands interpolated linear movement on an axes group to the specified absolute positions.
- MC_MoveLinRel which 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" on page 574). Once these steps have been performed, a linear move can be performed.

In the following examples, two linear moves will be performed. The first move is an absolute linear move that goes from (0, 0) to (100, 200). The second move is a relative linear move that goes a distance of (-75, 50) from the end of the first move. The BufferMode input is set to 'Buffered', meaning this move will wait for the first move to complete before it begins executing.

To Perform an Absolute Linear Move

Call MC_MoveLinAbs (Execute, AxesGroup, PositionArray, Velocity, Acceleration, Deceleration, Jerk, CoordSystem, BufferMode, TransitionMode, TransitionParameter). PositionArray is an array of absolute end positions containing one position for each axis in the group. The inputs velocity, acceleration, deceleration, and jerk establish the maximum values for the move.

In this example, PosArrayAbs[0] represent the x-axis and PosArrayAbs[1] represent the y-axis.

```
PosArrayAbs[0] := 100;
PosArrayAbs[1] := 200;
TransParam[0] := 0;
TransParam[1] := 0;
Inst_MC_MoveLinRel(TRUE, Group1_ref, PosArrayAbs, MaxVel, MaxAcc,
MaxDec, 0, MC_COORDINATE_SYSTEM_ACS, 1, 0, TransParam);
```

In the example a linear move will be 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 will be moved to 100.0. The axis stored in postiion 1 of the group will be 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/sec2'.
- The maximum jerk is currently not supported and can be set to a value of 0.
- The coordinate system is ACS
- The BufferMode is set to 1, indicating the move is buffered. For more information about buffer modes, see the Buffer Modes overview.
- The TransitionMode is set to 0, indicating no transition mode will be used. For more information about transition modes, see the Transition Between Moves section.
- The TransParam array is required and the contents can be set to 0 since the transition mode is not being used. There has to be one array entry for each axis in the group.

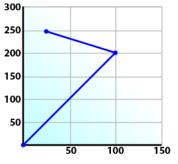
• To Perform a Relative Linear Move

Call MC_MoveLinRel (Execute, AxesGroup, Distance, Velocity, Acceleration, Deceleration, Jerk, CoordSystem, BufferMode, TransitionMode, TransitionParameter). The Distance input is an array of distances, one distance for each axis in the group. The inputs velocity, acceleration, deceleration, and jerk establish the maximum values for the move.

In this example, DistArrayRel[0] represent the x-axis and DistArrayRel[1] represent the y-axis.

```
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;
TransParam[1] := 0;
Inst_MC_MoveLinRel(TRUE, Group1_ref, DistArrayRel, MaxVel, MaxAcc,
MaxDec, 0, MC_COORDINATE_SYSTEM_ACS, 1, 0, TransParam);
```

In the example above, 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 will be moved a distance of -75.0. The axis stored in postiion 1 of the group will be moved a distance of 50.0.



To Perform a Linear Move With More Than Two Axes

NOTE

166

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 will do two dimensional moves if it only has two valid axes mapped to it. In order to perform higher dimensional moves, additional axes must be added to the group. The steps to do this are detailed in Create a Linear or Circular Coordinated Motion Application.

After the additional axes are added perform the following steps.

- 1. From within the Dictionary, update the array size of the variable being passed (PosArrayAbs and DistArrayRel in the examples above) to the Position input so that 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.

6.4.9.3.1.3 Performing 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" on page 574). Once these steps have been performed, a circular move can be performed.

In the following examples, two circular moves will be performed. The first move is an absolute circular move that goes from (0, 0) to (90, 90). CircMode specifies that the aux point (0, 180) will be crossed during the paths start to end. The second move is a relative circular move whose end point is (90, 90) from the end of the first move. In this move, CircMode specifies that the aux point (0, 90) is the relative center of the circle. The BufferMode input is set to 'Buffered', meaning this move will wait for the first move to complete before it begins executing.

• To perform an Absolute Circular Move:

Call MC_MoveCircAbs (Execute, AxesGroup, CircMode, AuxPoint[], EndPoint[], PathChoice, Velocity, Acceleration, Deceleration, Jerk, CoordSystem, BufferMode, TransitionMode, TransitionParameter).

In this example, PosCircAuxAbs[0] and PosCircEndAbs[0] represent the x-axis. PosCircAuxAbs [1] and PosCircEndAbs[1] represent the y-axis.

```
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;
Inst_MC_MoveCircAbs(TRUE, Group1_ref, MC_CIRC_MODE_BORDER,
PosCircAuxAbs, PosCircEndAbs, MC_CIRC_PATHCHOICE_CLOCKWISE, MaxVel,
MaxAcc, MaxDec, 0, MC_COORDINATE_SYSTEM_ACS, MC_BUFFER_MODE_
BUFFERED, MC_TRANSITION_MODE_NONE, TransParam);
```

In the example a circular move will be performed on axis group 'Group1_ref'.

- CircMode is defined as MC_CIRC_MODE_BORDER. This mode indicates that the AuxPoint array input will indicate a point on the circle which is crossed on the path from the starting point to the end point. See Circular Moves Diagrams for more information on 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, MC_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 will be 90.0. The absolute end point of the axis stored in position 1 of the group will be 90.0.
- PathChoice is only relevant when CircMode is set to MC_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/sec2'.
- The maximum jerk is currently not supported and can be set to a value of 0.
- The coordinate system is ACS
- The BufferMode is set to MC_BUFFER_MODE_BUFFERED, indicating the move is buffered. For more information about buffer modes, see the Buffer Modes overview.
- The TransitionMode is set to MC_TRANSITION_MODE_NONE, indicating no transition mode will be used. For more information about transition modes, see the Transition Between Moves section.
- The TransParam array is required. The TransParam array is a 2-element array containing the corner distance and velocity for the transition. Transitions are not used in this example and therefore the contents can be set to 0.

• To perform a Relative Circular Move:

Call MC_MoveCircRel (Execute, AxesGroup, CircMode, AuxPoint[], EndPoint[], PathChoice, Velocity, Acceleration, Deceleration, Jerk, CoordSystem, BufferMode, TransitionMode, TransitionParameter).

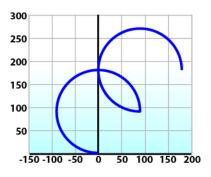
In this example, PosCircAuxRel[0] and PosCircEndRel[0] represent the x-axis. PosCircAuxRel[1] and PosCircEndRel[1] represent the y-axis.

```
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, MC_CIRC_MODE_CENTER,
PosCircAuxRel, PosCircEndRel, MC_CIRC_PATHCHOICE_CLOCKWISE, MaxVel,
MaxAcc, MaxDec, 0, MC_COORDINATE_SYSTEM_ACS, MC_BUFFER_MODE_
BUFFERED, MC_TRANSITION_MODE_NONE, TransParam);
```

In the example all the variables have the same meaning as the circular absolute example except:

- CircMode is defined as MC_CIRC_MODE_CENTER. This mode indicates that the AuxPoint array input will indicate the center point of the circle. See Circular Moves Diagrams for more information on 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, MC_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 will be 90.0. The relative end point of the axis stored in postiion 1 of the group will be 90.0.
- PathChoice is relevant when CircMode is set to MC_CIRC_MODE_CENTER. In this case, PathChoice is MC_CIRC_PATHCHOICE_CLOCKWISE which specifies the direction of the path.

168

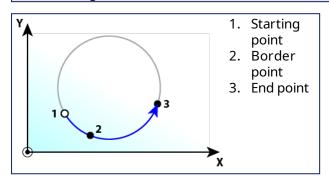


6.4.9.3.1.4 Circular Moves Diagrams

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.

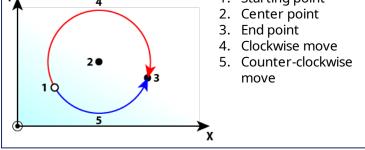
Advantages	 The border point can usually be reached by the machine, i.e. it can be taught.
Disadvantages	 Restricted to angles < 360° in one single command.



CircMode = CENTER

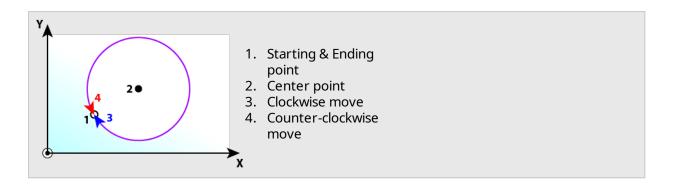
The user defines the end point and center point (= input 'AuxPoint') of the circle. The input 'PathChoice' defines clockwise or counter-clockwise motion. For Relative mode, both points are defined relative to the starting point.

Advantages Disadvantages	 Full 360° moves are possible. Cannot perform zero-distance moves. Over-determination of the circle equation.
Y 4	1. Starting point



NOTE

A 360° move will be performed if the end point is the same as the start point.



6.4.10 Add an HMI Device

To control your application, HMI panels can be downloaded to a dedicated HMI device (as described in the following procedure), but it can also be embedded into a targeted controller.

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. See KDN for more information.

When running the KAS Simulator, an internal HMI editor is also available to debug your application (for more details, Design the Control Panel with the Internal Control Panel Editor)

6.4.10.1 Create KVB Project

KVB projects are managed in the Project Explorer and can be created as follows:

- 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

💻 New HMI device	×
Select the type of HMI device to create.	
HMI Device	KVB minimum version
AKI2G CDA-MOD-05T-000: 5" Touchscreen	KVB 2.2
AKI2G CDA-MOD-07T-000: 7" Touchscreen	KVB 2.2
AKI2G CDB-MOD-07T-000: 7" Touchscreen	KVB 2.2
AKI2G CDB-MOD-12T-000: 12" Touchscreen	KVB 2.2
AKI CDA-04T: 4.3" Touchscreen	KVB 2
AKI CDA-07T: 7" Touchscreen	KVB 2
AKI CDA-10T: 10" Touchscreen	KVB 2
AKI CDB-07T: 7" Touchscreen	KVB 2
AKI CDB-12T: 12.1" Touchscreen	KVB 2
AKI CDB-15T: 15" Touchscreen	KVB 2
AKI CDC-12T: 12" Touchscreen	KVB 2
AKI CDC-15T: 15" Touchscreen	KVB 2
AKI CDC-21T: 21" Touchscreen	KVB 2
	OK Cancel

Figure 5-26: Select an AKI to add.

🗙 TIP

The content will vary depending upon what version of KVB is installed on your system. If you have KVB 1.1 installed you may open projects that have a KVB 2.1 panel, but you will not be able to 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

Note that this command is disabled when a KVB panel is already created for the current HMI device

6.4.10.2 Map Variables to HMI

For HMI, the variable mapping is done in two phases.

- Phase 1 Tag all of the variables to be exported in 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, as follows:

1. Open the Variable list editor available in the **Dictionary** toolbox

★ TIP

Double-click the Dictionary header to display the widget as a popup window in order to have more space.

2. In the KVB column, select the variables you want to map

Dictionary					
Controller:PLC		•	📃 Trad	k Selection	
Variables Enums	Bit Fields			\frown)
🝸 Name	Δ	Туре	Dim.	KVB	Init Attrit
🗆 📷 Global var	iables				1
ActPos1		LRE		✓	1
ActPos2		LRE			1
AND_LED		BOOL			1
Axis1		lib:AX			1
Axis2		lib:AX			1
AxisPowere		BOOL		~	1
CloseButto	n	BOOL		~	
CloseLoop		BOOL			
Current_Co	unt	DINT		✓	1
Dir		SINT			
Direction		BOOL		✓	1
EStop1		BOOL			
EStop2		BOOL			
EStopButto	n	BOOL		~	1
EtherCAT		Ether			Read
Input_1		BOOL			
Input_2		BOOL			
MoveAxis1		BOOL			- marine

Figure 5-27: Variable Mapping to KVB.

🗙 TIP

172

Selecting the variable alone does not guarantee it will be exported. The POU must be set to compile as well. See Ensuring Variables are Exported for more information.

IMPORTANT

Being 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 within 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 up to 240 characters.
- Variables of types "PNCode", "ProfilesCode", or instances of UDFB cannot be used within the HMI.
- 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 will be exported. See Compiler Output for

Information and	Logs								
Local log	Controller log	Find and Replace	Browse Variable	Breakpoints	Compiler Output				
Compile Project									
	Compile Dev	/ice							
and the second s	Compile Die	And in case of the local division of the loc	and the second data and the se	All the state of the second	CARGE STREET, SALES	Real Property lies and the same of the sam		Including the local division of the local di	and the second
Controller:	Generating	Modbus configuration	from the dictionary	KVB check hoxes					
Controller:KVB: V	Narning : (Global)	Sample Inst is having	a member NewVar			pported Modbus str	ing length of 24	0!	
		Sample_Inst is having Sample_Inst contains		(String variable)	length is more than s	pported Modbus str	ing length of 24	0!	- 1
Controller:KVB: V	Narning : (Global)		s variables which are	(String variable) not supported b	length is more than s by KVB				a length of 240!
Controller:KVB: V Controller:KVB: V	Warning : (Global) Warning : (Global)	Sample_Inst contains	s variables which are is having a member	(String variable) not supported b Inst_NewStructu	length is more than s by KVB ure.NewVar (String va				g length of 240!
Controller:KVB: V Controller:KVB: V Controller:KVB: V	Warning : (Global) Warning : (Global) Warning : (Global)	:Sample_Inst contains :Inst_NewStructure1	s variables which are is having a member contains variables w	(String variable) not supported b Inst_NewStructu hich are not sup	length is more than s by KVB ire.NewVar (String va ported by KVB	iable) length is more	than supported	l Modbus string	g length of 240!
Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller:KVB: V	Warning : (Global) Warning : (Global) Warning : (Global) Warning : Untitled	:Sample_Inst contains :Inst_NewStructure1 :Inst_NewStructure1	s variables which are is having a member contains variables w e is having a member	(String variable) e not supported b Inst_NewStructu /hich are not sup r NewVar (String	length is more than s by KVB are.NewVar (String va ported by KVB variable) length is mo	iable) length is more	than supported	l Modbus string	g length of 240!
Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller:KVB: V	Warning : (Global) Warning : (Global) Warning : (Global) Warning : Untitled Warning : Untitled	:Sample_Inst contains :Inst_NewStructure1 :Inst_NewStructure1 IST:Inst_NewStructure	s variables which are is having a member contains variables w e is having a member e contains variables	(String variable) e not supported b Inst_NewStructu /hich are not sup r NewVar (String	length is more than s by KVB are.NewVar (String va ported by KVB variable) length is mo	iable) length is more	than supported	l Modbus string	g length of 240!
Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller:	Warning : (Global) Warning : (Global) Warning : (Global) Warning : Untitled Warning : Untitled	:Sample_Inst contains :Inst_NewStructure 1 :Inst_NewStructure 1 IST:Inst_NewStructure IST:Inst_NewStructure Modbus Files successf	s variables which are is having a member contains variables w e is having a member e contains variables	(String variable) e not supported b Inst_NewStructu /hich are not sup r NewVar (String	length is more than s by KVB are.NewVar (String va ported by KVB variable) length is mo	iable) length is more	than supported	l Modbus string	g length of 240!
Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller: Controller:	Narning : (Global) Narning : (Global) Narning : (Global) Narning : Untitled Narning : Untitled Narning : Untitled	:Sample_Inst contains :Inst_NewStructure 1 :Inst_NewStructure 1 IST:Inst_NewStructure IST:Inst_NewStructure Modbus Files successf	s variables which are is having a member contains variables w e is having a member e contains variables	(String variable) e not supported b Inst_NewStructu /hich are not sup r NewVar (String	length is more than s by KVB are.NewVar (String va ported by KVB variable) length is mo	iable) length is more	than supported	l Modbus string	g length of 240!
Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller:KVB: V Controller: Controller: Controller:PLC:C	Narning : (Global) Narning : (Global) Narning : (Global) Narning : Untitled Narning : Untitled Narning : Untitled Compile PLC Compile PLC	:Sample_Inst contains :Inst_NewStructure 1 :Inst_NewStructure 1 IST:Inst_NewStructure IST:Inst_NewStructure Modbus Files successf	s variables which are is having a member contains variables w e is having a member e contains variables ful	(String variable) e not supported b Inst_NewStructu /hich are not sup r NewVar (String	length is more than s by KVB are.NewVar (String va ported by KVB variable) length is mo	iable) length is more	than supported	l Modbus string	g length of 240!

★ TIP

Non-primitive data types which are not exported over Modbus generate a warning message in the Compiler Output window. See Compiler Output for more information.

3. Compile the application to create the Modbus mapping file.

★ TIP

Generally you do not need to access this, but if you do the file used for manually importing variables into KVB (named **HMI Variable Import File.txt**) is located in the following location.

- Windows 7: C:\Users\(user)\AppData\Local\Kollmorgen\KAS\Project
- Windows XP: C:\Documents and Settings\(user)\Local Settings\Application Data\Kollmorgen\KAS\Project

NOTE

- If you modify the set of tagged variables in the dictionary, you have to update the text file by recompiling the project.
- The Modbus variables defined in KAS IDE are imported in Kollmorgen Visualization Builder only when you start KVB (there is no update in real-time between the two applications).
- 4. Open the **KVBProject** from the Project Tree. This will automatically transfer the tagged variables..

List of variables that you can export

The following types of variables can be exported to the HMI:

- The fundamental data types: BOOL, SINT, INT, USINT, UINT, BYTE, WORD, DINT, UDINT, DWORD, TIME, REAL, LREAL, STRING.
- Arrays of supported data types
- Structures that include members of supported data types

Examples of structures that you can 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 due to the embedded structure has an INT member and therefore the outer structure can be exported too.

Examples of structures that you cannot export

• A structure that includes instances of UDFB or LWORD members only.

6.4.10.3 Design KVB Panel with Kollmorgen Visualization Builder

• Double-click the new KVB panel to open the builder (for more details, refer to Using Kollmorgen Visualization Builder.)

🔳 Oper	Kollmorgen Visualization Builder?
?	Do you want to open the Kollmorgen Visualization Builder for this project?
	Yes Cancel

Figure 5-28: Open the Kollmorgen Visualization Builder Builder

IMPORTANT

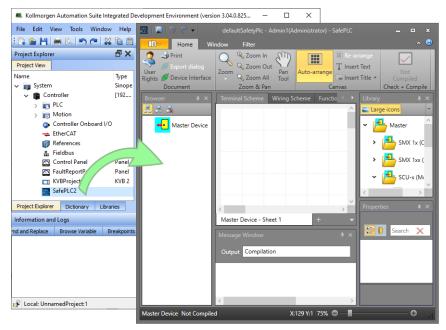
You must have the specific application already installed on your machine.

★ TIP

Be sure to close the Kollmorgen Visualization Builder before deleting the KVB Panel from the IDE.

6.4.11 Working with SafePLC2 Projects

A node may be added to the Project tree which connects to a SafePLC2 project. This allows you to synchronize a SafePLC2 project with a KAS project. The KAS IDE automatically generates the PDOs for the connected devices and will establish the Black Channel with the EtherCAT Safety network.

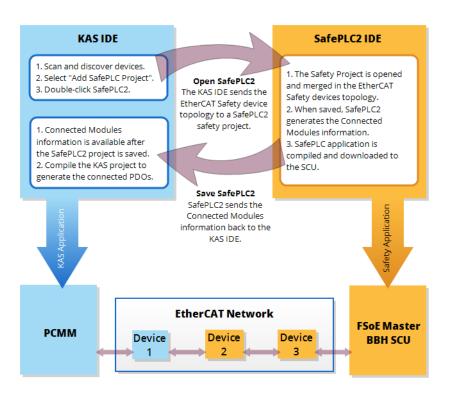


There are two methods for creating the synchronization between KAS Runtime and SafePLC2.

- Direct this is used when both the KAS IDE and SafePLC2 are installed on the same computer.
- Import this is used when KAS IDE and SafePLC2 are installed on different computers.

6.4.11.1 Connecting to a SafePLC2 Project - Direct Method

This method is used when SafePLC2 is installed on the same computer as the KAS IDE. The steps assume you have already added a SafePLC2 project to the KAS project tree (see Adding or Importing a SafePLC2 Project).



1. Double-click on the SafePLC2 node. Alternatively, right-click on **SafePLC2** and select **Open SafePLC2**.



- 2. SafePLC2 will be launched and the Safety Project will be opened. SafePLC2 automatically populates the safety devices from the EtherCAT Safe topology defined in the KAS IDE.
- 3. Complete and save the SafePLC2 Safety Project.
- 4. Compile the Safety Project and download it to the SCU1 controller.
- 5. Return to the KAS IDE, complete and compile the project.

The KAS IDE automatically creates the required black channel communication required for FSoE from the connected module information passed to the KAS IDE from SafePLC2.

- 6. Download the compiled KAS application to the controller and run it.
 - The Black Channel will establish the Safety network connection when EtherCAT goes into the operational state.
 - The SCU will transition to Run mode.

NOTE

- The KAS project supports a single SafePLC2 instance and project.
- The SafePLC2 node cannot be deleted while the application is running.
- The KAS IDE cannot be closed while SafePLC2 is running with a synchronized project.

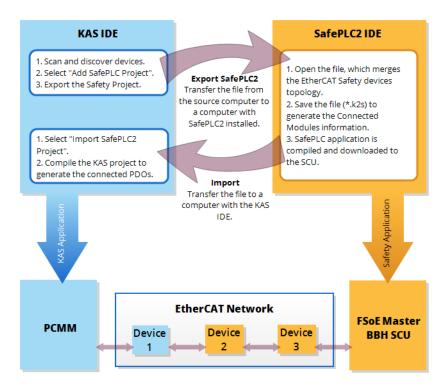
See Also:

176

AKD2G Safety Parametrization Using FSoE with SCU-1-EC and PCMM/AKD PDMM

6.4.11.2 Connecting to a SafePLC2 Project - Import Method

This method is used when the KAS IDE and SafePLC2 are not installed on the same computer. The steps assume you have already added a SafePLC2 project to the KAS project tree.



- 1. Export the SafePLC2 node (see Exporting a SafePLC2 Project).
- 2. Name and save the .k2s Safety Project file.
- 3. Transfer the exported file to a computer which has SafePLC2 installed.
- 4. Open the Safety Project file with SafePLC2. SafePLC2 automatically populates the safety devices from the EtherCAT Safe topology defined in the KAS IDE.
- 5. Complete and save the SafePLC2 Safety Project.
- 6. Compile the Safety Project and download it to the SCU1 controller.
- 7. Transfer the completed .k2s Safety Project file back to the computer with the KAS IDE.
- 8. Import the completed Safety Project (see Adding or Importing a SafePLC2 Project).
- 9. Complete and compile the KAS project.

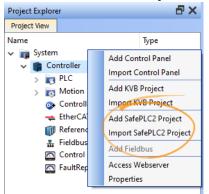
The KAS IDE automatically creates the required black channel communication required from the connected module information passed to KAS from SafePLC2.

- 10. Download the compiled KAS application to the controller and run it.
 - The Black Channel will establish the Safety network connection when EtherCAT goes into the operational state.
 - The SCU will transition to Run mode.

6.4.11.3 Adding or Importing a SafePLC2 Project

A SafePLC2 project may be added to the KAS IDE, or you may import an existing project.

- 1. Right click on **Controller** in the Project tree.
- 2. Select Add SafePLC2 Project or Import SafePLC2 Project.



When importing a project you are prompted to select the file's location.

There may only be a single instance of a SafePLC2 project in the KAS IDE. If none exists, a SafePLC2 Project will be added to the end of the Project tree. If there is a project you will be prompted to overwrite the project or cancel the import. Note that the import cannot occur if SafePLC2 is running.

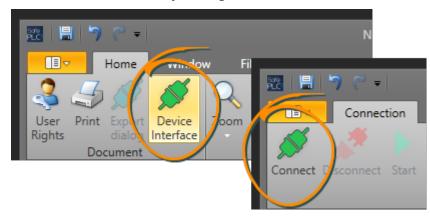
3. A node is added to the end of the Project tree.

6.4.11.4 Exporting a SafePLC2 Project

- 1. Right-click on the **SafePLC2** node in the Project tree.
- 2. Select Export.
- 3. Navigate to and select the location to save the exported file.
- 4. Click Save.

6.4.11.5 Connect SafePLC2 to an SCU-1-EC

1. Connect to the SCU-1-EC by clicking the "Device Interface" and then click the Connect button.



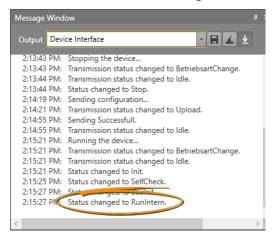
2. After successfully connecting to the device, click the "Send Configuration" to download the safety application.

Image: Second	CKAS_SafePLC2 - Admin1(Administrator)	
Connect Disconnect Start Stop Very Send Configuration	Lock Generate Serd Network Export Send Connection Diagnos on onfiguration Report Configuration Profile SCU-1-EC (Master)	
Browser 3 ×	Terminal Scheme Wiring Scheme Functional Scheme	
Peripherals Input Output		
Encoder Profile		
✓ Profile ✓ Statis ✓ AXD2C6 Axis 2 (Slave 2) ♦ Profile	SCU-1-EC (Master) - Sheet 1 +	
Axis Slaves Comparison Slaves Slaves	Message Window # × Output Device Interface	
SCU-1-EC (Master) - Sheet 1	2:10:09 PM: Compiling_ 2:10:10 PM: Compilation successful. Compilation time: 1 seconds.	
	2:10:30 PM: Scope initializing. 2:11:59 PM: Compiling 2:12:00 PM: Compilation successful. Compilation time: 1 seconds. 2:12:36 PM: Scope initializing.	
	2:13:16 PM: Connecting 2:13:17 PM: Connection status changed to Authenticating. 2:13:17 PM: Status changed to RunIntern. 2:13:17 PM: Connection status changed to Connected.	
SCU-1-EC (Master) Compiled	Connected Idle RunIntern	
NOTE		
The download s	status is available at the	status bar of the SafePLC2 application.
SCU-1-EC (Master) Comp	piled Sending Configuration (34%)	Connected Upload Stop

3. After a successful download of the application, click on the "Start" button to run the safety application.



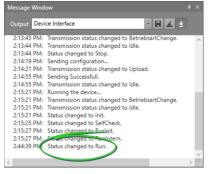
4. The status of the SCU will change to "RunIntern"



6.4.11.6 Put It Together: KAS, SCU-1-EC, and the AKD2G Working Together

- 1. Run the KAS application.
- 2. When the "Execute" input on the FSoEParamsInit function block goes high, the safety parameters are transferred to the AKD2G.
- 3. On a successful transfer, the safety network will start exchanging the Safety Process Data.

• SafePLC shows the message "Status Changed to Run" in the message window.



• The Status LED on the BBH SCU-1-EC will display "4".



• The status of the AKD2G drive in KAS IDEWorkBench or WorkBench will display "Safety State: 6 – Normal operation" and "FSOE State: 54 - Process Data"

Parameter Load/Save	You are connected to a drive and the communication is working.
 B Device Settings B Device Diagnostics 	Name: no_name
Axis 1 (1)	Device Model: AKD2G-SPE-6V06D-A2F3-0000
Axis 2 (2)	Device Type: 2 - EtherCAT
	For Six O Demor
	SS1[1]

🗙 TIP

Checking the AKD2G's status is also recommended.

6.4.12 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
- 2. Compile the Application
- 3. Connect to the Controller
- 4. Download the Application
- 5. Start/stop the Device

6.4.13 Advanced Project Creation

This section provides reference for setting up hardware, performing tasks which may not apply to every user, and other more advanced topics.

6.4.14 Add and Configure Drive

6.4.14.1 Add the Drive

- 1. In the Project Explorer, right-click the EtherCAT node to open the menu
- 2. Select the **Add Kollmorgen Device** command (this option is only enabled when you are **not** connected to the controller)
- 3. Select the specific device to add.

Go AXD AXD EnterCAT Drive (CoE) 0x(2) AXD-N AXD-N BeherCAT Drive (CoE) 0x(0) AXD-N AXD-N BeherCAT Drive (CoE) 0x(0) AXD-N AXD-N BeherCAT Drive (CoE) 0x(0)	evision
AKD-N AKD-N EtherCAT Drive (CoE) 0x1(1) AKD-N AKD-N EtherCAT Drive (CoE) 0x2(2)	
AKD-N AKD-N EtherCAT Drive (CoE) 0x2(2)	
AKD AKD EtherCAT Drive (CoE) 0x1(1)	
5300_5400_5600_5700 S300/5400/5600/S700 EtherCAT Drive (CoE) 0x1(1)	
S300_S700 (new core) S300/S700 EtherCAT Drive (CoE) with new core (DRVCNFG3.16 = 1) 0x1(1)	
Power supply	
Name Description R	evision
AKD-C AKD-C power supply (CoE) 0x1(1)	
AKD-C AKD-C power supply (CoE) 0x2(2)	
Terminal Couplers	
Terminal Couplers Name Description R	evision

NOTE

When choosing an AKD be sure to select the newest revision. Older revisions may show in the list of options, but they may not be feature complete. If one of these drives is selected, only the General Properties tab will be available.

4. Click **Ok** when you are done (for more details about the AKD drive GUI, click here)

5. In the Project Explorer, right-click the device node and select the Rename command



6. Define the name for the new device Note that the name is limited to 10 characters and can only include letters, numbers, and underscores.

🔳 Rename	? x
AKD_Drive3	
Ok	Cancel

7. Click OK

🗙 TIP

An alternative method to add a drive is to rely on the auto scan feature.

When an AKD drive is added to the project tree, it must be mapped to a physical drive. This step is explained in Add & Configure Third Party Devices.

6.4.14.2 Add and Configure Drive

6.4.14.2.1 Add the Drive

- 1. In the Project Explorer, right-click the EtherCAT node to open the menu
- Select the Add Kollmorgen Device command (this option is only enabled when you are not connected to the controller)
- 3. Select the specific device to add.

iollmoi rives			
Ľ	Name	Description	Revision
6			
	AKD-N	AKD-N EtherCAT Drive (CoE)	0x1(1)
	AKD-N	AKD-N EtherCAT Drive (CoE)	0x2(2)
	AKD	AKD EtherCAT Drive (CoE)	0x1(1)
	\$300_\$400_\$600_\$700	\$300/\$400/\$600/\$700 EtherCAT Drive (CoE)	0x1(1)
	\$300_\$700 (new core)	S300/S700 EtherCAT Drive (CoE) with new core (DRVCNFG3.16 = 1)	0x1(1)
ower st	apply		
	Name	Description	Revision
	AKD-C	AKD-C power supply (CoE)	0x1(1)
	AKD-C	AKD-C power supply (CoE)	0x2(2)
rmina	Couplers		
Ť	Name	Description	Revision
5	AKT-ECT-000	AKT-ECT-000 EtherCAT Fieldbus coupler	0x120000(1179648)

NOTE

182

When choosing an AKD be sure to select the newest revision. Older revisions may show in the list of options, but they may not be feature complete. If one of these drives is selected, only the General Properties tab will be available.

4. Click **Ok** when you are done (for more details about the AKD drive GUI, click here)

5. In the Project Explorer, right-click the device node and select the Rename command



6. Define the name for the new device Note that the name is limited to 10 characters and can only include letters, numbers, and

underscores.	
Nename	? ×
AKD_Drive3	
Ok	Cancel
[

7. Click OK

★	TIP	

An alternative method to add a drive is to rely on the auto scan feature.

When an AKD drive is added to the project tree, it must be mapped to a physical drive. This step is explained in Add & Configure Third Party Devices.

6.4.14.3 Configure the AKD2G/AKD Drive

The Configuration tab under the EtherCAT : AKD tab allows you to configure drives once the following conditions have been met:

- The scan has been performed
- Your project is compliant with the physical devices on the EtherCAT network
- You have activated the Online Configuration mode

You then 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.

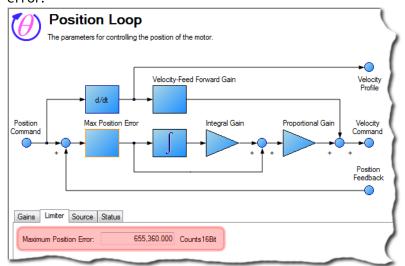
Project Work Provide	: <mark>[]을 H 북 S. (5 전)</mark>		<u>]</u> ;% ≅ 2 & 3 3 4 0 • 0 : <mark>8</mark> : 4 3 : 1 → ₩ 0 0 0
Nume type Sector	Project Explorer	8 ×	EtherCAT: AKD_2
State State <td< td=""><td>Project View</td><td></td><td></td></td<>	Project View		
Controller DisStor All Diver DisStor All Diver DisStor All Diver DisStor All Diver Diver Marchin Diver Marchi	Name	Туре	General Properties Configuration CoE Init-Commands CoE Object-Dictionary
Constraint of the constra			
Project Explore Detonery Ubraries Projec	PLC Programs Programs Programs Main Add Source Product Population Population	SFC SFC Pand AVD Drive	Construct [Childref] Construct [Child

Figure 5-29: AKD Configuration

2. Define the motion parameters in the **Limits** tab to configure the limited motor torque, speed, and acceleration

Limits		
This page shows all the drive limi	ts all in one place.	
Current Limits		
Positive Peak Current:	18.000	Arms
Negative Peak Current:	-18.000	Arms
Velocity Limits		
Positive Speed Limit:	3,000.000	rpm
Negative Speed Limit:	-3,000.000	фm
Over-Speed Limit:	3,600.000	фm
Position Limits		
Maximum Position Error:	655,360.000	Counts16Bit
Position Limit 0	0.000	Counts16Bit
Position Limit 1	1,048,576.000	Counts16Bit
Acceleration Limits		
Acceleration:	10,000.170	rpm/s
Deceleration:	10,000.170	rpm/s
Motor Limits		
Motor limits are set through the	Motor Foldback Screen:	Goto Foldback

3. Define the motion parameters in the **Position Loop** tab to configure the limiting following error.



- 4. Define the resolution of the feedback position in the **Feedback** tab Note that 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. Then, you must define the units to be used for the motion 1 :
 - For Pipe Network, refer to Set the Axis Block Position Units
 - For PLCopen, refer to Axis Data Parameters

NOTE

184

User units in the PLC language editors are:

- Position : User unit
- Velocity User unit/sec
- Acceleration: User unit/sec2

Several AKD tabs contain units that follow the standard AKD format:

1The normal units screen in the AKD Work bench GUI is not included in the IDE

- Position: 16 bits/rev
- Velocity: RPM
- Acceleration: RPM/ Sec
- 6. To ensure high performance, define the load for your servo system.. KAS IDE provides several options for performing the drive tuning:
 - Slider Tuning Allows adjustment to the desired bandwidth using the slider (precalculated tuning)
 - Performance Servo Tuner Takes the drive through an automatic tuning sequence
 - Manual Tuning Allows you to set gains individually for Current Loop, Velocity Loop, and Position Loop

For more details on AKD configuration, AKD Drive Configuration

For more details on AKD Firmware Download, refer to the FAQ section.

NOTE

After your application is downloaded to the controller you can activate the Online Configuration Mode to configure your drives with the **Setup Wizard...** For more details, AKD Setup Wizard...

6.4.14.4 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

Sho	w only drives							🔎 Scan De
	Name	Vendor	Product	Model	FW Version	Drive Axis Number	Mapped to Axis	Simulated
	AKD_1	Kollmorgen	AKD EtherCAT Drive (CoE)	AKD-M00306-MCEC-0000	01-20-00-001	1	(None) ~	N/A
	AKD_2	Kollmorgen	AKD EtherCAT Drive (CoE)	AKD-P00306-NAEC-0000	01-20-00-001	1	(None) ~	N/A
	Coupler 1	Kollmorgen	AKT-ECT-000 EtherCAT Fieldbus coupler	N/A	(Unknown)	N/A	N/A	N/A
	Coupler 2	Kollmorgen	AKT2G-ECT-000-000 EtherCAT Gateway	N/A	(Unknown)	N/A	N/A	N/A
	Al_Slice_1	Kollmorgen	AKT2G-AN-430-000 4Ch. Ana. Input +/-10V Diff., +/-20mA SingleEnded, 16 Bit		05	N/A	N/A	N/A
	AO Slice 1	Kollmorgen	AKT2G-AT-425-000 4Ch. Ana. Output +/-10V, 12bit	AKT2G-AT-42	04	N/A	N/A	N/A
	AO_Slice_2	Kollmorgen	AKT2G-AT-410-000 4Ch. Ana. Output 0-10V, 12bit	AKT2G-AT-41	04	N/A	N/A	N/A
	Al Slice 2	Kollmorgen	AKT2G-AN-240-000 2Ch. Ana. Input PT100 (RTD)	AKT2G-AN-24	07	N/A	N/A	N/A
	Al_Slice_3	Kollmorgen	AKT2G-AN-400-000 4Ch. Ana. Input Thermocouple (TC)	AKT2G-AN-40	07	N/A	N/A	N/A
	DI Slice 1	Kollmorgen	AKT2G-DN-002-000 2Ch. +/- Counter 24V, 100kHz	DN-002	04	N/A	N/A	N/A
	DI Slice 2	Kollmorgen	AKT2G-DNH-008-000 8Ch. Dig. Input 24V, 10us	N/A	(Unknown)	N/A	N/A	N/A
	DI Slice 3	Kollmorgen	AKT2G-DN-008-000 8Ch. Dig. Input 24V, 3ms	N/A	(Unknown)	N/A	N/A	N/A
	DO Slice 1	Kollmorgen	AKT2G-DT-008-000 8-channel digital output terminal 24 V DC, 0.5 A	N/A	(Unknown)	N/A	N/A	N/A
	DI SafeSlice 1	Kollmorgen	AKT2G-SDI-004-000, 4 Ch. Safety Input 24V	EL 1904	06	N/A	N/A	N/A
	DO SafeSlice 1	Kollmorgen	AKT2G-SDO-004-000, 4 Ch. Safety Output 24V, 0.5A	FI 2904	04	N/A	N/A	N/A
	AKD2G 1	Kollmorgen	AKD2G Dual Axis FS1 EtherCAT Drive (CoE)	AKD2G-SPE-6V06D-A1F3-0000	M 02-05-00-000			
						1	PLCAxis2 ~	N/A
						2		N/A
	Device_1	BBH Products GmbH		SCU-	03.00.00.61	N/A	N/A	N/A
	AKD2G 2	Kollmorgen	AKD2G Dual Axis FS2 EtherCAT Drive (CoE)	AKD2G-SPE-6V06D-A2F3-0000	M_02-05-01-000			

6.4.14.5 AKD Setup Wizard...

The wizard allows you to configure drives once the following conditions have been met:

- The scan has been performed
- Your project is compliant with the physical devices on the EtherCAT network
- You have activated the Online Configuration mode

You then have access to the AKD parameters that are used when the drive is running.

🛠 AKD Setup Wizard		X
🔿 🍞 Start	Welcome to the AKD Setup Wizard	O Learn more about this topic
 Communications Power Motor Feedback Brake Turning Move Finished 	You will be taken through the following steps to configure your drive. 1. Confirm communication with the correct drive 2. Monitor Bus Voltage and configure the Regen. Resistor. 3. Select the motor you are using 4. Select and test the motor's feedback 5. Parameter for controlling a motor's brake. 6. Turning 7. Perform a test move You can hit Finish to go directly to the expert view.	
	< Back	Next > Finish

Figure 5-30: AKD Setup Wizard

6.4.14.6 Configure Onboard I/O

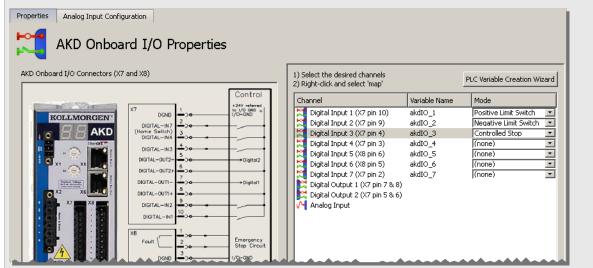
The procedure to define the local I/Os of the AKD drive is very similar to the one for I/O slices, with the following exceptions:

- Channel column also contains in brackets the connector and pin number
- PLC variable selection applies to digital inputs as well as analog inputs and outputs.

6.4.14.7 Digital Input Mode

NOTE

The way to set the Digital Input mode on an AKD drive has changed with KAS 2.6. Previously, this setting was done in the Onboard I/O view (seen below) where the IO was mapped to PLC variables.



The 'Digital Inputs and Outputs' are now accessed using the new AKD GUI integration inside KAS. This will allow you to save this setting inside the drive instead of pushing it every time the application is started. Files created with KAS 2.5 or earlier, with specific mode settings for any Digital Input of an AKD drive, will automatically be converted to the new project format and these settings will be removed. A dialog box will pop up prior to removal to inform you which settings will be removed from the project.

Project com	patibility: AKD Digital IO mode configuration	? ×
	This project file contains configurations for the AKD Digital Input mode settings are no longer supported in KAS projects.	e. These
-	Please apply these settings directly to the AKD parameters. Below is t Drives that have a Digital Input mode configuration that you need to	
	To set the AKD parameters from the IDE, switch to Online Config Mod navigate to the EtherCAT: AKD drive 'Digital Inputs and Outputs' view	
	Open the Help topic on AKD Views.	
Mode Mappir	ng Details	
AKD_1		
LimitSwitchPos	bigital input i	
LimitSwitchNeg		
ControlledStop	- Digital Input 5	
AKD_2		
LimitSwitchPos		
LimitSwitchNeg		
ControlledStop	- Digital Input 4	
	ОК	

For each drive you will find the list of digital inputs with their specific mode settings. You can copy/paste the content of this text area to a text editor to keep track of them before closing the dialog box.

6.4.14.7.1 How to configure Digital Input mode setting inside WorkBench

To set the mode of a Digital Input

- 1. Connect to the controller
- 2. Go to Online Configuration Mode
- 3. Open the desired drive's WorkBench by double-clicking on the AKD drive node in the project tree view.
- 4. Open the Digital Inputs and Outputs view

Ino_name (Online)* ₩ Ø Settings ⊕ ♥ Communication	[👥 Digital I	Input	s and	Outputs			@ Lea	rn more about this to
Power	This page show	s current s	tate of eacl	n of the I/O pins and allow	s to select the	function each pin perfor	ms.	
- 🔞 Motor	General Purpose I/Os	X9 I/Os						
🖲 🔮 Feedback 1	General Purpose Dig							
- O Feedback 2	General Purpose Dig	ital inputs-						
		State:	Beep:	Mode:	Param:		Filter:	Polarity:
- Modulo		-	_	The second			-	
🐳 Limits	DIN 1- High Speed:			0 - Off	~	0.000	1 - 10µs 💙	🗹 Active High
🐵 🕖 Current Loop	DIN 2- High Speed:	0		0 - Off	~		1 - 10µs 🗸	Active High
🕘 Velocity Loop	DIN 3:	-						-
{// Position Loop 		\bigcirc		19 - Negative Limit Sw	nt 🗹		2 - 163µs 💙	Active High
Analog Input	DIN 4:	\bigcirc		0 - Off	~		2 - 163µs 🗸	Active High
- Malog Output	DIN 5:			0 - Off	~	0.000	2 - 163µs 🗸	Active High
- Constant C	DIN 6:	\bigcirc		18 - Positive Limit Swit	ct 🗸		2 - 163µs 🗸	Active High
Enable/Disable Position Capture	DIN 7:	õ		0 - Off	~			Active High
	General Purpose Dig	ital Output:	6					
- 🎤 Performance Servo Tuner - 🛃 Slider Tuning								
🔄 Motion Tasks		State:		Mode:	Param:			
Drive Motion Status Braults and Warnings	DOUT 1:	0		0 - User (Default = 0)	~	0.000		
- Scope	DOUT 2:	õ		0 - User (Default = 0)	~			
🔚 Parameter Load/Save		I		U - Oser (Derauk - U)				
Parameters	Digital Relay:			0 - Fault Mode		 No faults. Relay clos 	sed.	
👢 Terminal								

On this screen, you can set the mode of each Digital IO on the drive using the dropdown list in the Mode column. For more information about this view see .

6.4.15 Add and Configure I/O Slices

This topic discusses adding AKT (K-Bus) and AKT2G (EtherCAT/E-Bus) I/O devices. All devices are children of the EtherCAT node in the Project tree.

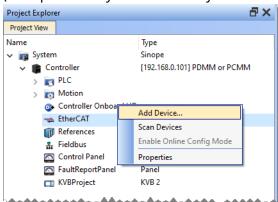
- AKT2G (E-Bus) devices are each directly under the EtherCAT node.
- AKT (K-Bus) slices are children of the coupler, which is a child of the EtherCAT node.

Α	KT2G exampl	e	AKT example	
~	⇒ EtherCAT		🗸 🔫 EtherCAT	
	🖉 Coupler_2	AKT2G-ECT-000-000	✓ Ø Coupler_1	KBus Coupler
	O DI_Slice_1	AKT2G-DNH-008-000	OlSlice_1	8 ch Digital Input
	OD_Slice_1	AKT2G-DT-008-000	🖉 DOSlice_1	8 ch Digital Output
	AKD2G_1	AKD2G Dual Axis SIL2	AKD2G_1	AKD2G Dual Axis SIL2

For local I/O, refer to Configure Onboard I/O (→ p. 186).

6.4.15.1 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
- 6. The coupler is added to the EtherCAT node.

6.4.15.2 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

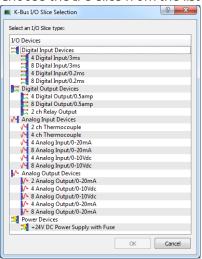


Figure 5-31: Add I/O Slice

- 4. Click OK
- 5. The device is added below the coupler.

6.4.15.3 Configure the I/O Slice

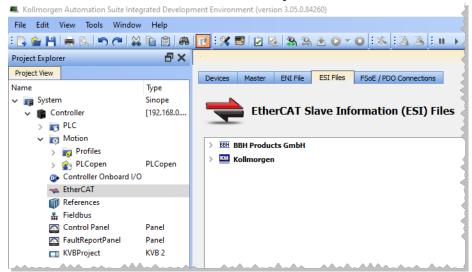
For more details, refer to Map Input and Output to Variables

6.4.16 Add & Configure Third Party Devices

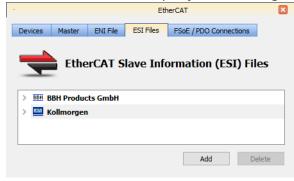
EtherCAT devices which are physically connected to the network can be scanned and discovered. EtherCAT devices may also be manually added to the EtherCAT node in the Project view. Manually added devices allow the project to be configured before connecting.

NOTE

- KAS IDE does not support third party drives.
- The KAS IDE supports third-party FSoE Masters. BBH SCU-1-EC support is integrated with the KAS IDE and is recommended. Please contact BBH for the latest ESI file that matches with the SCU-1-EC firmware version. For more details, 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.
- 1. Double-click on the EtherCAT node in the Project View and select the ESI Files tab.

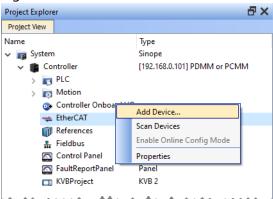


2. Add ESI files from a third-party vendor using the Add button.



3. Navigate to where you have the third-party ESI file saved on your computer and click **Open**. This will add the manufacturer and device to the list of available devices.

4. Right-click on the EtherCAT node and select Add Device... from the menu.



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.

>	🔤 Kollmorgen
>	Balluff GmbH
>	BBH BBH Products GmbH
>	ROFF Beckhoff Automation GmbH
	Bronkhorst High-Tech BV
	Danfoss Power Electronics
	EtherCAT Technology Group
	F Ferag
	nsm Festo AG & Co. KG
	 *** HMS Industrial Networks S. K. A. Schmersal GmbH + Co. KG
	🗃 K. A. Schmersal GmbH + Co. KG
	Schneider Electric
	WAGO Kontakttechnik GmbH & Co. KG
	🔀 Weidmueller Interface
>	Yaskawa Electric Corporation

6. Click Ok.

The device is added to the project tree in the EtherCAT node.

6.4.16.1 Add Modules to Third Party EtherCAT Devices

Modules may be added to third party MDP (Modular Device Profile) EtherCAT devices in the Project view. This helps to preconfigure the EtherCAT network in the project before connecting to the controller.

1. Right-click on a MDP EtherCAT device and select Add Module... from the menu.

Project Explorer		₽×
Project View		
Name	Туре	
 ▲ System ▲ Controller ▷ SPLC ▷ Motio 		PAC Controller
🖾 Contro a 🤿 Ether	ol Panel Panel	
,	evice_1 750-354	Rename
Refere	nces	Delete
品 Fieldb	us	Add Module
		Properties
And the second se	بمحادقتين بالمتحدين بالكرد	And a state of the

- 2. Select a module from the EtherCAT Module Selector dialog box.
- 3. Click OK.

The module is added to the project under the EtherCAT device.

6.4.16.1.1 Limitations

When adding modules from the Project View, there are several limitations to be aware of.

- Modules cannot be manually added to devices that are mapped to physical devices. Modules may only be added to an unmapped, manually added MDP device.
- Manually adding a MDP device will not automatically add mandatory modules. Either consult the device documentation to identify the mandatory modules and add them manually or discover the mandatory modules under the physical MDP devices, using the EtherCAT network scan.

6.4.16.2 Modules - General Properties Tab

A device may have no modules, or it may have several. Double-clicking or right-clicking on the module allows access to the General Properties tab. This tab provides information about the selected module.

🔺 🗢 EtherCAT		General Properties	4
► 🖾 AKD_1	AKD Drive		{
AKD_2	AKD Drive	Modul	e_2 (Branch 0) General Properties 🖡
Electric Electric Electric Coupler_1	KBus Coupler	Modul	e_2 (branch v) General Properties
Device_1	EK1100		1
Device_2	EL7031	Module Details	
Device_3	EL1018	Name:	Module_2
Device_4	EL2008	Description:	Branch 0
Device_5	EL3102	Type:	Branch 0
Device_6	EL4102	Class:	Cube67 Strangmodule
Device_7	BL20-E-GW-EC	Ident:	0xDCCF0001 (3704553473)
A B Device_8	750-354	Ident.	0XDCCF0001 (3704333473)
↑I Module_1	750-4xx / 750-14xx (4b in)	Slot Details	1
🗸 Module_2	750-5xx / 750-15xx (4b out)	Name:	Slot
↑I Module_3	750-4xx / 750-14xx (4b in)		
Device_9	Cube67+ BN-EC	Number:	2

Figure 5-32: Devices in the EtherCAT list and the General Properties tab of a Device's Module.

The tab is divided into two sections, Module Details and Slot Details. Module Details provides the module's basic information. Slot Details provides the module's slot information.

6.4.17 Use Pre-defined Libraries

The Libraries toolbox allows you to select the functions.

★ TIP

- The (All) category at the top enables you to see the full list of available blocks.
- You can 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).

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.

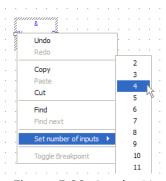


Figure 5-33: Set the Pins Number of the Block

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.

6.4.18 Create and Use Custom Libraries

You first need to create a custom library before you can use it to define a new item: function, function block or variable (for more details on library usage, refer to Use the Custom Library).

NOTE

There is a difference between **Libraries**' usage and the **Import / Export** commands related to PLC programs.

- **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.
- **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.

6.4.18.1 Create the Custom Library

- 1. In the File menu, click the New command (save your current opened project if necessary)
- 2. Select the Library application template

Choo	se an applicati	on template	
	ate Type	• 	
-	eNetwork	Library	
		S cibrary	
-			
00	ordinated Motion		
Name	Description		
	An application used to	o create a library	
Library	An application used to		
Library	An application used to	,	
Library	An application used to	,	
Library	An application used to		
Library	An application used in		

Figure 5-34: Create a Custom Library - Select the Library Template

- 3. Click the **Finish** button
- 4. Click the Save As command in the File menu
- 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, then select the type of programming language
- 9. Expand the Subprograms node and rename the new UDFB



It is the name of the variable type which 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 (in the current procedure, the prefix is: NewLib_).

- 10. Create the UDFB program (for more details, refer to Create Programs)
- 11. In the File menu, click the Save command

6.4.18.2 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 on PLC and choose Libraries in the menu
- 4. Click Add
 - NOTE

You can add as many external libraries as you want

5. Select the *.kal file already created before and click Open

Libraries	×
Libraries	
tion\~ KAS Related\KAS Project Files\Testing.kal	Add
otion\~ KAS Related\KAS Project Files\test.kal	Remove
	Close
	Cancel
< >	
Save as Relative Path	

Figure 5-35: Use a Custom Library - Select the Library

🗙 TIP

You can 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

196

7. 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 5-36: Use a Custom Library - Display the Library

8. In the Dictionary toolbox, right-click on the program and choose Add variable in the menu

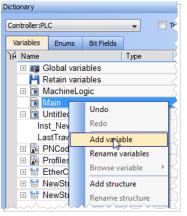


Figure 5-37: Use a Custom Library - Add a Variable

9. In the Type drop-down menu, select the type defined in the external library (it can be at the bottom of the list)

Create Variable		×
Name:	NewVar	OK
Туре:	ValidateAxisDuringMove	Cancel
Init value:	TON TP	
User Group:	TPR UNPACK8	
Tag:	ValidateAxisDuringMove	
Description:		
	🔲 External	
	🗖 Syb.	

Figure 5-38: Use a Custom Library - Select the Type

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



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 is displayed:

🛋 KAS	Libraries 🛛 🔀				
⚠	The following libraries were not found or were replaced with a newer library with a KAL extension.				
	Not found: C:\Documents and Settings\philippe.moreno.DHRMOTION\Desktop\MyLib.kal				
	Functions from missing libraries will not be available. (Libraries in italics were included in other libraries)				
	OK				

To recover the libraries, you have two options:

- Enter the new path to this library (assuming it still exists on your machine) using the library dialog (see Use the Custom Library).
- 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.

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.

It can happen for 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, you have to open LIB-4 and fix the issue (i.e. LIB-1-ND broken link), then re-open this project again.

6.4.19 Adding Cam Profiles

6.4.19.1 Create Cam Profiles

To create a cam profile, do as follows:

1. In the Project Explorer, right-click the **Profiles** item and select the **New profile** command in the contextual menu

Project View			
Name			Туре
-	ntroller PLC		Sinope [10.50.67.44] PDMM
4 🔯	Motion Profiles	_	v profile
	PipeNet Control Pan	Imp Sho	oort w compiled code
-	PDMM Onbo EtherCAT	oard I/O	
同 品	References Fieldbus		

Figure 5-39: Cam - New Profile

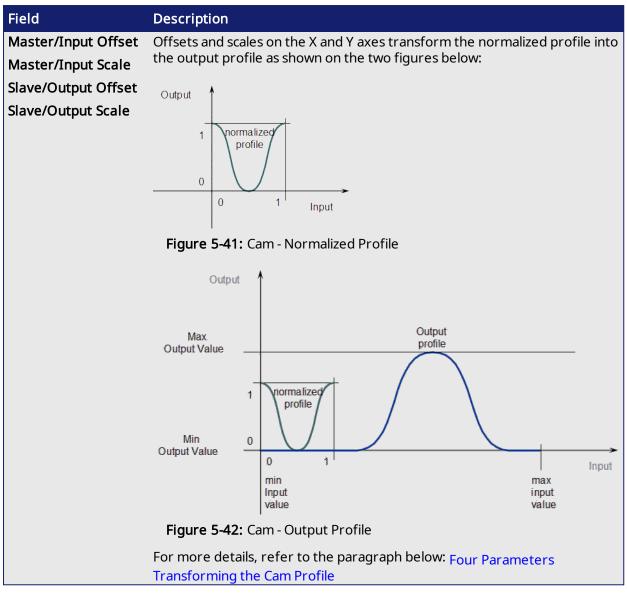
- 2. A new profile named "Profile" is created with default parameters.
- 3. Right-click on the new profile to rename it or change its properties.

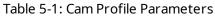
			Cam Profile Properti	es ? X
			Profile name :	Profile
			Master/Input Offset :	0
		Rename ? X	Master/Input Scale :	360
R	ename		Slave/Output Offset :	0
D	elete	My New Profile	Slave/Output Scale :	360
E	xport	OK Cancel	_	
P	roperties			OK Cancel

Figure 5-40: Cam - Define Profile Filename

4. Click on the new profile to edit it.

Field	Description
Profile name	The name of the Profile which is:
	 displayed in the Project Explorer used in the Properties of the cam Pipe Block. see "Use Cam Profiles" on page 202.





Four Parameters Transforming the Cam Profile

Master/Input offset

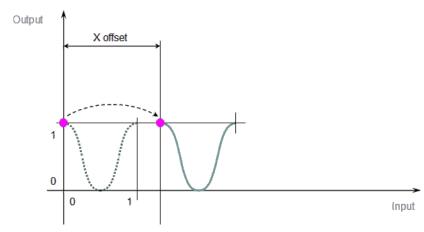


Figure 5-43: Cam Profile Transformation - Step 1

Master/Input scale

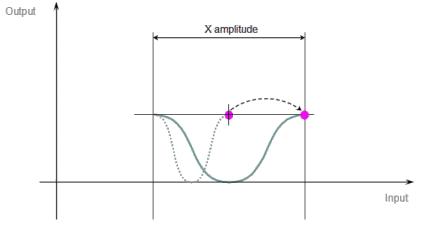


Figure 5-44: Cam Profile Transformation - Step 2

Slave/Output scale

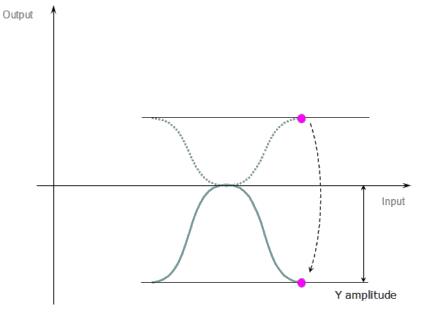


Figure 5-45: Cam Profile Transformation - Step 3

Slave/Output offset

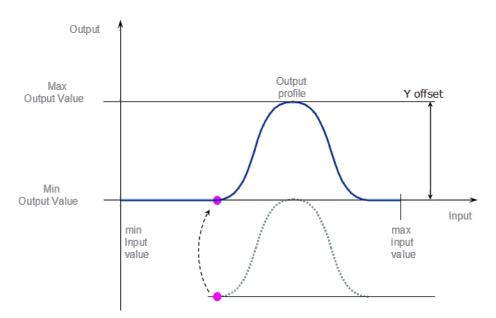


Figure 5-46: Cam Profile Transformation - Step 4

NOTE

When you change a Cam Profile property, a dialog box indicates the progression of the operation.

For more details about editing the profile, refer to Cam Profile Editor.

See also: Use Cam Profiles

6.4.19.2 Use Cam Profiles

Once defined (see "Create Cam Profiles" on page 199), 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

-	CAM : CAM1	×
	General Parameters	
	Profile Name MyProfile Output Modulo Position Profile Profile 1 Profile2	
	ОК Сапсеі	

Figure 5-47: 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.

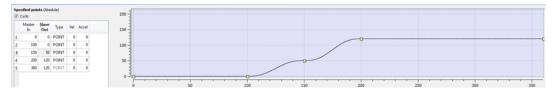
NOTE

If you change the profile's filename, do not forget to update the cam Pipe Block accordingly.

6.4.19.3 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.

For example if the following profile is switched in and the present pipe network position at the input of the cam block is 150 the position at the output of the cam block will be 50. Then as the input to the Cam block increases or decreases the Cam output value wiill increase or decrease respectively. And it will go through the modulo without a postion jump.



6.4.20 Define Scheduling

6.4.20.1 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.

	MASTER : MASTER		×
+			
A	General Parameters		
MASTER +	Sampling Period	1.0	
	Mode	Modulo	
	Modulo Position	360.0	
+	Travel Speed	1000.0	
	Acceleration	10000.0	
	Deceleration	10000.0	
	Initial Position	0.0	
		Cancel	

Figure 5-48: Set the Period of Execution

All the pipe values are computed independently of events and sequences execution.

6.4.20.2 Order of Execution

The order of execution of programs within a project is determined from the Cycle window.

Vame		Period	Phase
⊿ Hig	ah Priority		
	👔 Common 🛛 🔽	1 💠	0 ≑
▲ SFC	Section		
	👔 Main 🛛 🔽		
\triangleright	👔 UntitledSFC 🛛 🖳		0 🜲
	🚺 UntitledSFC1 🔽	1 📫	0 🜲
⊿ No	rmal Priority		
	📄 UntitledFBD 🛛	1 📫	0 ≑
	👔 UntitledST1 🛛		0 🜲
	📄 UntitledFBD1 🔽	1 ≑	0 👙

The order of execution is always

- 1. High Priority
- 2. SFC Section
- 3. Normal Priority

The order within each grouping is determined by the vertical ordering. Using the image above, the order of execution for Normal Priority programs is:

- 1. UntitledFBD
- 2. UntitledST1
- 3. UntitledFBD1

For more information see Define the PLC Cycle.

6.4.20.3 Define the PLC Cycle

The cycle specification defines the number of cycles between successive executions of the programs.

1. In the Project Explorer, expand the PLC node and right-click on the Programs item to open the contextual menu and select the **Cycle** command

Project Explorer		
Project View		
Name		Туре
	ntroller	Sinope [127.0.0.1] AKC
i i i i i i i i i i i i i i i i i i i	PLC Programs	New Program 🕨
	🕞 Subprogram:	Cycle
	S Defines Motion	Import 1
17	profiles	

Figure 5-49: Edit the Cycle

The Cycle window allows the regulation of the following parameters: Period and Phase.

Name		Period	Phase	
High Priority				
Normal Priorit		1 🗘	0 💠	
	1.1			

Figure 5-50: Define the Cycle

The cycle configuration dialog box is used to 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.
	<i>High Priority</i> means "executed before SFC", and <i>Low Priority</i> 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. Default period is "1" (the program is executed on each cycle). Giving a slower period to some programs is an easy way to give higher priority to some other programs.
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.
	Example: 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

Table 5-2: Cycle Parameters

In the **High** and **Normal** Priority sections, you can adjust the order of the programs with a drag-anddrop operation according to the expected sequence. In each section, the program on the top is executed first.

Select the program you want to set with a higher priority, then drag and drop it to the relevant priority level.

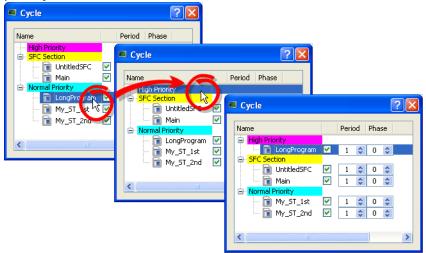


Figure 5-51: Change Priorities by Defining the Cycle

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.

Tasking Model / Scheduling and Order of Execution.

6.4.20.3.1 How to specify the duration of a cycle

This parameter is defined in EtherCAT Master Settings Tab.

6.4.20.3.2 Ensuring Variables are Exported

Program Organization Units (POUs) which contain variables (see "Map Variables to HMI" on page 172) must be compiled in order for the variable to be exported. For example, in the following set of images we see a POU (*UntitledST*) with two variables, *NewVar* and *NewVar1* and only NewVar1 is set to be exported (1). The POU, however, is not set to be executed in the Cycle dialog box (2). This will cause a compile error (3).

ÌΑ	Name	Туре	Dim.	Attrib.	Init value	User	. HMI	1				
	🗄 📷 Glo	bal variables						1	💻 Cycle			? ×
💾 Retain variables									~ -,			
	🕀 📄 Ma	chineLogic							Name		Period	Phase
	📄 Ma	in							High Priority			
	🗆 💼 Un	titledST							▲ SFC Section	-	line l	- Local
	New	Var BOO	L					١	 Main Normal Priority 	1	1 🌲	0 🌲
		Var1 BOO	L					1	UntitledST		1 🛋	0 🚔
	🗄 🔀 PN	Code								1	2) - 💌	
	🗄 🔀 Pro	ofilesCode										
	1 🐩 Etł	nerCATCode										
In	formation ar	nd Logs										
-	Local log*	Controller log	Find an	nd Replace	e Breakp	oints	Compiler Outp	out				
				DINT/RE/	AL; 11 LINT/	LREAL; 0	TIME; 22 STRI	INC	G; - CRC = d409fe92 >			
Co	ntroller:PLC:	On Line Change [MODBUS-S]: (• < CT Segment	f): UntitledS		1: Unknown	variable	symbol 3					
Co	Controller:PLC:< CT Segment = 60 byte(s) > Controller:PLC:Error(s) detected											
	Controller: PLC failed											
	Controller: Device compile failed Project compile failed											
L.	Jeer complie	Tuncu										
						Compila	tion Failed		<i>D</i>			

Figure 5-52: Example of a variable not being exported and the resulting compile error.

206

6.4.20.4 About Parent-Child relationships and execution order

The SFC Section allows for editing the period and phase values of parent and child programs. Parent and child programs follow certain 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 will follow the movement of a parent. Child programs are not allowed to move independently.
- When a child program is created or imported, it will inherit the enabled/disabled state of the parent program.
- The SFCs are executed at the set cycle period and phase. All parent programs will be executed first and then the children programs will be executed in order.

To understand the last rule, consider the following Cycle example. There are three parent programs, each with a number of child programs. All parents are executed, followed by the children, in order. The actual flow is illustrated below the image.

E Cycle				?×
Name		Period	Phase	
High Priority				
🖮 📄 Parent1	~	1 🌲	0 😂	
🗐 📄 Parent1_Child1	\checkmark	1 🌲	0 🗘	
🗔 🖬 Parent1_Child1_Child1	\checkmark	1 🌲	0 🗘	
🖃 💼 Parent1_Child2	\checkmark	1 🇘	0 🗘	
🗔 🖬 Parent1_Child2_Child1	\checkmark	1 🌲	0 🗘	
🖃 💼 Parent2	~	1 🗘	0 🗘	
📄 💼 Parent2_Child1	\checkmark	1 🗘	0 🗘	
🔤 Parent2_Child1_Child1	\checkmark	1 🌲	0 🗘	
🖻 💼 Parent3	~	1 🌲	0 🛟	
🖃 💼 Parent3_Child1	\checkmark	1 🌲	0 🗘	
🔤 🖬 Parent3_Child1_Child1	\checkmark	1 🌲	0 😂	
Normal Priority				
<				>

Parent1 Parent2 Parent3 Parent1_Child1 Parent1_Child1_Child1 Parent1_Child2 Parent1_Child2_Child1 Parent2 Child1 Parent2 Child1 Child1 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.

6.5 Running the Project

This chapter explains how to build, download and run your project.



Step	Description
1	Set the compilation options to run your project in Debug or Release modes, and 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

6.5.1 Set the Compilation Options

You can open the **PLC options** with the 🏾 icon.

If you want step-by-step debugging to be available during simulation or online testing, you need to select the **Debug** compiling mode. If step-bystep debugging is no longer required, select the **Release** compiling mode in order to give highest performance to your application.

When you incorporate additional statements (such as trace outputs) in your code, you must select the **Debug** compiling mode so that they are taken into consideration (in RELEASE mode, those statements are not included).

- For Conditional Compiling, Conditional Compiling
- For Online Changes, PLC Online Change

Code generation

There are two code compilation options, Normal and Optimized.

- Normal code allows for use of all PLC features in the KAS IDEand Runtime.
- Optimized code allows for faster PLC execution with the following limitations:
 - Debug features are disabled, including breakpoints and step-by-step debugging.
 - PLC Online Change is unavailable.
 - WAIT / WAIT_TIME are not supported.
 - Supported by AKD PDMM and PCMM controllers.

Why select Normal?

Normal code 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".

For development and debugging: Normal code allows you to set breakpoints, step through PLC code, and use online change. Additionally, as normal code is interpreted (which means that the

PLC options
Compilation Image: Code generation: Normal
Online Change Image Image
PLC Variable Locking Allow Variables to be Locked
OK Cancel

code is read by the KAS Runtime engine that then determines the instructions to run), the interpreter can apply many additional runtime checks that would be harder to implement with optimized code.

Why select Optimized?

Optimized code generation creates C code (as in the C programming language), which is compiled into native machine code, and executes directly.

Execution speed:Optimized code executes significantly faster, allowing the controller to run more complex PLC applications.

🗙 TIP

Using **Normal** code is recommended in order to take advantage of all KAS PLC features, unless the controller is unable to meet performance requirements.

6.5.1.1 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. Below is the main syntax:

```
#ifdef CONDITION
    statementsYES...
#else
    statementsNO...
#endif
```

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 facilitates the reading by coloring in gray the part of the program which is not active.

How to define conditional compiling directives?

Languages	Description
ST and IL	Directives must be entered alone on one line of text <pre>#ifdef DEF_A1_PeriodicAxis MLPhaSetPhase(PipeNetwork.PHASE1, DEF_A1_PosPeriod-A1_RefPos4); #else MLPhaSetPhase(PipeNetwork.PHASE1, DEF_A1_LinearPeriod -A1_RefPos4); #endif</pre>
FBD	Directives must be entered as the text of network breaks

Languages	Description							
FFLD	Directives must be entered as a network pragma with the गन icon.							
	In the example below, 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.							
	 Network #1 E-stop both axes when the E-stop button is pressed ##def CONDITION Network #2 Reset the axis errors of both axes when the Reset button is pressed Network #3 Close the servo loop and enable the drive when the Enable button is pressed. Open the servo loop and disable the drive when the Disable button or the E-stop button is pressed. Network #4 Close the servo loop and enable the drive when CloseLoop is high. Open the servo loop and disable the drive when CloseLoop is low. # telse Network #6 Get the Axis 1 actual position for the Control Panel to display Network #6 Get the sale position for the Control Panel to display Network #8 Read the states of the axes Network #9 In Network #9 Network #10 Montror Velocity of each Axis Network #10 Montror Velocity of each Axis Network #11 EtherCAT status word of each Axis # metwork #11 Network #12 Drive Fault observer 							
	End of Module							

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.

Running the Project

6.5.2 Compile the Application

After creating all the elements of your project, you are ready to compile it. The project must be compiled before it is simulated or downloaded to the target.

You can compile your project with the compile icon 🛃 in the toolbar (**Ctrl+B** shortcut).

The compiler reports messages in the Information and Logs toolbox (see Compiler Output tab).

No other actions are possible when the compilation is in progress.

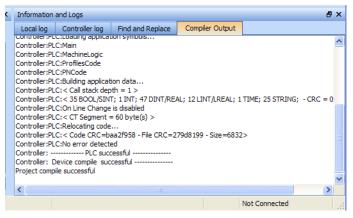


Figure 5-53: Compiler Output

🗙 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.

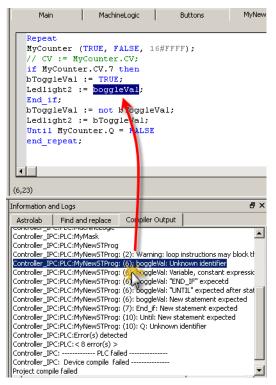


Figure 5-54: Error Location when Compiling

To locate source code, you can also use the **Find and replace** feature (for more information, refer to Find and Replace Tab)

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** for function blocks.

6.5.3 Launch KAS Simulator

If you want to simulate your application, click on the Simulator button in the toolbar.

File		View		Window						۲.
: 💽	<u>ê</u> 💾		(~ 🐰	#a 📑	: 🗶 🗖	2	S S	📩 🖸	- 0
	ct Explor				x		Simulator	ſ		~~{
Proj	ect View				 	~~~~~			~~~~	\sim

Once the program opens, you can start the simulation using the Control Panel.

Using the KAS Simulator

6.5.4 Connect to the Controller

KAS provides all the commands for controlling the target in the **Device** toolbar:

: 🛠 🖻 😥 🛃 🍇 🏂 🕑 🗸 🔘

Figure 5-55: The Device Toolbar

Icon	Description
×	Show the controller communication properties dialog (PLC options) (see details below)
	Change the controller IP address to connect with KAS Simulator. In this simulated mode, the controller properties change as shown below and the Controller Type includes the IP address 127.0.0.1.
-1-	For more details, refer to Using the KAS Simulator
	Compile project
	Compile and download the project. This downloads the application and Project Source files.
\$	Establish connection with the target controller (see possible statuses here)
\$	Close connection with the target controller
*	Download the application to the targeted controller (note that the application must not be running). For more details, refer to <mark>Download the Application</mark>
•	Start the application. It can be either: Warm Start (default mode): Retain Variables 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 with their default values. Such starts occurs from time to time but are few. To choose a Cold Start, you have to drop-down the Start button as follows: Cold Start - Use initial values for retain variables
0	Stop the application

NOTE

If the previous download failed (due to events such as the Ethernet being disconnected or the controller being turned off), you will not be able start your application. 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
- if the AKD PDMM or PCMM tries to auto-start, an E24 error is flashed on the 7-segment display.

Review the Controller logs for more information.

Ensure the Simulated device mode is active (the icon 🔤 must be selected)

To establish the connection with the target controller, click the Connect Device icon 🏹

NOTE

You need to configure the device before connecting (see Configure the Controller)

6.5.4.1 Actions to Prevent Compatibility Issues

The software versions of the KAS IDE and the KAS Runtime have to match to avoid compatibility problems.

See KAS IDE to Runtime Compatibility for more information.

🗙 TIP

The software versions of the KAS IDE and the KAS Runtime are also available in the local log messages (the level for this message is INFO).

When another KAS IDE is already connected to the controller, a warning is displayed and the connection is discarded to prevent any conflict.

6.5.4.2 Application Status Bar

The status bar provides global information about the target and the name of the running application currently stored in the device.

Text displayed with orange background means that the version of the application is different between the KAS IDE and the target.

For more details, The following table summarizes all cases for the labels of the status bar.

6.5.4.3 Message Window

Every log message has the following 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 Application)

IMPORTANT

Depending on the number of AKD 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.

6.5.5 Download the Application

- Click the *Compile and Download* button (**Ctrl + F7**) or the *Download* button. The application is compiled and downloaded or just downloaded to the controller.
 - If you enabled the "Download Project Source in the controller" option in the Controller Properties dialog box (see Configure the Controller) then the source code is also copied
 - Properties dialog box (see Configure the Controller) then the source code is also copied to the controller.
 If you disabled the "Download Project Source in the controller" option and the controller
 - has project source loaded you will be prompted with a message. The message presented will vary, based on whether your current project and the project on the controller have the same name or not.

Project Names	Message	Actions
Same name	Download Project Source The source code on the controller will be lost. Press 'OK' to continue. Press 'Save' to save the source to your local machine. Press 'Cancel' to cancel the download of the application. OK Save Cancel	OK: Source on controller is deleted. Save: Source on controller is overwritten and is also saved to "My Documents". If that source file already exists in My Documents you will be prompted to rename the source file, overwrite that file or not. If you do not rename or overwrite the file then the download to the controller will be canceled. Cancel: Download is stopped.
Different name	Download Project Source The controller contains the project file UnnamedProject.kas. Press 'OK' to replace this file with Drag_Drop.kas. Press 'Save' to save UnnamedProject.kas to your local machine. Press 'Cancel' to cancel the download of the application. OK Save Cancel	OK: Source on the controller will be replaced. Save: Source on the controller will be saved to "My Documents" as above. Cancel: Download is stopped.

- You will be prompted with an alert to connect to a controller if you try to Compile and Download without being connected.
- You will be prompted with an alert if there is not sufficient space on the controller for the application. Clearing the User Data is one method to create more space on a AKD PDMM or PCMM.

★ TIP

The versions between the KAS IDE and the KAS Runtime must be the same if you want to be able to debug your application (for example to display the animated values in the editors).

① IMPORTANT

If the IDE version differs from the runtime after compiling an application, the function blocks defined in the IDE and those implemented in the virtual machine of the runtime can possibly be different. To prevent this potential mismatch, you must compile and download your application again.

NOTE

In addition to downloading the application to the controller's flash memory, you can download the project's source code, allowing you to store the project. See File Menu for information on retrieving saved projects.

6.5.5.1 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.

Local: Scan EtherCAT:4	Controller: S	can E
	Project: Scan EtherCAT Version: Build: 4 Compile time: 2010-11-05T14:34:11	

Figure 5-56: Device Tooltip displays Version

6.5.6 Device Control

6.5.6.1 Start/stop the Device With the KAS IDE

You can start / stop the device with the buttons $igsidem{O}$ and $igsidem{O}$

With the KAS Runtime

In the KAS Runtime menu you can click the start / stop command.

Kollmorgen Automation Suite Simulator - [Axes]					
💐 File	Help				
	Start	ditor			
	Cold Start - Use initial values for retain variables				
	Stop				
	Options	1			
	Exit]			
		_	"		

Figure 5-57: Start Device with the KAS Runtime

6.5.6.2 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

6.6 Testing and Debugging 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.

6.6.1 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 mode.

In addition to the cycle-by-cycle execution mode, the debugger has a rich collection of powerful features for making step-by-step debugging in the source code of your application.

N	0	T	Е

Step-by-step debugging is available only if the project has been compiled with the **DEBUG** option. This option can be selected from the project compiling PLC options dialog box, accessible with the **Select** icon.

PLC options	
Compilation	
🔘 Release 🔘 Debug	
Code generation: Normal 👻	
Online Change	
Online Change Enabled	
PLC Variable Locking	
Allow Variables to be Locked	
OK Cancel	

Figure 5-58: PLC Options - Debug Compiling Mode

- An application compiled in **Debug** mode includes additional information for stepping. This leads to bigger code size and reduced performance.
- When debugging is finished, it is recommended to compile your application in **Release** mode to give highest performance to your application.

Step-by-step debugging is available:

- In ST and IL text programs (a step is considered as a statement)
- In FFLD programs (a step is considered as a rung)
- In FBD (a step is considered as a graphic symbol corresponding to an action)

① IMPORTANT

Step-by-step debugging is not possible in SFC programs, see About SFC for more information.

There are two possibilities for entering the step-by-step debugging mode:

• Set a breakpoint in a program (for more details, see "About Breakpoints" on page 220)

UntitledST	
Printf('Manual mode', 0, 0,	0, 0);
// Start motion MLMstRun(PipeNetwork.Master	11 , TravelSpeed 50.000000);

When you start your application and the breakpoint is reached, the execution stops at the specified location and 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 will not interrupt the current VM cycle. The current cycle will finish and execution will be paused before the beginning of the next VM cycle.

The following commands are available from the Debug Toolbar for stepping:

Icon	Description
0 +	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 will be at the beginning of the called block (if the next instruction is not a call of a function block or a sub-program, then the Step Into behaves like the Step Over)
ᠿ	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.

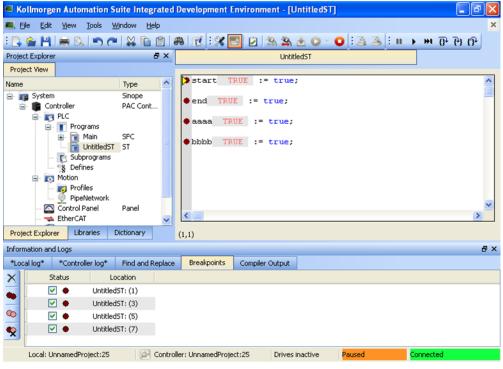
In addition to these commands, you can click at any time:

Icon	Description
HI	Execute the cycle (from the current position up to the end of the last program)
•	Restart the target in "normal" execution mode (RUN)

6.6.2 About 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 when you have chosen the DEBUG mode (for more details, Set the Compilation Options).



6.6.2.1 About Breakpoints

• Breakpoints are a marker that is 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 will not interrupt the current VM cycle. The current cycle will finish and execution will be 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 (🔘).
- Breakpoints are active only when the 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 IDE is not connected to a target
 - the IDE is connected but not running
 - the IDE is connected to a different version of the code
 - the IDE is connected to the code but a modification has been made in Edit mode.
- Breakpoints will always be applied to the target, based on their position in the editor. If a breakpoint is moved in the editor, then you reconnect to a target, the breakpoint in the target will be 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 the KAS application and are reloaded when loading a KAS application.
- See Setting, Removing, Enabling, and Disabling Breakpoints for information on working with breakpoints.
- See Breakpoints tab for information on the Breakpoints tab in the Information and Logs widget.
- Projects support a maximum of 16 breakpoints. This includes both enabled and disabled breakpoints.

NOTE

Breakpoints can significantly increase the PLC cycle time execution. This is due to the fact that the VM must evaluate the breakpoint condition at every cycle.

6.6.2.1.1 About 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.

💶 Onlii	🗉 Online Change need a running target! 🛛 🛛 🔀					
	The Online Change cannot be enabled because the runtime is paused. You must first resume the runtime!					
	Online Change has been canceled!					
	OK					

• Every breakpoint is activated if an Online Change is performed successfully.

NOTE

The breakpoints are not activated synchronously but in a reasonable time.

• All breakpoints become inactive when an Online Change is reverted.

6.6.3 Setting, Removing, Enabling, and Disabling Breakpoints

This section discusses working with breakpoints within the editor. See Breakpoints tab for information on the **Breakpoints** tab in the **Information and Logs** widget, including modifying breakpoints in bulk. For background information see About Breakpoints.

6.6.3.1 How to 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.

IF Star	rtTe	est THEN	
Ins		Undo	Ctrl+Z
Ins ELSE		Redo	Ctrl+Y
Ins		Сору	Ctrl+C
END_IF		Paste	Ctrl+V
IF Ins		Cut	Ctrl+X
Sta END IF	Find	Ctrl+F	
-		Find All	Ctrl+Shift+F
Elapse		Find Next	F3
Liupse		Add/Remove Breakpoint	F9
		Enable/Disable Breakpoint	Ctrl+F9

Figure 5-59: Setting Breakpoints

4. A Breakpoint circle is added to or remved from the left margin. The Breakpoint will be set as either active (), based on the IDE's connectivity (see About Breakpoints).

Even when you are **not** connected to the Controller, breakpoints can be placed in programs, subprograms 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.

① IMPORTANT

When you close the connection with the target, all the breakpoints are removed in the KAS Runtime.

6.6.3.1.1 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.

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.

6.6.3.2 How to Enable/Disable a Breakpoint

¹See limitation explained in paragraph below: About SFC

To enable a breakpoint, right click on an inactive breakpoint and select **Enable / Disable Breakpoint** (or Ctrl + F9). This is only available when the IDE and runtime are connected.

To disable a breakpoint, right click on an active breakpoint and select **Enable / Disable Breakpoint** (or Ctrl + F9). Selecting this option will remove the breakpoint from the runtime; the breakpoint will be remain in the editor and be changed to an inactive state (⁽⁾).

6.6.4 Printf Function

You can use the Printf function to display string in debug mode.

MacroEditor Oscilloscope	Control Panel	Controller:Motion:PipeNetwork	MachineLogic	
First Level Actions P1 N	P0 Notes			
⊙ ST/IL ○ FBD ○ LD ○ FFLD				
Printf('Manual mode'	, 0, 0, 0, 0);			~
// Start motion MLMstRun(PipeNetwork	Magter 12	ter 2 ravelSpeed	90);	
minio civan (Fipenecwort		cci z raveropeeu	50 //	
				~
	1			>
Information and Logs				8 >
Local log Controller log Find a	nd Replace Compile	er Output		
	nd Replace Compile	er Output		
Local log Controller log Find a	nd Replace Compile	er Output	Configura	tion
Filter * Time	nd Replace Compile		Configura Message	tion
Filter *	Source Level			tion
Filter * 22 (1) 13/08/2009 17:42:25 (892) 23 (1) 13/08/2009 17:42:26 (154)	Source Level Controller NFO KMS IDE INFO		Message	tion
Filter * 22 0 13/08/2009 17:42:25 (892) 23 0 13/08/2009 17:42:26 (154) 24 0 13/08/2009 17:43:37 (371)	Source Level Controller NFO KMS IDE INFO Motion INFO	Starting driver MSBusDriver Reset CndIDs of Object Databas Activation Pipes	Message	tion
Filter * 22 (1) 13/08/2009 17:42:25 (892) 23 (1) 13/08/2009 17:42:26 (154)	Source Level Controller NFO KMS IDE INFO Motion INFO Motion INFO	Starting driver MSBusDriver Reset CmdIDs of Object Databas Activation Pipes PowerON	Message	tion

Figure 5-60: Printf Function

This is a good way to trace your SFC programs. Note that you can also use the PrintMessage (Function).

How to customize output in the log window?

Raise warnings or errors icons

First column in the log window displays an information icon which can be replaced with a warning or error icon as follows:

0							
First Lev	el Actions P1 N FBD UD FFLD	P0 No	tes				
Pri	ntf('Manual mode@W	• •, •,	0, 0)	;			^
	Start motion stRun(PipeNetwork.)	Master	12,	TravelSpe	ed 90);		
							~
<							
Local log	on and Logs Controller log Find an	nd Replace	Compiler	Output			5 × 5
Filter	*					Configura	ation
	Time	Source	Level			Message	<u>^</u>
52	13/98/2009 17:47:13 (100)	Controller	INFO	Starting driver	MSBusDriver		
53 🚺	13/08/2009 17:47:13 (411)		INFO		of Object Database ok		
54 🕦	3/08/2009 17:47:40 (698)		INFO	Activation Pipes	s		
55	13/08/2009 17:47:40 (699)		INFO	PowerON			
<u><u></u></u>	13/08/2009 17:47:40 (713)	Motion	WARN	Manual mode	1		>
		1111			J		

Figure 5-61: Customizing Output for Printf Function

224

6.6.5 Soft Oscilloscope Debugging

It can be interesting to access the values of the variables of the application. It is particularly important during development and debugging.

There is a way of visualizing and changing variables via the Graphics HMI panel (see IEC 61131-3 Editor Debugging). You can also access and change variables via the Variable Dictionary (see Variable Monitoring).

However, these two methods can only access and change variables from the PLC part and not from the Motion part of the application. Furthermore, the temporal evolution of the Motion variables would 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.

For more details on Softscope description and usage, refer to Soft Oscilloscope

6.6.5.1 How to Plug Motion Variables

NOTE

The Softscope 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.

When your application is running, do the following:

- 1. Open the PipeNetwork of your Controller in the Workspace
- 2. Right-click on Gear1 to open its menu
- 3. Choose the command Plug on channel...

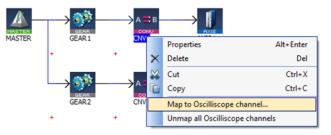


Figure 5-62: Plugging a Motion Variable

NOTE

Your application must be connected and running to plug a variable to the channel.

4. Set Channel to 1 and choose the relevant Data

Plug on chann	nel ? X
Name:	GEAR1
Type:	GEAR
Channel:	1 •
Data:	OutputValue 👻
Derivation order:	0 🗸
Derivation Period:	0.0
Ok	Cancel

Figure 5-63: Plugging a Motion Variable - Parameters

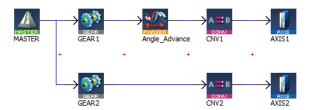
NOTE The complete list of data are only visible when your application is running

For more details on the parameters, refer to Plugging Probes, Mapping Variables

6.6.5.1.1 Usage example with the Pipe Network

The Softscope allows the recording and display of motion at points any where in a Pipe Network.

The following example shows the difference between the input and output of the Phaser Pipe Block (called AngleAdvance).



The **red** line is the input, the green line is the output and the blue line shows when the phase advance change was active.



Figure 5-64: Example of Plugging a Pipe Block

How to Plug PLC Variables

6.6.5.2 How to Plug PLC Variables

- 1. In the Variable Dictionary, right-click on the variable lastMachineSpeed to open its menu
- 2. Choose the command Plug on channel

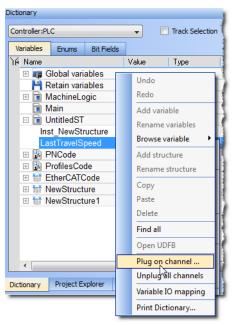


Figure 5-65: Plugging a PLC Variable

3. Set Channel to 2(because channel 1 is already plugged)

🛋 Plug on channel	? 🛛
Select a variable/member to p	olug on the channel
Group : MachineLogic	
lastMachineSpeed	
Channel:	1
Derivation Order:	1 2 3 4
Ok	Cancel

Figure 5-66: Plugging a PLC Variable - Parameters

You can start the Softscope now to see traces, as shown in the following figure:

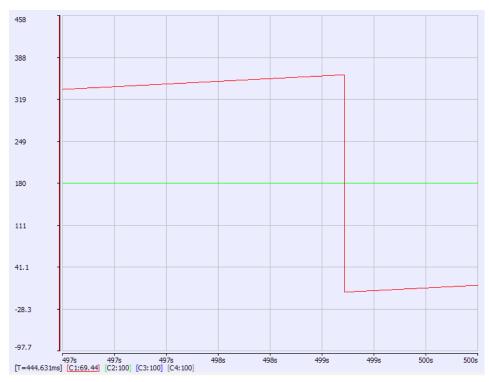


Figure 5-67: Traces Displayed with Soft Oscilloscope

★ 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.

How to Plug Motion Variables

6.6.6 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	
•	To compare project files, use the Compare Projects function from the File menu. 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.
	Local: CamSwitch_SthOrder_Parabolic:373 Controller: CamSwitch_SthOrder_Parabolic:: Drives: Unknown Running Connected

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.

Project Comparison : C:\Users\william.gaffga\AppData\Loo	cal\Kollmorgen\KAS	IDE_Instances\12868\Project\Controller\PLC <> C:/Users/willian	n.gaffga/AppData/Local/K — 🗆 🗙
1 😑 🔛 🛉 🚦 🗰 🥦 🔠 🥄			
🔺 🚡 PLC 💠 PLC	Programs\Cycle and	properties	
🔺 🚰 Programs		Programs:	Programs:
☐ Cycle and properties ☐ CAN (ST) - left only Harin (SFC) - right only ☐ Main (ST) - left only ☐ Main (ST) - left only ☐ MCFB_AKDFault_QASTI - left only ☐ ProfilesCode ☐ Bridge ▲ Si and structures ▷ ☐ Defines ☑ Binding ☑ MUDBUS ▲ Si ☑ BusDriver		<pre>> ProfilesCode_FFLD [ST] > MCFB_AKDFaultLookup [ST] Main [ST] FaultMonitor [ST] CAM [ST] UDFBs: MCFB_JOG [ST] MCFB_AKDFault_v3 [ST] ProfilesCode [ST] PLCopenCode [ST]</pre>	Main (SFC) UDFBs: ProfilesCode (ST) PLCopenCode (ST)
		-	-

Figure 5-68: Listing the Differences

The following message will be shown if you click on the compare button but the project is not present on the controller.

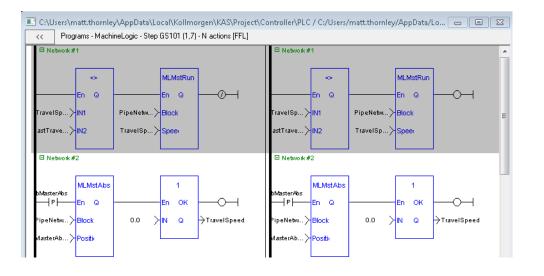


Limitation with FFLD Diagrams

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.

For example, in the following image we have a comparison of two FFLD programs.

- 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.



6.6.7 Variable Animation

When your application is running, all variables in the IEC 61131-3 Editors, in the Dictionary and in the Watch Window are animated. This means that 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.

So 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.

6.6.7.0.1 About Online Change

When Online Change is enabled, the animated values only take place when you are in Debug mode (and not edit).

6.6.7.0.2 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

6.6.7.1 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:

- 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

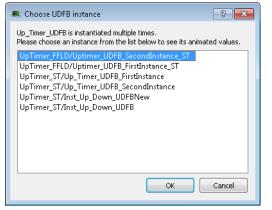
Controller:PLC	• [Track Selection				
Variables Enums Bit Fields						
IÁ Name	Value	Туре	Dim.	Attrib.	Init value User	K
🗆 📭 Global variables	(1				(2)	
TravelSpeed	0.0	LREAL			LREAL	
MasterAbsPos	0.0	LREAL			LREAL	
MasterDeltaPos	90.0	LREAL			LREAL	
MachineSpeed	0.0	LREAL			LREAL	
Axis1Status	185	DINT				1
Axis2Status	185	DINT				
MachineState	0	DINT			0	
bMasterAbs	FALSE	BOOL			FALSE	
bMasterRel	FALSE	BOOL			FALSE	1L
bEStop	FALSE	BOOL			FALSE	
bLedStatus		BOOL	[03]			
Profiles		ProfilesCode				
EtherCAT		EtherCATCode		Read Only		
Sample_Inst		NewStructure				
Inst_NewStructure1		NewStructure1				
PipeNetwork		PNCode				
Dotain variables						Þ

Figure 5-69: Variable Dictionary

6.6.7.1.1 Monitoring UDFBs

¹To 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.

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 will open the UDFB in the workspace. You will first be presented with a list of the UDFB's instances if it is instantiated more than once or declared inside of a structure.



If the UDFB is open in the Workspace before running the program it will 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 will be presented.

6.6.7.1.2 Forcing a variable

At run-time, 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**: 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.

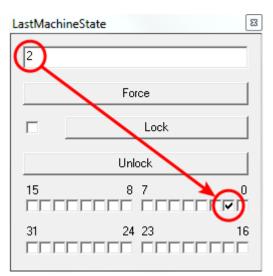


Figure 5-70: Forcing a Variable

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:

Variables	Enums	Bit Fields		
🍸 Name		Value	Туре	Dim. K
🗄 📭 G	Slobal vari	ables		
💾 R	letain vari	ables		
🗆 📄 N	1achineLo	gic		
las	tTrave	0.0	IREAL	
1.1	astTravelS	peed		E
		Force		
		Unloc	<	

If disabled (default), the Lock and Unlock buttons are not accessible:

1 7	Name	🔺 Va	ue Type	
	🗉 📷 Global vari	ables		
	Axis1Status	185	DINT	
	Axis2Status	185	DINT	
	bEStop	FAI	SE BOO	L
	bLedStatus		BOO	L (
	bMasterAbs	FAI	SE BOO	L
	bMasterRel	FAI	SE BOO	L
	EtherCAT		Ether	rCAT
	MachineSpe	ed 0.0	LREA	
	MachineSp	eed		
		Force		
		Loc	k	
8		Unlock		
8				
9	ProfilesCo EtherCATC EtherCATC			
0.00				0.00

① 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.

• Lock: 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. Note that all variables can be locked and forced at run-time.

🗙 TIP

The value of a locked variable is displayed with square brackets.

• Unlock: Remove the lock on a variable so it can be changed again by the runtime.

6.6.7.2 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 on a variable name to force or lock the variable
- Values of variables, contacts and coils are displayed in FFLD diagrams. Double-click on 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 on the variable name with the Shift key pressed to force or lock the variable

```
Repeat
MyCounter (TRUE, FALSE, 16#FFFF);
// CV := MyCounter.CV;
if MyCounter.CV; 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 5-71: Animation in Editors

See also Forcing a variable

6.6.7.3 Pipe Network Editor Debugging

See How to Plug Motion Variables.

¹To better track variables and expressions of the PLC programs in Test mode, the KAS IDE dynamically compute 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.

6.7 Managing 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.

NOTE

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, the KAS IDE proposes you to save your project before it is closed.

The Print... command allows you to create documentation containing editors' programs or diagrams.

For more details on the File menu, also refer to Menus and Toolbar Overview.

🗙 TIP

With the **Recent Projects** command in the File menu, the last four projects can be opened easily.

When editing your project, the KAS IDE has the following restrictions:

- You cannot work with several projects in parallel. You may use multiple instances of the KAS IDE as a work-around.
- Modifications that impact the project structure cannot be reversed with the **Undo** command (you have to make a backup first using the **Save As** command)
- 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)

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

6.7.1 Print

6.7.1.1 Printable Elements

The elements that you can print are:

- All PLC programs (see PLC 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)

6.7.1.2 Page Setup

This dialog enables you to define the following settings:

Page Setup tab

• Orientation:

Allows you 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:

You can 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.

Margin and Header/Footer tab

If you specify new margins or header/footer for a program, it affects the entire project when printed.

About field items used in Header/Footer

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 within your project and are applied to each printed program.

This dialog box also contains two buttons:

- Print... displays the Printer dialog box as described below
- **Print Preview** displays a printout on the screen so you can see how it looks like before printing it.

6.7.1.3 Print

This dialog enables you 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)

6.7.1.4 Print Preview

This dialog box enables you to display a printout on the screen so you can see how it looks before printing.

NOTE

Print preview limits the number of pages to display to the first 30 pages.

6.7.1.5 Print Project

A Print Project dialog displays all the items that are printable. Then you can select those you want to include in your output and click **OK**.

🔳 Print Pr	roject		? ×
-	Print Project Select the items to print		
Print to:	\\sbaprinters\SBAPR03		
	Project Items	Туре	Printer
	 PLC Programs Sefines Dictionary 		Print Preview
			Cancel

Figure 5-72: 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.

6.7.2 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

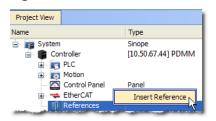


Figure 5-73: Inserting a Reference

2. Define the Name and choose a valid URL

🔳 Inse	rt Reference
Name	Tip for calling a function
URL	C:/Calling A Function.rtf
	Ok Cancel

Figure 5-74: Defining the Reference

3. You can double-click the new reference to open it in the workspace

NOTE

You can link files that are on your local machine (or to a server shared with a mapped drive) and of the following types: pdf, doc, xls, drawings, etc.. You must ensure the link is not broken if you want the KAS IDE to open it correctly.

7 KAS IDE Reference

This section provides in-depth reference to using elements of the KAS IDE interface.

7.1 Choose a Workspace Layout

7.1.1 Move Child Windows

In the integrated workspace, all child windows are integrated into a single larger application window.

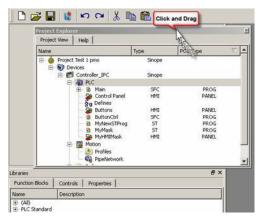
With the MDI/Tabbed workspace command in the Window menu, you can choose to display the child windows either as Tabbed Document Interface (TDI) or as Multiple Document Interface (MDI).

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.

7.1.2 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.

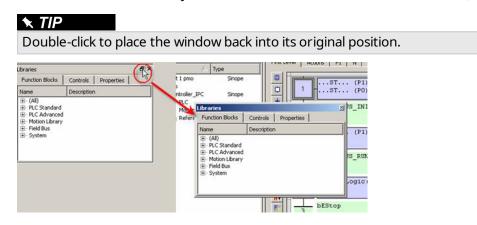


7.1.2.1 Dock Window

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).

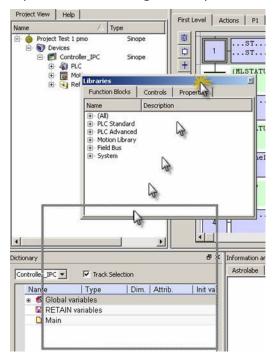


NOTE

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:



Dropped in the bottom border of the Project Explorer toolbox, then the **Libraries** toolbox is moved nearby.

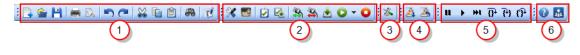
roject Explorer				8×
Project View	Help			
Name	1	Туре		1
÷ 4	es ontroller_IPC		Sinope Sinope	6 6
•	g Kererence	,		Þ
ibraries				đΧ
Function Blocks	Controls	Proper	ties	
Name	Description			
 ⊕ (All) ⊕ PLC Standard ⊕ PLC Advanced ⊕ Motion Library ⊕ Field Bus ⊕ System 				

7.2 Menus and Toolbar Overview

The KAS IDE contains the five following menus:

- File
- Edit
- Tools
- Window
- Help

...and the following toolbars:



- 1. Tools
- 2. Device
- 3. EtherCAT
- 4. Online Change
- 5. Debug
- 6. Links to the Help and Kollmorgen Developer Network (KDN)

A specific toolbar is also available for the AKD drive.

For details about icons available in the graphical PLC editors, see these sections:

- FBD toolbar
- FFLD toolbar
- SFC toolbar

7.2.1 File Menu

Command	Description
New (Ctrl + N)	Close the current project if any, and then launch the project _{wizard} to create a new one

Command	Description				
Open	Open an existing project				
(Ctrl + O)	About 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.				
	PLCAxisVersionDlg				
	The following PLCopen axes were found to be of an older version. Would you like to update them to the version supported by this software?				
	(PLCopen axes cannot be compiled until the axes are updated.)				
	Axis Name Old Version New Version PLCOpenAxis1 1.1 1.2				
	PLCOpenAxis2 1.1 1.2				
	Update				
	About 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.				
	Load Project X				
	To open an old ALS project, the ALS project and supporting files must be converted into a new KAS project file.				
	After conversion, all files in the ALS directory will be				
	replaced with a single KAS project file.				
	Do you want this conversion to continue?				
	Yes No Cancel				
	After conversion, all the files located in the folder structure are replaced with a				
	single KAS file.				
	About 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.				
	Recover Project – 🗆 X				
	Project deleteMe did not close correctly.				
	Do you want to recover it? Warning: If you choose NO, recovery cannot be performed later.				
	For more details on recovery, press the F1 key				
	Yes No Cancel				
	This file can be automatically saved periodically, so you recover recent work instead of the last manually saved file. See User Options from the Tools Menu.				
	About Corruption				
	If your project is corrupted, KAS IDE opens a pop-up window				

Command	Description
	Copen Project × File Open Error. Project This_is_a_Corrupted_Project is corrupted! OK Or you can also 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.
Save	Save the current project <i>About Auto Save</i> At compile time, an auto-save operation is performed which allows you 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 reflected 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 will open which displays the differences between the files. See Compare PLC Programs 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 you will be presented with the following error.
Close Project	Close the current project. (if changes have not been saved, a prompt is displayed first)

Command	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 and may consist of any alpha-numeric characters (a-z, 0-9, \$, &, *, }, etc,); spaces, tabs, and apostrophes are not permitted. Additionally, a company name must be provided, this should be the name of the company which owns the application source code.		
	Set password X		
	Enter new password:		
	Ok Cancel		
	The status bar has a project security status icon in the lower left corner: protecter and unprotected states.		
	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		
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		

Table 5-3: File Menu Commands

7.2.2 Edit Menu

Command	Description
Cut	Cut selected data and copy it to the clipboard
Сору	Copy selected data to the clipboard
Paste	Paste the data currently stored in the clipboard
Undo	Undo last command
	NOTE
	This action is not possible for all operations.
Redo	Redo last command
Find	Show the Find and Replace tab in the Information and Logs toolbox

Table 5-4: Edit Menu Commands

★ TIP

See Can I copy and paste or drag and drop between instances of the KAS IDE for more information on the copy/paste functions, including drag-and-drop.

NOTE

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.

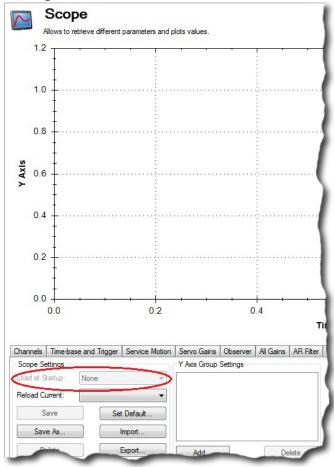
7.2.3 Tools Menu

Command	Description
Oscilloscope	Show the soft oscilloscope
Compile (Ctrl + B)	Compile the whole project
Compile & Download (Ctrl + F7)	Compile the whole project and download it to the controller.
User Options	Open the User Options dialog box. User Options AKD GUI Scope Settings Reload settings when scope page is opened Prive parameter will be loaded with the default settings. AutoSave Recovery Files Settings AutoSave Recovery Files Period (min): 5 🐑 🖉 Enabled Apply Cancel

Table 5-5: Tools Menu Commands

7.2.3.1 User Options

7.2.3.1.1 AKD GUI Scope Settings



The check box found in this dialog enables/disables the "Load at startup" option in the AKD GUI Scope "settings" tab.

7.2.3.1.2 AutoSave Recovery File Settings

This option Enables / Disables automatically saving project recovery files at a specified interval. 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? for information about recovery files.

Drives inactive	Stopped	Connected	

7.2.4 Window Menu

	a l	
Command	Shortcut	Description
MDI/Tabbed Workspace	ALT+W	Toggle the workspace between the MDI and the tabbed mode
Cascade	ALT+C	Re-arrange all workspace children windows in cascade mode
		Marbinel ooic
		 Controller: Motion: Pipel letwork Controller: Motion: Pipel letwork Controller: Motion: Pipel letwork Controleterito: Motion: Pip
Tile	ALT+T	<complex-block></complex-block>

Command	Shortcut	Description
Reset IDE Layout	Alt+R	This option will reset widgets to their default location and visibility. If an item is hidden or floating, it will be made visible and returned to its default position. The following interface items are affected by this option.
		 Project Explorer Libraries Dictionary Information and Logs Watch Window Toolbars

Table 5-6: Windows Menu Commands

7.2.5 Help Menu

Command	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. By default the Online version is used. If you prefer, you may specify which Help version to use, the Online (default) or Offline (built-in) help system.
Kollmorgen Developer Network (KDN)	Opens a web browser to the Kollmorgen Developer Network (www.kollmorgen.com/developer-network/), a community support site.
About	Show version numbers and other information about the KAS IDE Find KAS IDE Version

Table 5-7: Help Menu Commands

7.2.6 Main Toolbar

The main toolbar of the KAS IDE (Tools) contains the following icons:

Icon	Description
	Create a new project
<u>6</u>	Open an existing project
H	Save the project
-	Print the project item currently open in the workspace For more details, refer to Print
	Print preview
5	Undo
~	Redo
X	Cut
1	Сору
Ê	Paste
<i>8</i> 2	Find
1	Toggle Edit/Debug mode (enabled when the application is running)
	For more details, refer to PLC Online Change

Table 5-8: Main Toolbar Icons

7.2.7 Device Toolbar

Each icon provided in this toolbox has a brief explanation provided below in order to explain the functionality.

Icon	Description	
*	 Show the PLC options dialog which has parameters for Compilation that can be modified for the target For more details about the compilation options, see "Set the Compilation Options" on page 209 In the Online Change frame provides for allowing or forbidding Online Changes. For more details, see "PLC Online Change" on page 601 The PLC Variable Locking frame provides for allowing or forbidding PLC variables to be forced. For more details, see "Forcing a variable" on page 232. Please see the note below. 	PLC options Compilation Release Debug Code generation: Normal Online Change Online Change Enabled PLC Variable Locking Allow Variables to be Locked OK Cancel
	The PLC options selections are stored in a KAS project file and ar project is opened.	e loaded when the
2	Change the Controller IP address to connect with KAS Simulator For more details, refer to <u>Using the KAS Simulator</u>	
2	Compile project	
2	Compile and download project	
\$ \$	Establish a connection with the target Controller (for possible statuses, The following table summarizes all cases f status bar.)	or the labels of the
\$	Close connection with the target Controller	
≩ ≜	Download the application to the targeted Controller (Note that t not be running). For more details, refer to _{Download} the Application	the application must
0	Start the application. It can be either a Warm or Cold start.	
0	Stop the application	

Table 5-9: Device Toolbar Icons

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.

7.2.8 EtherCAT Toolbar

Each icon provided in this toolbox has a brief explanation provided below in order to explain the functionality.

Icon	Name	Description
1st	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. Additionally, KAS IDE displays a quick status overview of all the drives.
		The AKD Setup Screens allow functions such as enabling/disabling the drive, service motion, tuning, and a scope where you can plot up to six different parameters from the drive.
		For more details on the AKD Setup Wizard, AKD Setup Wizard
		See also "FAQ" for a potential issue when resetting the factory parameters.

Table 5-10: EtherCAT Toolbar Icons

7.2.9 Online Change Toolbar

Each icon provided in this toolbox has a brief explanation provided below to explain the functionality.

Icon	Description
Ž,	When Online Change has been activated in the PLC options; the new code is loaded even if the application is running. See also the Warning in How to Activate Online Change
3	Revert your changes done after an Online Change, and go back to the previous application

Table 5-11: Debug Toolbar Icons

7.2.10 Debug Toolbar

Each icon provided in this toolbox has a brief explanation provided below in order to explain the functionality.

Icon	Description
11	Pause application in Cycle to Cycle mode
•	Restart application in normal execution mode
HI	Execute a cycle step
<u>0</u> +	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 will be at the beginning of the called block (if the next instruction is not a call of a function block or a sub-program, then the Step Into behaves like the Step Over)
ቶ	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.

Table 5-12: Debug Toolbar Icons

7.3 Windows Standard Conventions

7.3.1 Windows Manipulation

The following standards apply to the KAS IDE windows:

- Move
- Resize
- Minimize
- Maximize
- Close (Alt+F4)

Press **Esc** to exit a pop-up window.

7.3.2 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)

7.3.3 Table Manipulation

7.3.3.1 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.

7.3.3.2 Selecting a Cell

Click a cell in the table to select it. Once selected, press **F2** to edit the value.

<mark>\A</mark> Name	Туре	Dim. Attrib.
New∀ar	BOOL	
🗆 🗋 Machin	eLogic	2,10,4

🗙 TIP

A double-click directly opens the pop-up window for editing.

7.3.3.3 Selecting a Row

When available, press the **Spacebar** to toggle the selection mode from cell to row. Then click a cell in the table to select the entire row. Once selected, press **F2** to edit the values of the row.

Create Variable		×
Name:	MasterAbsPos	OK
Туре:	LREAL 💌	Cancel
	Read Only Dim.:	
lnit value:	LREAL#0	
User Group:	•	
Tag:		
Description:		
	External	
	Syb.	

7.3.3.4 Resizing 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 the and move right or left to resize your column.

After this operation, you need to scroll horizontally to see the other columns.

7.4 Keyboard Shortcuts

List of accelerator keys sorted by context:

- Common Keyboard Shortcuts
- FBD Editor Keyboard Shortcuts
- FFLD Editor Keyboard Shortcuts
- SFC Editor Keyboard Shortcuts
- ST Editor Keyboard Shortcuts
- Graphic Editor Keyboard Shortcuts
- Table Keyboard Shortcuts
- CAM Editor

NOTE

A shortcut can be unavailable depending on the context.

7.4.1 Common Keyboard Shortcuts

Shortcut	Command
F1	Display the help
-	Collapse
+	Expand
Del	Delete
Ctrl + B	Compile the project
Ctrl + F7	Compile and Download the project
Ctrl + C	Сору
Ctrl + F	Find
Ctrl + Shift + F	Find All
Ctrl + N	New
Ctrl + O	Open
Ctrl + P	Print
Ctrl + S	Save
Ctrl + V	Paste
Ctrl + X	Cut
Ctrl + Y	Redo
Ctrl + Z	Undo
Alt + Shift + O	Open from controller
Alt + Shift + C	Compare projects
Page Up/Down	Scroll Page up/down
RETURN	Equivalent to double-click

Table 5-13: List of Common Keyboard Shortcuts

🗙 TIP

See Can I copy and paste or drag and drop between instances of the KAS IDE for more information on the copy/paste functions, including drag-and-drop.

NOTE

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.

7.4.2 FBD Editor Keyboard Shortcuts

- FBD Editor (common)
- FBD Editor (when editing)
- FBD Editor (during debug)

7.4.2.1 FBD Editor (common)

Shortcut	Command
Arrows	Scroll window
Ctrl + d	Display FBD execution order
Ctrl + page UP/DOWN	Go to previous/next section
Ctrl + mouse-wheel or + / – on the keypad	Zoom in / out
Escape	Cancel linking/resizing/dragging if selection: deselect if no selection: select mode active
Page UP/DOWN	Scroll page up/down
Return	Equivalent to double-click
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
Tab	Select next position item
Tab + shift	Select previous position item
Ctrl+F2	Toggle Bookmark (Note that you first have to select the Network header)
Shift+F2	Go to Next Bookmark
Ctrl+Shift+F2	Go to Previous Bookmark

Table 5-14: List of FBD Shortcuts

7.4.2.2 FBD Editor (when editing)

Shortcut	Command
char	Start editing a symbol (variable, constant, instance) On jump/comment/break: open dialog box to enter text
Ctrl + arrows	Align selected items
Del	Delete selection
Shift + arrows	Move selection
Shift + page UP/DOWN	Move selection (4 cells)
Spacebar Ctrl + Shift + down	Swap item style Insert blank lines at the position of the mouse

7.4.2.3 FBD Editor (during debug)

Shortcut	Command
Spacebar	Swap TRUE/FALSE Boolean value
*	Lock var
/	Unlock var

7.4.3 FFLD Editor Keyboard Shortcuts

- FFLD Editor (when editing)
- FFLD Editor (during debug)

7.4.3.1 FFLD Editor (when editing)

List of accelerator keys (sorted by action types)

7.4.3.1.1 Insert

Shortcut	Command
Ctrl+Shift+D	Insert Coil De-Energize
Ctrl+Shift+E	Insert Coil Energize
Ctrl+Shift+R	Insert Coil Reset (Unlatch)
Ctrl+Shift+S	Insert Coil Set (Latch)
Ctrl+Shift+K	Insert a positive coil to the destination cell
Ctrl+Shift+L	Insert a negative coil to the destination cell
Ctrl+Shift+C	Insert Contact NC
Ctrl+Shift+A	Insert Contact NC, Negative Transition
Ctrl+Shift+I	Insert Contact NC, Positive Transition
Ctrl+Shift+O	Insert Contact NO
Ctrl+Shift+N	Insert Contact NO, Negative Transition
Ctrl+Shift+P	Insert Contact NO, Positive Transition
Ctrl+Shift+M	Insert Data In
Ctrl+Shift+W	Insert Data In Inverted
Ctrl+Shift+Q	Insert Data Out
Ctrl+Shift+B	Insert Wire (both)
Ctrl+Shift+H	Insert Horizontal Wire
Ctrl+Shift+V	Insert Vertical Wire
Shift+C	Insert Comment
Shift+Insert	Insert Network
Ctrl+Shift+J	Insert Jump
Ctrl+Shift+T	Insert Return
Insert Key	Insert Row
F8	Insert FB

Table 5-15: List of FFLD Shortcuts

7.4.3.1.2 Trace

Shortcut	Command
Ctrl+J	Trace Horizontal Wire Left
Ctrl+K	Trace Horizontal Wire Right
Ctrl+M	Trace Vertical Wire Down
Ctrl+I	Trace Vertical Wire Up

7.4.3.1.3 Move

Shortcut	Command
Ctrl+End	Go to End of Network
Ctrl+End followed by Ctrl+End	Go to End of Ladder
Ctrl+Home or Home followed by Home	Go to Top of Network
Ctrl+Home followed by Ctrl+Home	Go to Top of Ladder
Ctrl+Page Up	Go to Previous Network
Ctrl+Page Down	Go to Next Network
Ctrl+Left Arrow or Home	Move focus to begin of row.
Ctrl+Right Arrow or End	Move focus to end of row.
Tab	Move focus cell right
Shift+Tab	Move focus cell left
Arrows	Move focus cell or scroll through ladder
Page up	Scroll 1 page up
Page Down	Scroll 1 page down

7.4.3.1.4 Select

Shortcut	Command
Shift+Arrow	Multiselect cells
Shift+left Arrow	Select current cell and one cell to left
Shift+right Arrow	Select current cell and one cell to right
Ctrl+Shift+ right Arrow or Shift+End	Select from current cell to end of line
Ctrl+Shift+ End	Select from current cell to end of network (Bottom element of network and the furthest to the right)
Ctrl+Shift+ left Arrow or Shift+Home	Select from current cell to beginning of line
Ctrl+Shift+ Home	Select from current cell to beginning of network
Shift+up Arrow	Select Cell above or below when focus is on cell.
Shift+down Arrow	Select Row above or below when focus is on left rail
Ctrl+A	Select the contents of a network/rung
Ctrl+A followed by Ctrl+A	Select the entire ladder
Shift+Page Up	Selection Page-Up
Shift+Page Down	Selection Page-Down

7.4.3.1.5 Edit

Shortcut	Command
Ctrl+C	Copy Item
Ctrl+X	Cut Item
Ctrl+V	Paste Item
Return	Equivalent to double click
Space	Change contact or coil
Ctrl+Y	Redo
Ctrl+Z	Undo
Ctrl + mouse-wheel or + / – on the keypad	Zoom in / out
Ctrl+S	Save
Esc or Shift-ESC	Close the rename widget. Exit Dialog

7.4.3.1.6 Find

Shortcut	Command
Ctrl+F	Find
Ctrl + Shift + F	Find All
F3	Find Next

7.4.3.1.7 Delete

Shortcut	Command
Delete Key	Delete cell, selection, or row
Shift+Delete	Delete Network

7.4.3.1.8 Bookmark

Shortcut	Command
Ctrl+F2	Toggle Bookmark (you must first select the Network header)
Shift+F2	Go to Next Bookmark
Ctrl+Shift+F2	Go to Previous Bookmark

7.4.3.2 FFLD Editor (during debug)

Shortcut	Command
Spacebar	Swap TRUE/FALSE Boolean value
*	Lock var
/	Unlock var

7.4.4 SFC Editor Keyboard Shortcuts

-	
Shortcut	Command
?	Show/Hide notes
arrows	Move caret
Page UP/DOWN	Scroll page up/down
Return	Equivalent to double-click
Shift + arrows	Select multiple cells
Shift + Home	Select from left to caret
Shift + Page Up/Down	Selection Page Up/down
b or B	Insert macro body
c or C	Insert convergence
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
s or S	Insert step
Spacebar	Swap item style
t or T	Insert transition
x or X	Insert the left side corner of a divergence/convergence
Ctrl + mouse-wheel or + / – on the keypad	Zoom in / out

Table 5-16: List of SFC Shortcuts

7.4.5 ST Editor Keyboard Shortcuts

- ST Editor (common)
- ST Editor (when editing)
- ST Editor (during debug)

7.4.5.1 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 (Note that you first have to 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

Table 5-17: List of ST Shortcuts

7.4.5.2 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)

7.4.5.3 ST Editor (during debug)

Shortcut	Command
*	Lock variable
1	Unlock variable
Shift + double-click	Force a variable
Spacebar	Toggle Boolean value or bring the dialog to force, lock, unlock the variable (equivalent of Shift + double click)

7.4.6 Graphic Editor Keyboard Shortcuts

Shortcut	Command
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
Arrow	Scroll
Ctrl + F2	Toggle bookmark
Ctrl + arrow	Align on main selected item
Del	Delete selection
Escape	Cancel resizing/dragging if selection: unselect if no selection: select mode active
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
Shift + F2	Go to next bookmark
Shift + Page UP/DOWN	Offset selection
Shift + Arrow	Move selection
Tab	Select next position item
Tab + shift	Select previous position item

Table 5-18: List of Graphics Editor Shortcuts

7.4.7 Table Keyboard Shortcuts

Shortcut	Command
Arrows	Move selection
Shift + Tab	Move selection to the left
Spacebar	Line selection/cell selection
Tab	Move selection to the right

Table 5-19: List of Table Shortcuts

7.5 Bookmarks

Bookmarks are used for navigating in a document. You can insert bookmarks anywhere in a document. Then you can jump from one bookmark to another with a single command for browsing the document. Bookmarks are supported in all program editors and the Variable editor.

Below are the available commands for using bookmarks:

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 text editor, a bookmark is placed on a line of text.

- In the SFC editor, a bookmark is placed on an SFC symbol (step, transition, jump...).
- 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 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.

7.6 Create Programs

This chapter provides details on the syntax, structure and use of the declarations and statements supported by the KAS IDE application language.

7.6.1 Project Structure

Structuring the application with care is important in creating your project (see Project Structure Guidelines in Advanced Topics).

7.6.2 IEC 61131-3 Editors

The KAS IDE programming environment provides language dedicated editors for:

- Sequential Function Chart (SFC)
- Function Block Diagram (FBD)
- Free Form Ladder Diagram (FFLD)
- Structure Text (ST) and Instruction List (IL)

When SFC must be used?

- 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 SFC must **not** be used?

- SFC must never be used as a decision diagram or flow chart for describing an algorithm (i.e. when you 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 also Program Limitations and the PLC Online Change feature.

7.6.3 Some Tips...

7.6.3.1 About Drag-and-Drop

The editor provides you with an ideal programming environment, including drag-and-drop features:

- Drag a variable from Dictionary and drop it into the program to insert it
- Drag a definition from Libraries 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

7.6.3.2 About Autocompletion

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. Click the relevant element and validate it with the check mark.

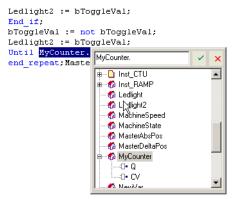


Figure 5-75: Autocompletion

Auto-completion of words

7.6.3.3 About tooltip on variable

When you leave the mouse cursor on a variable in Editors, a tooltip is displayed to give you more details on the item.

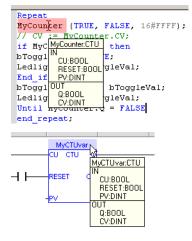


Figure 5-76: Tooltip on Variable

The header of the tooltip displays the name of the variable and its type.

7.6.3.4 About Bookmarks

See Bookmarks

7.6.4 Select Function Blocks

All available Operators, functions and function blocks are listed in the Libraries 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.

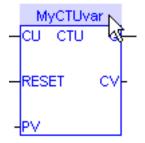
★ TIP

First 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.

7.6.5 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.

How to access a single bit of an Integer variable?

<variable>.<Bit number>(e.g. MachineState.7)

7.6.6 Drag and Drop Programming

The editor provides you with an ideal programming environment, including drag-and-drop features:

- Drag a variable from Dictionary and drop it into the program to insert it
- Drag a definition from Libraries 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

7.6.7 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. Click the relevant element and validate it with the check mark.

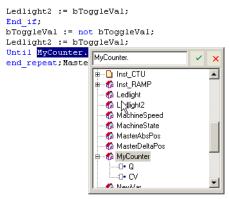


Figure 5-77: Autocompletion

Auto-completion of words

7.6.8 Tooltips on Variables

When you leave the mouse cursor on a variable in Editors, a tooltip is displayed to give you more details on the item.

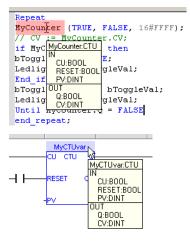


Figure 5-78: Tooltip on Variable

The header of the tooltip displays the name of the variable and its type.

7.6.9 Selecting Function Blocks

All available Operators, functions and function blocks are listed in the Libraries 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.

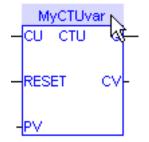
★ TIP

First 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.

7.6.10 Selecting 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.



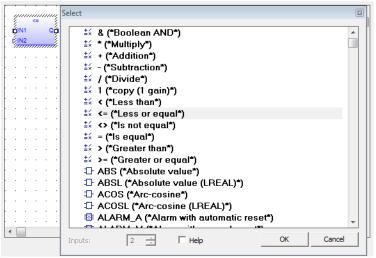
• When the variable editor is visible in the editor window, you can drag a variable from the list and drop it in the program to insert it.

How to access a single bit of an Integer variable?

```
<variable>.<Bit number> (e.g. MachineState.7)
```

7.6.11 Change Operators and Functions in FFLD and FBD Editors

Functions and operators can easily be edited and changed in the FFLD and FBD Editors. Doubleclicking on a item in the editor opens a **Select** window which lets you change the entry to another function or operator, even of another type.



- When appropriate you can select the number of inputs. This is most common for mathematical operations.
- Clicking the **Help** option enables opening 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 will need to be managed.

NOTE

Please note that this list includes numerous functions that are not valid for KAS. They are either not supported or have been deprecated.

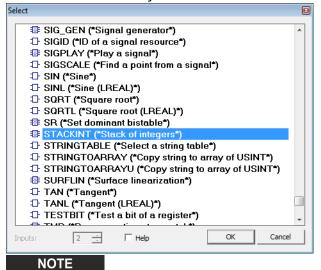
See also Change Function Blocks in FFLD and FBD Editors.

7.6.12 Change Function Blocks in FFLD and FBD Editors

• Double-clicking on the name of a function block in the FFLD or FBD Editors allows you to select a different function block from a list of other FB instances.

test OK X
▲ Variables: (all) ▲ MyFBD ■ test
Variables: (all)
☐ Local ∨ariables only ☐ Hide FB instances

• Double-clicking on the main area of a function block in the FFLD or FBD editors allows you to change to a different operator, function, or function block. Please be aware of any assigned variables which will likely need modification.

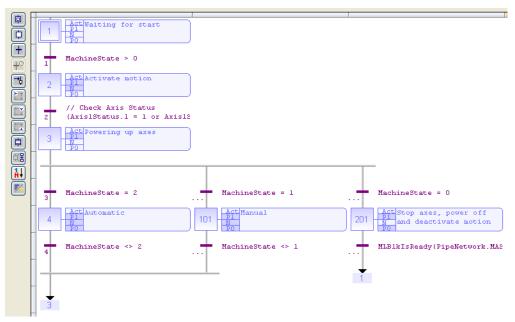


Please note that this list includes numerous functions that are not valid for KAS. They are either not supported or have been deprecated.

See also Change Operators and Functions in FFLD and FBD Editors

7.6.13 Sequential Function Chart (SFC) Editor

The SFC Editor is a powerful graphical tool that enables you to enter and manage Sequential Function Chart according to the IEC 61131-3 standard. The editor supports advanced graphic features such as drag-and-drop, so that you can freely and rapidly arrange the elements of your diagram. It also supports automatic chart formatting when inserting or deleting items, and thus enables quick input using the keyboard.



NOTE

For each step, the cells referring to P1, N and P0 actions are colored when they are defined.

SFC diagram components	Related Sections
Steps Transitions Divergences Parallel branches Jump to a step Macro steps Actions Conditions Timeout check	Using the SFC toolbar Drawing divergences Viewing the chart Printing the chart Moving or copying parts of the chart Entering macro-steps Renumbering steps and transitions Entering actions of a step Entering condition of a transition Notes for steps and transitions Bookmarks Program Limitations

🗙 TIP

- To change the number of a step, transition or jump, select it and press the Ctrl+ENTER keys.
- Hit **Spacebar** on the main corner (on the left) of a ulergence or convergence, to set either double or single horizontal line style.

7.6.13.1 Using the 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, and the chart is automatically re-arranged when a new item is inserted.

Icon	Description
向	Insert an initial step
¢	Insert a step
+	Insert a transition
→Ļ	Insert a jump to a step
●·+ ↓	Insert the main (left side) corner of a divergence or convergence
⊨±	Insert a divergence corner
L.L	Insert a convergence corner
自	Insert a macro-step
€ [1]	Insert the body of a macro-step

Table 5-20: SFC Toolbar - List of Icons

Use the following keyboard commands when an item is selected:

- ENTER: edit the level 2 of a step or transition
- Ctrl+ENTER: change the number of a step, transition or jump

The last button of the toolbar enables you to switch between possible displays:

- 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

7.6.13.2 Draw SFC divergences

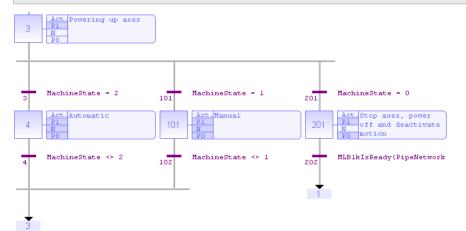
When using the SFC editor, you just need to place items in the grid. The editor calculates and draws lines automatically to link the steps, transitions, and adjusts your place in the chart.

The same method is used for drawing divergences: you just need to place the "corners" that identify divergences, convergences and branches. The editor takes care of drawing vertical and horizontal lines. Use the following buttons in the SFC toolbar:

Icon	Description
€ L.L	Insert the main (left side) corner of a divergence or convergence
⊢ ∟.⊥	Insert a divergence corner
L.T.	Insert a convergence corner

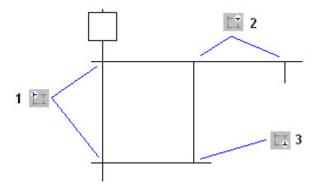
IMPORTANT

Divergences are always drawn from left to right. The first branch, on the left, contains the "corners" that identify the ulergence. It must be aligned with the preceding step or transition:



How to proceed?

- 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



Simple or double divergence lines:

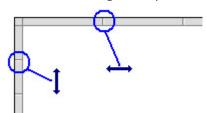
You can change the drawing of a divergence or convergence horizontal line, for drawing simple or double lines according to the SFC definition. To do this, move the selection on the main corner (on the left) and press the **Spacebar**.

7.6.13.3 View SFC charts

The chart is entered in a logical grid, and all objects are snapped to the grid. The (x,y) coordinates of the mouse cursor are displayed in the status bar. This helps you to locate errors detected by the compiler, or to align objects in the chart.

At any moment you can zoom in or out of the edited diagram using a Ctrl + mouse-wheel operation. You can also press the [+] and [-] keys of the numerical keypad to zoom the diagram in or out.

You can also 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). The last button of the toolbar enables you to switch between displays:

- 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

7.6.13.4 Move or copy SFC charts

The SFC Editor fully supports drag-and-drop for moving or copying items. To move an item, select and drag it to the desired position.

To copy an item, do the same, and just press the **Ctrl** key while dragging. 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, you can press ESCAPE to cancel the operation.

Alternatively, you can use the Copy / Cut / Paste commands from the Edit menu. The Paste action is performed at the current position.

7.6.13.5 Enter SFC macro-steps

A macro step is a special symbol that represents, within an SFC chart, a part of the chart that begins with a step and ends with a step. The body of the macro-step must be declared in the same program. The body of a macro-step begins with a special "begin" step with no link before, and ends with a special "end" step with no link after. The symbol of the macros step in the main chart has double horizontal lines.

Use the following buttons of the SFC toolbar to enter macro-steps:

- 📮 Insert a macro-step
- 🔢 Insert the body of a macro-step

() IMPORTANT

The symbol of the macro-step and the first step of its body must have the same number. Press Ctrl+ENTER when a macro-step symbol or a first step is selected to change its number.

7.6.13.6 Renumber steps and transitions

Each step or transition is identified by a number. A jump to a step is also identified by the number of the destination step. The SFC Editor allocates a new number to each step or transition inserted in the chart.

To change the number of a step, transition or jump, select it and press Ctrl+ENTER.

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".

7.6.13.7 Enter actions of a step

Actions and notes attached to a step (level 2) are entered in a separate window. To open the level 2 editing window of a step or transition, double-click on its symbol in the chart, or select it and press ENTER.

The level 2 editing window proposes five 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

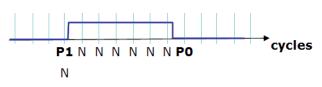


Figure 5-79: SFC Step Action Blocks

Use the tab buttons in the level 2 editing window to select a view:

```
First Level Actions P1 N P0 Notes
```

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. However, it is possible to directly enter action blocks programmed in ST together with other actions in this view. Use the following syntax for entering ST action blocks in the first pane:

ACTION (qualifier) : statements... END_ACTION;

Where qualifier is "P1", "N" or "P0".

7.6.13.8 Enter the condition of a transition

The conditions and notes attached to a transition (level 2) are entered in a separate window. To open the level 2 editing window of a step or transition, double-click on its symbol in the chart, or select it and press ENTER.

The level 2 editing window proposes 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:

Condition Notes

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.

7.6.13.9 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. Refer to the following topics for further information about the level 2 editing window:

- entering Level 2 for steps
- entering Level 2 for transitions

Notes can be displayed in the chart. The last button of the toolbar enables you to switch between possible displays:

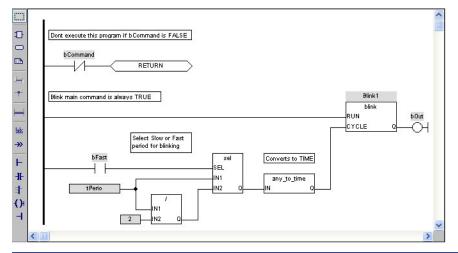
Swap between possible overviews of level 2 in the level 1 chart:

- display code of actions and conditions
- display notes attached to steps and transitions

Notes have no meaning for the execution of the chart. Entering notes for steps and transitions enables you to enhance the auto-documentation of your programs. It also provides an easy way to write and exchange specifications of an SFC program before actions and conditions are programmed.

7.6.14 Function Block Diagram (FBD) Editor

The FBD Editor is a powerful graphical tool that enables you to enter and manages Function Block Diagrams according to the IEC 61131-3 standard. The editor supports advanced graphic features such as drag-and-drop, object resizing and connection line routing features, so that you can rapidly and freely arrange the elements of your diagram. It also enables you to insert in a FBD diagram graphic elements of the FFLD (Ladder Diagram) language such as contacts and coils.



FBD diagram components

Function Variable Commen Corners Network Labels Jumps Use of ST

Related sections

- Isla also	Listen the EDD to the sec
n blocks	Using the FBD toolbar
e tags	Selecting function blocks
nt texts	Drawing connection lines
5	Selecting and entering variables and FB instances
k breaks	Viewing the diagram
	Moving or copying parts of the diagram
	Inserting an object on a line
ST instructions	Resizing objects
	Bookmarks
omponents:	
c	

FFLD cor

Contacts Coils "OR" vertical rail Power rails

NOTE

278

When a contact or a coil is selected, you can press the **Spacebar** to change its type (e.g. normal, negated, pulse)

Boolean connections can be negative at the entry of a block.

How to toggle the connection to make it negative?

1. Select the Boolean connection



Connections in FBD Programs

2. Press the Spacebar (a small circle is displayed)



Toggle Connection in FBD Programs

Execution order can be displayed.

How to display the execution order?

Data flow is executed from top left to bottom right (CTRL+d shows the execution order)

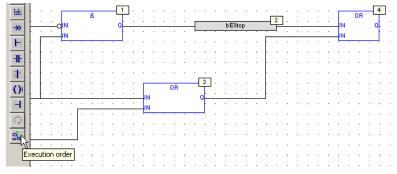


Figure 5-80: Execution Order on FBD

7.6.14.1 Using the FBD toolbar

The vertical toolbar on the left-hand side of the editor contains buttons for all available editing features. Push the desired button before using the mouse in the graphic area.

Icon	Description
	Selection : In this mode, you cannot insert any elements in the diagram. The mouse is used to select object and lines, select tag name areas, or move or copy objects in the diagram. At any moment you can press the ESCAPE key to go back to the Selection mode.
Ð	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 of the main toolbar.
	Insert variable : In this mode, the mouse is used for inserting variable tags. Variable tags can then be wired to the input and output pins of the blocks. Click in the diagram and drag the new variable to the desired position.
	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 then be selected and resized.
Ŧ	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 below). You also 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 so that you can continue drawing the connection to the desired pin and force the routing while you are drawing the line.
-9	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, as the FBD editor imposes a default routing only between two pins or user-defined corners. Corners can then be selected and moved to change the routing of existing lines.

Icon	Description
Π	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; they just help in understanding big diagrams, by splitting them into a list of networks.
<u>lab:</u>	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 below).
⇒	Insert jump : In this mode, the mouse is used to insert jump symbols in the diagram. A jump indicates that the execution must be directed to the corresponding label (having the same name as the jump symbol). Jumps are conditional instructions. They must be linked on their left-hand side to a Boolean data flow.
F	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, and represents a "TRUE" state that can be used to initiate a data flow. Power rails can then be selected and resized vertically according to the desired network height.
╢	Insert contact : In this mode, the mouse is used to insert a contact in the diagram, as in Ladder Diagrams.
ŧ	Insert "OR" rail : In this mode, the mouse is used to insert a rail that collects several Boolean data flows for an "OR" operation, in order to insert parallel contacts, as in Ladder Diagrams.
O	Insert coil : In this mode, the mouse is used to insert a coil in the diagram, as in Ladder Diagrams. It is not mandatory that a coil be connected on its right-hand side.
Η	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, and is commonly used for terminating Boolean data flows. However, it is not mandatory to connect coils to power rails. Right power rails have no meaning for the execution of the diagram.
Q	Swap item style: change the text justification
20	Execution order : Display the execution order of the elements in the diagram.

Table 5-21: FBD Toolbar - List of Icons

7.6.14.1.1 FBD variables

All variable symbols and constant expressions are entered in FBD diagrams using small boxes.

1. Press the following button in the FBD toolbar to insert a variable tag:

Insert variable: 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 on a variable tag to open the variable selection box
- 3. Either select the symbol of the desired variable or enter a constant expression.

Variables tags must then be linked to other objects such as block inputs and outputs using connection lines.

4. You can resize a variable box vertically in order to display, together with the variable name, its tag (short comment text), its description text, plus its I/O location if the variable is mapped to an I/O channel.

The variable name is always displayed at the bottom of the rectangle:

```
tag
description
%location
name
```

280

For more details on Variable Tag and Description, see "Variables tab" on page 70

7.6.14.1.2 FBD comments

Comment text area can be entered anywhere in an FBD diagram.

Press the following 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.

Double-click on the comments area to enter or change the attached text. When selected, comment texts can be resized.

NOTE

You can insert hyperlink on external files as shown below. Only TXT and BMP extensions are allowed. When the link is valid, the hyperlink is replaced with the file's content.

First Level	Actions P1	N PO	Notes	First Level	Actions P1 N
🔾 st/il 💿	FBD 🔘 FFLD			⊖ st/il ⊙	FBD 🚫 FFLD
	D:VAKD.bmp				

Figure 5-81: FBD Comments - Inserting Graphic

7.6.14.1.3 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.

Press the following 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.

You can 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, so that you can 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.

7.6.14.1.4 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. Press the following button in the FBD toolbar to insert a new break:

Insert network 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.

7.6.14.1.5 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.Press the following button in the FBD toolbar to insert a new "OR" rail:

Insert "OR" rail: In this mode, the mouse is used for inserting a rail that collects several Boolean data flows for an "OR" operation, in order to insert parallel contacts, as in Ladder Diagrams.

The "OR" rail has exactly the same meaning as an "OR" block regarding the execution of the diagram.

7.6.14.2 Draw FBD connection lines

Press this button before inserting a new line.

NOTE

As shown below, the editor enables you to terminate a connection line with a Boolean negation represented by a small circle:

(* use of a r	negated link: Q is i	IN1 AND NOT IN2 *)
	&	
In1	Q .	
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: 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 the following possible objects:

NOTE

Line is colored in red when the two linked items are not the same type.

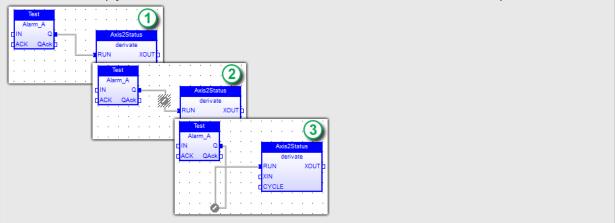
Ð	Block: Refer to the help on the block for the description of its input and output pins, and the	
	expected data types for the coherence of the diagram.	

- **Variable**: A variable can be connected on its right-hand side (to initiate a flow) or on their lefthand side to force the variable, if it is not "read only". The flow must fit the data type of the variable.
- >> Jump: a jump must be connected on its left-hand side to a Boolean data flow.
- **Left power rail**: Left power rails represent a TRUE state and can be connected to a non limited number of objects on their right-hand side.
- **Contact**: A contact must be connected on its left-hand side and on its right-hand side to Boolean data flows.
- **"OR" rail**: Such a rail collects several Boolean data flows for an "OR" operation, in order to insert parallel contacts, as in Ladder Diagrams. It may have several connections on its left-hand side and on its right-hand side. All connected data flows must be Boolean.
- () Coil: A coil must be connected on its left-hand side to a Boolean data flow. It is not mandatory that a coil be connected on its right-hand side.
- **H 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.

★ TIP

Connection lines automatically move and follow FBD elements (1) and you can also manually specify

the corners. Simply double-click on the line to show a "handle" (2) which can be repositioned (3).



7.6.14.3 Select FBD variables and instances

Press this button or press ESCAPE before any selection.

To select the name of the declared variable to be attached to a graphic symbol, you must be in "Selection" mode. Simply double-click on the tag-name gray area. The following types of object must be linked to valid symbols:

- **Block**: If it is a function block, you must specify the name of a valid declared instance of the corresponding type.
- **Variable**: Must be attached to a declared variable. Alternatively, a variable box may contain the text of a valid constant expression.
- Label: Must have a name. The name must be unique within the diagram.
- → Jump: Must have the same name as its destination label.
- **IF** Contact: Must be attached to a declared Boolean variable.
- () Coil: Must be attached to a declared Boolean variable.

Symbols of variables and instances are selected using a variable list, that can be used as the variable editor. Simply enter a symbol or constant expression in the edit box and press OK. You can also select a name in the list of declared objects, or declare a new variable by pressing the "Create" button.

For more details, Select Variables and Instances

7.6.14.4 View FBD 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 you to locate errors detected by the compiler, or to align objects in the diagram.

At any moment you can zoom in or out of the edited diagram by means of a Ctrl + mouse-wheel operation. You can also press the [+] and [-] keys of the numerical keypad to zoom the diagram in or out.

7.6.14.5 Move or copy FBD objects

Press this button or press ESCAPE before selecting objects

The FBD editor fully supports drag-and-drop for moving or copying objects. To move objects, select and drag them to the desired position.

To copy objects, you can do the same, and just press the CONTROL key 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, you can press ESCAPE to cancel the operation.

Alternatively, you can 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.

Using the keyboard

When graphic objects are selected, you can move them in the diagram by pressing the following keys:

Shift + Up	Move to the top
Shift + Down	Move to the bottom
Shift + Left	Move to left
Shift + Right	Move to right

When an object is selected, you can extend the selection by pressing the following keys:

Shift + Control + Home	Extend to the top: select all objects before the selected one
Shift + Control + End	Extend to the bottom: select all objects after the selected one

To insert or delete space in the diagram, you can simply select an object, press Shift+Ctrl+End to extend the selection, and then move selected objects up or down.

Auto alignment

When objects are selected, the following keystrokes automatically align them:

Control + Up	To the top
Control + Down	To the bottom
Control + Left	To left
Control + Right	To right

7.6.14.6 Insert FBD objects on a line

The FBD editor enables you 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, such as variable boxes, contacts and coils. This feature is mainly useful when entering elements of Ladder Diagrams. Just draw a horizontal line between left and right power rails: it is the rung. Then you can simply insert contacts and coils on the line to build the FFLD rung.

7.6.14.7 Resize FBD objects

Press this button or press ESCAPE before selecting objects.

When an object is selected, small square boxes indicate how to resize it with the mouse. Click on the small square boxes to resize the object in the desired direction.

Dont execute this program if bComm	and is FALSE
bCommand	

Not all objects can be resized. The following table indicates possible operations:

Variable	Horizontally and vertically (*)
Block	Horizontally
Labels and jumps	Horizontally

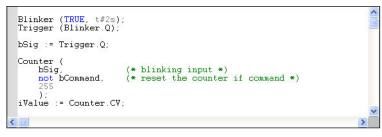
Power rails	Vertically	
OR rail	Vertically	
Comment area	In all directions	

(*) Resizing a variable box vertically enables you to display together with the variable name its tag (short comment text), its description text, plus its I/O location if the variable is mapped to an I/O channel. The variable name is always displayed at the bottom of the rectangle:

% location description tag name

7.6.15 Structured Text (ST) / Instruction List (IL) Editor

The ST / IL editor is a powerful 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.



Related sections:

Language selection Syntax coloring Autocompletion of words Drag-and-drop Active tooltips Selecting function blocks Inserting variable and FB instances symbols Reading output of a FB instance Bookmarks

★ TIP

Ctrl + Spacebar opens the Variable Editor dialog box

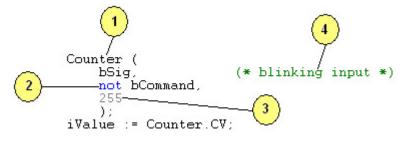
7.6.15.1 ST / IL Language selection

The KAS IDE allows you to mix ST and IL languages in textual program. ST is the default language. When you enter IL instructions, the program must be entered between "BEGIN_IL" and "END_IL" keywords, such as in the following example

```
BEGIN_IL
FFLD var1
ST var2
END_IL
```

7.6.15.2 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 the following kinds of words:



- 1. Default (identifiers, separators)
- 2. Reserved keywords of the language
- 3. Constant expressions
- 4. Comments

★ TIP

Reserved words such as function names are blue. Subprograms including Kollmorgen UDFBs are not reserved and will show as black.

7.6.15.3 Intellisense

The following features are available with Intellisense in ST and FBD programs:

NOTE

They do not apply to actions in an SFC step.

Conditional compiling coloring

Parts of conditional compiling code (declared with #ifdef pragmas) that are not validated are grayed

```
#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

Commenting the CONDITION changes the active part of the program

```
// #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

NOTE

Save your project to have the code with the correct colors.

Auto-indentation

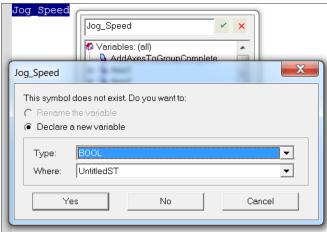
Lines are automatically indented on the left when you enter structured ST statements

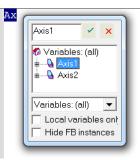
Autocompletion

7.6.15.4 Auto-completion of words

The ST / IL editor includes powerful commands for automatic completion of typed words, according to declared variables and data types. Pressing Ctrl-space (or alternatively, Ctrl-J) activates the auto-completion.

If you use auto-complete and there is no match for a variable name in the Dictionary, pressing ENTER will open the variable creation box where you can define a new variable.

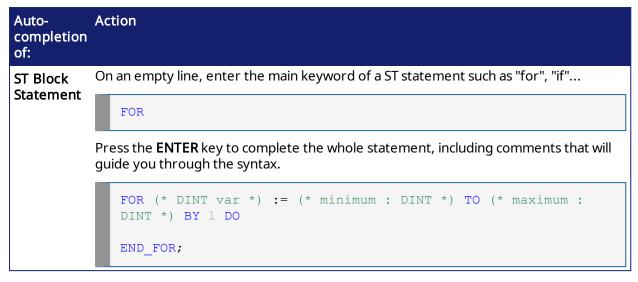




The following features are available:

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 is displayed with possible choices if several declared variable names match the typed characters.
Missing Symbols	When you press ENTER at the end of a line containing an unknown variable symbol, you are prompted to declare it immediately.
FB Member	When you type the name of a function block instance (used 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 members.
FB Call	Type the name of a function block followed by an opening parenthesis MLMstRun (
	Press the ENTER key to complete the instructions with the appropriate argument list, including comments and possibly default values so that you are guided through the list of values to be passed to the called function.
	<pre>MLMstRun((* BlockID : DINT *) , (* Speed : LREAL *)</pre>

);



7.6.15.4.1 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.

7.6.15.5 ST / IL Drag-and-drop features

The ST / IL Editor supports powerful drag-and-drop features that help you to develop and test your programs. You can:

- Drag text (words or lines) from the ST / IL editor to another application (such as 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.

7.6.15.6 How to Read Output of a MC Function Block in ST

In the following 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.

The syntax to read one of the outputs in ST for this instance is:

<FUNCTION BLOCK NAME>.<OUTPUT>

This FB has the following 5 outputs: Busy, Active, CommandAborted, Error, and Error ID. So for instance, the **Active** output has the following ST expression: A6_Inst_MC_MoveRelative.**Active**

Example 1:

UserVariable1 := A6_Inst_MC_MoveRelative.Error ;

Example 2:

```
IF A6_Inst_MC_MoveRelative.Active THEN
UserVariable2 := 1 ;
ELSE
UserVariable2 := 0 ;
END_IF;
```

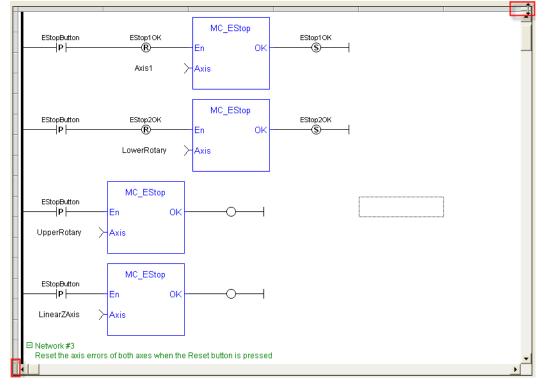
7.6.16 Free Form Ladder Diagram (FFLD) Editor

The FFLD Editor is a powerful graphical tool that enables you to enter and manage Ladder Diagrams according to the IEC 61131-3 standard. This Editor enables free drawing and arrangement of FFLD items, and supports advanced graphic features such as:

Drag-and-drop operation

Split Window capability

Allows multiple views of the same ladder program to be displayed simultaneously. You can drag the two splitters located in the vertical and horizontal scroll bars to organize your windows.



Change the size of the ladder diagram

You can drag the column separator to increase or decrease the size of the columns.

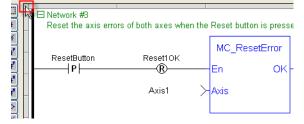


You can drag the row separator to increase or decrease the size of the rows as well as the texts.



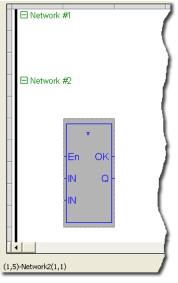
Select all the Ladder Diagram

You can click the border in the upper left corner to select the entire ladder.



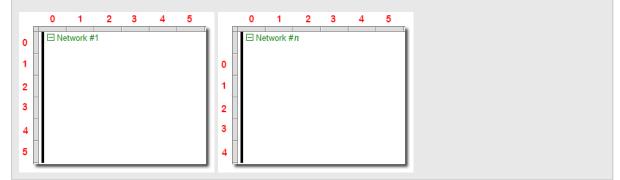
Coordinate-based position

The coordinates for a selected function are displayed at the bottom-left corner of the editor. The coordinates are shown as (absolute coordinates)-(network number).(relative coordinates) in (column, row) format.



NOTE

Relative coordinates are measured from below the Network label.



Color Coded Function

Functions and Function Blocks are color-coded as blue, while subprograms and Kollmorgen UDFBs are black.

CloseLoop	MC_Power Enable Status	PowerStat1
Axis1		
	Axis Busy	PowerBusy1 PowerErr1
bMotionEngineStarted	Enable PositiveError Enable NegativErrorID	PowerErrlD1
] Network #15 Single axis jog commands	-Axis1JogInst-	
	MCFB_Jog	
Axis1 Axis1JogPlus Axis1JogMinus	AxisID JogPlus	
	Jogminus	

FFLD diagram components	Related sections
Networks Power rail and lines Contacts and coils Function blocks Data In/Out Jumps and RETURN	Using the FFLD toolbar Selection grid Moving and copying items Run-time

X IIP

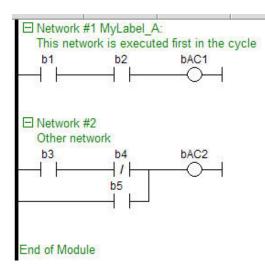
When a contact or a coil is selected, you can press the **Spacebar** to change its type (normal, negated, pulse)

<u>Networks</u>

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 the following pieces of information:

- The number of the network (from 1 to N)
- (Optional) A label name used as a target for jump operations
- (Optional) A directive for conditional compiling
- (Optional) A multiple line description (comment)

No item can be put on a network header row. 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.



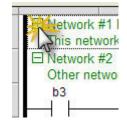
Double-click on the header of a network to enter its label, directive (sometimes called pragma) and description. Network headers are green, but they became blue when a directive is defined (see also Conditional Compiling for more details).

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 its whole contents.

There cannot be two networks having the same label in a program. If such a situation occurs in the case of a copy operation, you will be prompted to either specify another label name for the new network, or remove the label on the new network.

You can also collapse/expand a network with the minus/plus sign located next to the Network number in the header.



<u>Run-time</u>

When your application is running, you can force and lock a variable or a contact directly in the editor with a double-click operation. For more information, see "Forcing a variable" on page 232.

NOTE

In FFLD, when a function, function block or UDFB is not connected on the left, then it is ignored (removed at compiling time).

7.6.16.1 Using the FFLD Editor

This section describes the Toolbar icons and Contextual Menu of the FFLD Editor.

For FFLD accelerator keys, refer to FFLD Editor Keyboard Shortcuts

7.6.16.1.1 Toolbar

294

The vertical toolbar on the left-hand side of the Free Form Ladder editor contains buttons for inserting items in the diagrams. Items are inserted at the current position in the diagram.

ICON	Shortcut	Description
		Mode selection
HE	Ctrl+Shift+O	Insert a contact to the destination cell
¥	Ctrl+Shift+C	Insert an inverted contact to the destination cell
$\exists F_{i}^{\mathbf{b}}$	Ctrl+Shift+P	Insert a Pulse contact to the destination cell
_ж ,	Ctrl+Shift+I	Insert an inverted Pulse contact to the destination cell
-HEM	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
Þ	Ctrl+Shift+D	Insert an inverted coil to the destination cell
-0 ⁵	Ctrl+Shift+S	Insert a set coil to the destination cell
-0 ^R	Ctrl+Shift+R	Insert a reset coil to the destination cell
- O P	Ctrl+Shift+K	Insert a positive coil to the destination cell
- 0 "	Ctrl+Shift+L	Insert a negative coil to the destination cell
—	Ctrl+Shift+H	Trace a horizontal line to the destination cell
1	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
<r>></r>	Ctrl+Shift+T	Insert a return
in >	Ctrl+Shift+F	Insert a data in
•	F8	Insert a function block
€out	Ctrl+Shift+Q	Insert a data out
Q	Spacebar	Swap item style of the current cell for a contact or coil
Lob		Define a network label, pragma, and/or comment
 com	Shift+C	Define a comment in a cell. Click the button, then select the cell you wish to add a comment to. For more information, see Contextual Menu for a description of cell comments.

Table 5-22: FFLD Toolbar - List of Icons

7.6.16.1.2 Contextual Menu

A right-click in the FFLD workspace gives you access to the following commands:

• Insert Comment

Comments may be added to cells. If the text is long it can display beyond the bounds of the cell (to the right, and below) until another item is found.

È,			Enable
	Insert a comment here. The Browse 🛛 🔺 📗	,	1
	button below lets you choose an image	Axis1 >	Axis
	to use as a comment.	This is a long comment. The te	
		This is a long comment. The te	
		This is a long comment. The te	
HI	- T		1

- Insert Network
- Insert Row
- Delete Cell

- Delete Network
- Delete Row

7.6.16.2 Power rail and lines

Vertical power rails are used in FFLD language to represent the limits of a rung.

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 operation.

You can insert a segment of horizontal line at any location in order to freely draw flow lines. The "vertical line" button enables you to set or remove (toggle) a segment of vertical line on the right of the selected cell.

7.6.16.3 Contacts and coils

The table below contains a list of the contact and coil types available:

Contacts	Coils
Normally Open - -	Energize -()-
Normally Closed - / -	De-energize -(/)-
Positive Transition - P -	Set (Latch) -(S)-
Negative Transition - N -	Reset (Unlatch) -(R)-
Normally closed positive transition - /P -	Positive transition sensing coil -(P)-
Normally closed negative transition - /N -	Negative transition sensing coil -(N)-

7.6.16.4 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, and means that 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 the case of a function block, the instance of the block must be specified and is shown on the top of the block. Double-click on the top of the block to select the instance. You can also 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 also be specified using specific data in/out items (see below).

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, then it is ignored (removed at compiling time). It is the case for Functions only - **not** for function blocks.

A Function is just part of an expression (same as a contact) and is just intended to provide a result. In case of 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.

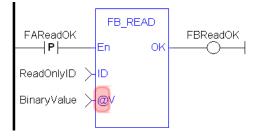
7.6.16.5 Data In/Out

The "data in" and "data out" items are used to initiate a flow (line) with the value of a variable, or to force a variable on output with the value of a flow:

VarIn>- ->VarOut

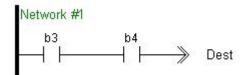
When used with a block, the "data in" and "data out" items can be put close to the block, without any line in-between to connect a variable to an input or output of the block.

In the following example, the **@** symbol in front of the V variable indicates it is used for input and output.



7.6.16.6 Jumps and RETURN

A jump to a label branches the execution of the program after the specified label. In FFLD language, the ">>" symbol (followed by the target label name) is used as a coil at the end of a rung.



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.

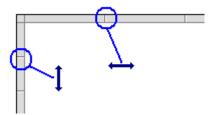
NOTE

The special "<RETURN>" destination specifies a jump to the end of the program.

7.6.16.7 Selection grid

The diagram is entered in a logical grid, and all objects are snapped to the grid.

At any moment you can zoom in or out of the edited diagram (for shortcuts about zooming, see "Using the FFLD Editor" on page 294). You can also drag the separation lines in vertical and horizontal rulers to freely resize the cells of the grid:



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 and its coordinates (row, column) are displayed at the bottom left-hand corner of the editor.

If you click on the current position, then the cell is drawn as gray, meaning that it can be dragged somewhere else in the diagram (see below). You can also select multiple cells with the mouse, or use the arrows of the keyboard with the SHIFT key pressed.

Click on the power rail (gray ruler at the left border) to select a full row.

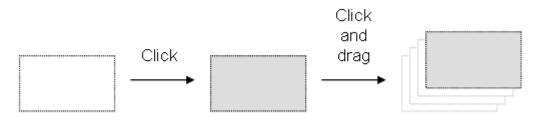
Other selection commands are available from the keyboard:

Keystroke	Description
Home	moves the caret to the left of the line if pressed again, moves the caret to the head of the network
End	moves the caret to the end of the line if pressed again, moves the caret to the end of the network
Ctrl + Page Up Down	moves the caret to previous or next network header
Ctrl + Home/End twice	moves the caret to the beginning or the end of the program
Ctrl + A	selects the whole network if pressed again, selects the whole program
Page Up / Down	scroll 1 page
Shift-Page Up / Down	selection page up or down
Return	equivalent to a double-click
Space	change contact or coil
Tab	move focus cell right
Shift-Tab	move focus cell left
Arrows	move focus cell or scroll through ladder
Shift-Arrow	multi-select cells
Ctrl + F	performs a Search and Replace (+ add hyperlink on the topic) within the whole program
Ctrl-Shift-F2	go to previous bookmark
Esc / Shift-Esc	close the rename widget

7.6.16.8 Move and copy items

298

When you click on the current position, then the cell is drawn as gray, meaning that it can be copied or moved. Click again on the selection to drag it with the mouse.



Dragging the selected items moves them to the specified location. If you press the **CTRL** key while dragging, then items are copied (for shortcuts, Using the FFLD Editor).

To move a function block, you must select it entirely.

If you move or copy items on a non-empty area, you will be 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 in order to make the required extra space for moved items.

The "Copy / Cut / Paste" commands can also 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).

7.6.16.9 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 you to locat errors detected by the compiler, or to align objects in the diagram.

At any moment you can zoom in or out of the edited diagram by means of a Ctrl + mouse-wheel operation. You also can press the [+] and [-] keys of the numerical keypad to zoom the diagram in or out.

You can also 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 pieces of information about the variable:

- its tag (short description)

- its description text

- its I/O name (%...) if the variable has a user-defined name.

7.7 Create Variables

★ TIP

As a naming convention for variables, it is recommended to use the initial to reflect the variable type (e.g. Boolean with **b**; long integer with **L**)

7.7.1 Use the Dictionary

For explanations on dictionary usage, including how to create and rename variables, Dictionary Contextual Menu

7.7.2 Create Variables from the Editors

You can create variables directly from the IEC 61131-3 editors, as follows:

FBD editor

1. Click the dedicated button

⊖ st/IL i	• 1	FBD	0	LC	>									
Ð			÷	÷		÷	·	÷	·		·		÷	÷
	÷		•	•	•		•	•	•		÷	÷	÷	÷
Add		i - h			Ne	w∖∕a	ır1							
AUU	var	IdDi	e							_				•
Her 1	•		·	·	·	·	·	·	·	·	·	·	·	·
			•		·		·					·		·

Figure 5-82: Add Variable in FBD Editor

- 2. Click a location in the editor (or double-click the variable if it is already created)
- 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)

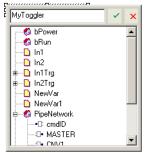


Figure 5-83: Define Variable Name in FBD Editor

- 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

MyToggler							
This symbol does not exist. Do you want to:							
C Rename		l					
O Declare	a new variable	l					
Туре:	BOOL	l					
Where:	MyFBDProg 💌	I					
Ye	MyFBDProg GLOBAL RETAIN						

Figure 5-84: Define Variable Type in FBD Editor

FBD variables

FFLD editor

1. Double-click the in or out pins of the function block

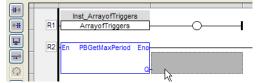


Figure 5-85: Add a Variable in the FFLD Editor

2. Edit the name (or select an existing variable within the list which is already filtered according to their relevant data type)

En	PBGetMaxPeriod	Eno		Ç
		Q.	MyNewPeriod	×
			PreviousPeriod	_~~
			1	

Figure 5-86: 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)
 - Specify where it is defined

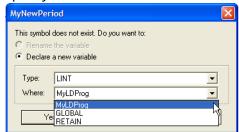


Figure 5-87: Define a Variable Type in the FFLD Editor

7.7.3 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)

16 Name	Туре	Dim. Attrib.
NewVar	BOOL	
🗆 🗋 Mac	hineLogic	2,10,4
NewVar	BOOL	[01,09,03]

Figure 5-88: Declare an Array for an Internal Variable

Arrays

7.7.4 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 Structures

Declare the structure

- 1. Right-click in the Dictionary to open the menu
- 2. Select the Add structure command

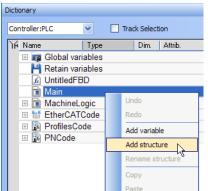


Figure 5-89: Add a Complex Structure

3. Right-click on the newly created structure and select the Rename structure command

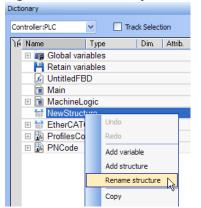


Figure 5-90: Rename Complex Structure

4. Right-click on the new structure and select the Add variable command

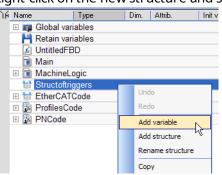


Figure 5-91: Add Variable to a Complex Structure

- 5. Expand the new structure
- 6. Double-click on the new nested variable and define its name and type

-	Tructof	triggers
	trigger1	BOOL
	Trigger2	BOOL

7. Repeat steps 5 and 7 to add all the requested variables

Create an instance of the structure

When finalized, you can drag-and-drop the structure from the library in the **(Project)** node to a program just like any other function block. A new instance is automatically created.

1. Select the new structure and move it with a drag-and-drop operation to the program declaration within the Dictionary

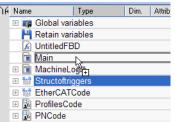
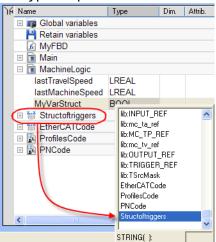


Figure 5-92: Create an Instance of the Structure

2. You can also add a variable in the Dictionary with the **Add variable** command. Then doubleclick on the new variable to define its type by selecting the structure type which is displayed in the Type drop-down menu.



3. Then you can drag this new instance and drop it in your program like any other variable

7.7.5 Variable Editor

You can edit variables directly from each IEC 61131-3 editor.

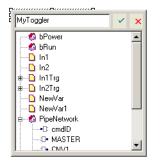


Figure 5-93: Edit the Name in the Variable Editor

★ 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

MyToggler	2	X
C Rename	does not exist. Do you want to: the variable a new variable	
Type:	BOOL	
Where:	MyFBDProg 🗸	
Ye	MyFBDProg GLOBAL RETAIN	1

Figure 5-94: Define Type and Scope of the Variable

7.7.6 Editing Variables as Text

As an alternative for editing variables, they may be edited as text. Text editing applies to all the variables of a group. Selecting "Edit variables as text" from the right mouse menu opens a dialog box which contains all of the variables in the group. From here, variables may be added, deleted, or edited using the IEC61131-3 format.

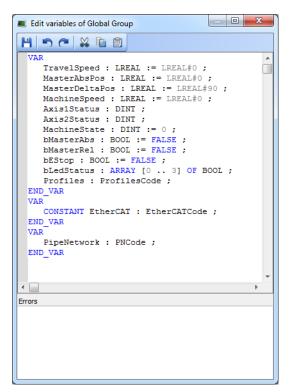


Figure 5-95: Editing variables as text

The editing dialog contains the basic text-editing functions, Save, Undo, Redo, Cut, Copy, and Paste. Upon saving changes, 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.

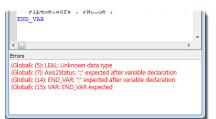


Figure 5-96: Errors caused by editing variables.

Each error includes the group name, the line number where the error occurs, and what the error is. Double-clicking on an entry will put the cursor at the point of the error.

IMPORTANT

If variables that are used in PDO mapping are renamed or deleted, the mapping will be lost upon saving changes. If this occurs, you will be informed with an alert.

For more information on editing variables as text, see "Editing variables as text using IEC 61131-3 syntax" on page 83.

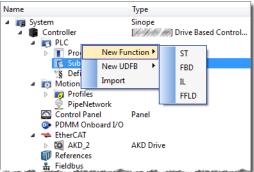
7.8 Create Functions and Function Blocks

For explanation about the difference between functions and function blocks, refer to Program Organization Units.

7.8.1 Declare Functions or Function Blocks

This section explains how to create a new function or UDFB.

1. Expand the PLC node in the Project Explorer, and right-click on the Subprograms item and select the action you wish to perform (i.e., *New Function* or *New UDFB*). The New Program dialog box will open.



2. Enter the program name (MyFBD, for example), and click OK.

💻 New program	? 🗙
MyFBD	
Ok	Cancel

3. The *Program Properties* dialog box opens. This is where you will enter the input and output parameters for the function or UDFB.

Program properties	×
Parameters	
Inputs	
Input1 : BOOL Input2 : BOOL	<u>E</u> dit
	Delete
	Move Up
	Move <u>D</u> own
	> OUT
) Outputs:	
Output 1 : DINT Output 2 : BOOL	
ОК	Cancel Help

4. Clicking on the editable space ("...") in Inputs or Outputs enables the **Edit** button. Clicking the Edit button or double-clicking on the editable space opens the *Parameters* dialog which lets

you add variables for Inputs and Outputs.

Parameters			×
Name:	Input2		ОК
Туре:	BOOL	•	Cancel
	IN_OUT		
Dimensions:			
Description:			

Item	Description
Edit	Press this button to change the definition of the selected parameter. Pressing this button when the last line () of the list is selected lets you add a new parameter.
Delete	Press this button to remove the selected parameter.
Move Up/Down	Press these buttons to move the selected parameter in the list to arrange the order of parameters. The order is very important as it defines the calling prototype of the UDFB or sub-program.
Swap	Pressing this item moves the selected parameter between the Inputs and Outputs lists. When an item is selected the button's text changes to show where the parameter will be moved.

NOTE

New UDFBs are added to the (Project) node in the Library toolbox

7.8.2 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 let you add the following:

- Input1 Parameter
- Output2 Parameter
- Private³ Variable

Dictionary						₽×
Controller:PLC -		Track Selection				
Variables Enums Bit Fields						
∖A Name	Туре	Dim. Attrib.		Init value	User	KVE
🗄 📷 Global varia	ables					
💾 Retain varia	ables					
🗉 📄 MachineLo	gic					
Main	•					
🗄 🖟 MyUDFB	_		_			
🗉 🛃 PNCode		Undo				
🗉 🙀 ProfilesCod	le	Redo				
🗄 👹 EtherCATC	ode	Add variable	•	N Input p	arameter	
		Rename variables		Outpu	t paramet	er
		Browse variable	Þ		variable	
		Add structure				
and a subtract of the second second		Reparte structure	-			

Figure 5-97: Parameters and Private Variables

¹Externally supplied, not modifiable within the organization unit

2Supplied by the organization unit to external entities

3Supplied by external entities - can be modified within organization unit

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 type

7.8.3 Finalize Functions or Function Blocks

Double-click the item in the Project Explorer to open and complete it in its corresponding editor.

7.8.4 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.

Libraries			₽×	Ether	CAT: /	AKD.	Drive	2	Ether
Function Blocks	Controls	Properties							
Name	C	escription	^						
🖳 🌆 Profile					· ·		• •		· ·
In PNCode				- ·		· · ·	·	· ·	
🌇 MyFBD		8		- ·		N		,	
Structoftriggers							. htz		
PLC Standard							. <u>"</u> 14	<u>.</u>	
PLC Advanced									
Motion/Pine Net	twork		~	-9-					

Figure 5-98: 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

7.9 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). Defines are used to determine which parts of a program's code will be compiled using an *ifdef* statement (see Conditional Compiling). This creates more efficient code for a given machine type. For example, you can write a program that covers many machine types but compile for a specific machine with more efficient code.

Defined constants have three levels of scope:

Level	Scope
Internal Defines	All the projects present on your machine
Global Defines	All the programs within your project. These are user-defined.
Local definitions	Only the current program currently open

NOTE

Important! To guarantee precision when evaluating the expression, you need to pay special attention to the data types of variables used in the expression. For example when mixing **LREAL** and **REAL**, the precision will be that of REAL.

KAS IDE supports the definition of aliases. An alias is a unique identifier that can be used in programs to replace other text. See Alias Definitions for more information.

7.9.1 Internal Defines

These are pre-defined, common constant definitions which are declared for all projects.

IMPORTANT

To ensure consistency, you should not modify these declarations.

To see the set of declarations currently installed on your machine, you can view the file (named: *lib.eqv*) located under: C:\Program Files\Kollmorgen\Kollmorgen Automation Suite\Astrolabe\Bin\HwDef (the folder location differs if you chose another location when installing KAS).

Below is a an example of predefined constants that you may find in your system.

```
#define MLPN_CREATE_OBJECTS 1 (* Creation of blocks and pipes *)
#define MLPN_ACTIVATE 2 (*Activation of pipes*)
#define MLPN_CONNECT 3 (*Connections from convertors to axes*)
#define MLPN_POWER_ON 4 (*Power ON of axes*)
#define MLPN_POWER_OFF 5 (*Power OFF of axes*)
#define MLPN_DEACTIVATE 6 (*Deactivation of pipes*)
#define MLSTATUS_NOT_INITIALISED 0 (*Motion not initialised*)
#define MLSTATUS_RUNNING 1 (*Motion is running*)
#define MLSTATUS_STOPPED 2 (*Motion is stopped*)
#define MLSTATUS_INITIALISED 2 (*--DEPRECATED-- Motion is initialised*)
#define MLPR_CREATE_PROFILES 1 (* Creation and initialization of profiles
*)
#define MLFI_FIRST 0 (* ID of the first FastInput of an axis *)
#define MLFI_SECOND 1 (* ID of the second FastInput of an axis *)
```

#define MLFI DISABLE 0 (* configures a FastInput as disabled *) #define MLFI_RISING_EDGE 1 (* FastInput is sensible to rising edges *) #define MLFI FALLING EDGE 2 (* FastInput is sensible to falling edges *) #define PI 3.1415926535897932 #define EC POSITION DEMAND VALUE 10000 #define EC_VELOCITY DEMAND VALUE 10001 #define EC_TORQUE_DEMAND_VALUE 10002 #define EC ADDITIVE TORQUE VALUE 10003 #define EC MAX_TORQUE 10004 #define EC OPERATION MODE 10005 #define EC CONTROL WORD 10006 #define EC LATCH CONTROL WORD 10007 #define EC ANALOG OUTPUT 10009

NOTE

The exact contents of the list depend on the version of the KAS IDE.

7.9.2 Global Defines

Global Defines are user-generated constants to be used in a program. Global Defines let you 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 from the Project Explorer toolbox under **PLC**.

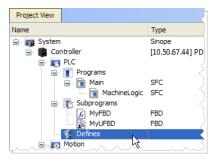


Figure 5-99: Global Defines

Double-click the **Defines** item to open your global definitions file (named: *appli.eqv*) in a text editor as follows:

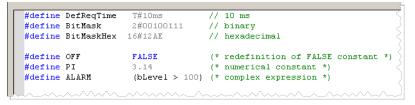


Figure 5-100: Edit the Global Definitions

Each definition must be entered on one line of text according to the following syntax:

#define Identifier Equivalence (* comments *)

You may use a definition within the contents of another definition. The definition used in the second must be declared first. See example below:

```
#define PI 3.14
#define TWOPI (PI * 2.0)
```

7.9.3 Local definitions

Local definitions are user-created defines that are being 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 to avoid misunderstandings when reading or debugging a program.

7.10 Map Input and Output to Variables

This procedure describes how to map EtherCAT motion bus I/O or AKD PDMM/PCMM Onboard I/O to PLC variables.

NOTE

This operation is disabled when the controller is running.

For more information see:

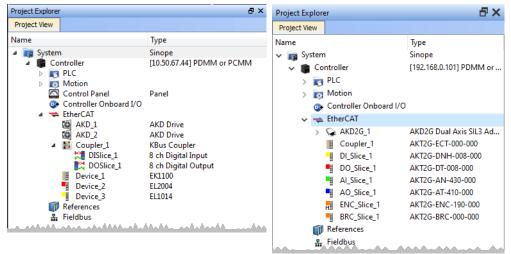
- Map I/O from the Project Explorer
- PLC Variable Creation Wizard
- PLC Variable Selector
- Analog I/O Parameters

7.10.1 Map I/O from the Project Explorer

★ TIP

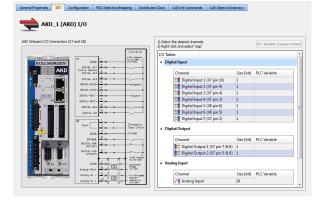
For information on mapping I/O with third-party devices, please see PDO Selection/Mapping Tab.

1. In the Project Explorer, expand the Controller node and the EtherCAT node(s) to access your devices.



AKT (K-Bus) devicesAKT2G (E-Bus) devices2. Depending upon the device, the properties are accessed in slightly different ways.

AKD or AKD2G: Double-click on the drive and select the **I/O** tab



AKD PDMM or PCMM Double-click Controller Onboard I/O

Double-click the I/O Slice under EtherCAT (E-Bus) or in the Coupler

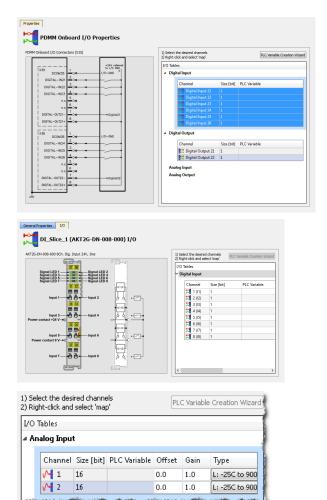
thermocouples, you also have to

define offset and gain parameters.

For Slice Analog I/O and

I/O Slice:

entry (K-Bus).



Safety I/O Slice: Double-click the Safety I/O Slice under EtherCAT (E-Bus).

Safe Digital Inputs and Output channels are clearly identifiable by a yellow table in the **I/O** tab.

Digital output logic on the safety slices / devices is typically disabled by default from the standard PLC. See the safety digital output documentation and the AKT2G-SDO-

004-000Safety terminal with 4 digital fail-safe outputs table for details.

- 3. There are several ways to map variables:
 - Drag and drop a variable from the Dictionary onto a table entry.
 - Use the PLC Variable Creation Wizard for Kollmorgen devices.
 - Directly map/unmap the Inputs/Outputs to PLC variables using the PLC Variable Selector.

Select the channel(s) you want to map. Selection may be done by click-dragging or shiftclicking a range of entries. The entire table may be selected by clicking on its border. After the selection is made, right-click and select **Map** or press the **Enter** key. This opens

General Properties 10 PDD Selection Mapping Coll Int-Commands Coll Object DD_SafeSlice_1 (AKT2G-SDO-004-000) I/0	d-Dictorery	
AKT2G-SDO-004-000, 4 Ch. Safety Output 24V, 0.5A	1) Select the desired channels 2) Richt-click and select "map"	PLC Variable Creation Wizard
	VO Tables V Digital Output	^
	Channel Soc [bit] PLC Variable B-1 01 1 B-1 02 1	
	Safe Digital Output Channel Size [bit] PLC Variable Second	
	□ 1 □ 02 □ 03	v
	10 Table entries in yellow are safety objects. PLC variables mapped to these entries must be rea Digital Output logic is disabled by default. Use safety parameters to enable. See the safety digital	

the PLC Variable Selector.

	lect the desir ght-click and :	PLC Varial	ole Cr	eation Wizarc	
I/O Tables					
⊿ Di	igital Input				
	Channel	Size [bit]	PLC Varia	ble	
	1	1	(Global)/I	oMas	terAbs
	🔀 2	1	(Global)/I	oMas	terRel
	間 3	1	(Global)/I	-FSto	<u>n</u>
	🔀 4	1			Map
	5	1			Unmap
	🎇 6	1			
	2 🔀	1			
	🔀 8	1			
in.,		-		فليسا	-

🗙 TIP

The **Unmap** command in the contextual menu allows you to remove the link between the variable and the associated channel(s). In addition, deleting a variable from the dictionary which is mapped to the channel(s) also removes the link(s).

Choose the variable to be linked to the channel(s) or PDO object.

Select a va	ariable below.			
Name		Туре	<u>^</u>	
V	Global) 'elOverridePercent)verrideBusy /loveAxis1	INT BOOL BOOL		
R	leady	BOOL		
0	irection	BOOL		
E	StopButton	BOOL		
R	esetButton	BOOL		
C	penButton	BOOL		
0	loseButton	BOOL		
S	tartMove	BOOL		
0)ir	SINT		
S	tartMoveNow	BOOL		
A	xisPoweredON	BOOL		
S	tartMoveG	BOOL	-	
Create PLC Variable				
	Ok	Cancel		

NOTE

The list of variables is filtered to display only those with relevant types.

- PLC Variable Selector displays Read Only variables when mapping the Input channels from the I/O tab.
- Variables assigned to the Safe Digital Input and Safe Digital Output channels must be Read Only.

IMPORTANT

Please be aware of the following limitations of PLC variable mapping.

- 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 (AKD, AKD PDMM, PCMM, or Slice).

For 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 Also:

- For details on the Create PLC Variable and Advanced buttons, see PLC Variable Selector.
- For more details on parameters, see Analog I/O Parameters.
- For more information on the AKD Onboard EtherCAT I/Os, see Configure Onboard I/O.
- For more information on the AKD PDMM or PCMM local digital I/Os, see Configure Controller Onboard I/O.

7.10.2 PLC Variable Creation Wizard

This wizard allows you to automatically create a list of variables used for the mapping. The variable type is **Boolean** for digital I/Os and **UINT** for analog I/Os.

Create PLC Variable
Creation parameters
From channel 4 💌 to 7 💌
Scope (Global) 👻
Base Name AKD_2_DI_##
Ok Cancel

Figure 5-101: Wizard to Create PLC Variable - Parameters

Field	I	Description				
From chan		Defines the range of channels you want to map automatically				
Scope		Defines where the variables are created (if you select the Global scope, then the variables are created under the Global node in the Dictionary)				
		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 will replace ##.				
5 I 6 I	Variable N DInput_4 DInput_5 DInput_6 DInput_7					

Figure 5-102: Wizard to Create PLC Variable - Mapped Channels

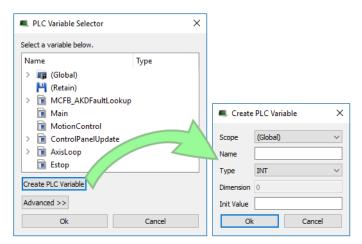
C	Dictionary							₽×
	Controller:PLC	Track	Selectio	n				
	Variables Enums Bit	Fields						
	14 Name	Туре	Dim.	Attrib.	Init val	User	КVВ	Тε
	bMasterAbs	BOOL			FALSE			*
	bMasterRel	BOOL			FALSE			
	bEStop	BOOL			FALSE			
	bLedStatus	BOOL	[03]					
	Profiles	ProfilesCo						
	EtherCAT	EtherCAT		Read Only				
	PipeNetwork	PNCode						
	AKD_2_DI_4	BOOL)					
	AKD_2_DI_5	BOOL						
	AKD_2_DI_6	BOOL						
	AKD_2_DI_7	BOOL						
	Retain variable	s						
	and the second s	and the state of t	ALC LOLO	and the second second	And a state of the	and and a state of	and the second	and the second

Figure 5-103: Wizard to Create PLC Variable - Variables in the Dictionary

7.10.3 PLC Variable Selector

7.10.3.1 Create PLC Variable Button

This button allows the creation of a new variable to be mapped to the selected item(s).



Field	Description
Scope	Defines where the variable is created
Name	See Name a variable
Туре	Defines the Type of the variable, and its Dimension if the variable is an array
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 Initial Value of a Variable

Based on the variable **Type** we select, the KAS IDE will automatically set 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.

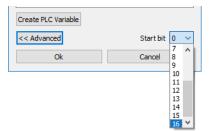
★ TIP

- Safety PDOs are highlighted in yellow with a border.
- Any variable assigned to a FSoE PDO must be Read Only. See About Safety, PDOs, and Variables for more information.
- See How To Set Up a Variable to Monitor Safety Data for step-by-step instructions.

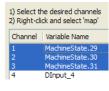
See also Create Variables

7.10.3.2 Advanced Button

For integer variables with types stored on several bits, the Advanced button gives access to the **Start bit** definition. This allows you to link a set of channels to a specific range of bits within an integer variable.



For example, when you 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.



NOTE

When the selected variable is an array, the "Start bit" is disabled. An offset is not allowed when mapping to an array.

7.10.4 Analog I/O Parameters

7.10.4.1 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.

General Properties	I/O	PDO Selection/Mappin	g Distributed Clock	CoE Init-Com	mands	CoE Obje	t-Dictionary			
AL_Slice_1 (AKT2G-AN-430-000) I/O										
4Ch. Ana. Input	+/-10V Diff	., +/-20mA SingleEnded,			1) Sel 2) Big	lect the desir	ed channels		PLC Varia	able Creation Wizard
	G		100-1			Tables				
	E F	Errien			~ Ar	nalog Input				
Run L	ED-	Err LED CH14				Channel	Size [bit]	PLC Variable	Offset (V)	Gain
	;	EE:5				1	16	(Global)/Al_SliceAl_1	0.0	1.0
Ch1	u++	Ch1 U-	00 -Ø-G	11		2	16	(Global)/Al_SliceAl_2	0.0	1.0
Ch2	U+	- Ch2 U-	00 - Ø- 9	2		3	16	(Global)/Al_Slice_Al_3	0.0	1.0
Ch3 Power contact	U+	Ch3 U-		Voltage		4	16	(Global)/Al_Slice_1_A	0.0	1.0
Ch4	U+	Ch4 U-	o o _⊗→ci							
	• v (i	linput Ch1	ୁର୍ବ, -Ø-ସ	11-						
Power contac +2	•v		0,0, -Ø→C	12 Current						
+2	4V		, 2, 0, −,⊗→C							
+2	·vj	Input Ch4	<u></u>	14_						
dan	·~~		~~	mail	dh		A	···· ···		

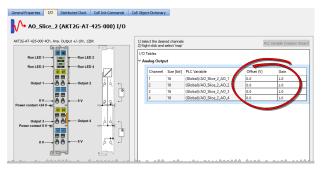
See also E-Bus Analog I/O Data Representation

7.10.4.2 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 also E-Bus Analog I/O Data Representation

7.10.4.3 AKT Input Terminals

318

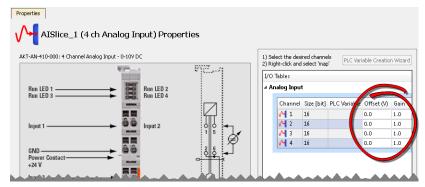
The process data that are transferred to the Bus Coupler are calculated using the following equation:

 $Y_a = (B_a + X_{ADC}) \times A_a$

 $Y_{out} = B_w + ((A_w + A_h) \times Y_a)$ With the following parameters:

Xadc	Output values of the Analog Input Modules A/D converter
Yout	Process data to the controller
Ba , Aa	Manufacturer offset and gain compensation [‡]
Ah	Manufacturer scaling: default gain [‡]
Bw , Aw	User scaling: Offset and Gain as set in the <i>Analog Input Properties</i> (see image below).

[‡] 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 zero for all supported terminals.

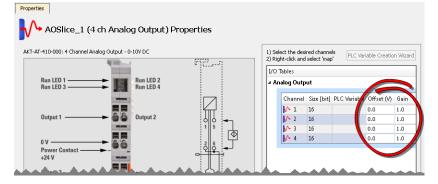


7.10.4.4 AKT Output Terminals

The process data that are transferred to the Bus Coupler from the controller are calculated using the following equations:

$Y_2 = B_w + ((A_w \times A_h) \times X)$						
$Y_{dac} = Y_2 \times A_a + B_a$						
x	Controller Process data					
Ydac	Controller data to analog output module D/A converter					
Ba , Aa	Manufacturer offset and gain compensation [‡]					
Ah	Manufacturer scaling: default gain [‡]					
Bw , Aw	User scaling: offset and gain as set in the <i>Analog Output Properties</i> (see image below).					

[‡] The manufacturer default offset is zero for all supported terminals. The manufacturer default gain is 1 for all supported terminals.



7.11 Configuring EtherCAT

This section provides reference information for the EtherCAT entry in the Project Tree. Specifically this includes the EtherCAT Devices tab, EtherCAT Master Settings Tab, ENI File Tab, ESI Files Tab, and FSoE / PDO Connections Tab tabs. These are accessed by double-clicking on the EtherCAT item in the Project Tree or right-clicking and selecting **Properties**. Reference information for devices in the project can be found in EtherCAT Devices.

Project Explorer	Ξ×	1.0		E	therCAT			X	5
Project View		Devices	Master	ENI File	ESI Files	FSoE / PD	O Connections		
Name	Туре								2
🗸 📷 System	Sinope								5
✓ ■ Controller	[10.50.67.76]		🔓 Ether	CAT De	vices				>
V 📷 PLC									<
> T Programs		Show o	nly drives						5
> 📷 Subprograms			,						
S Defines		# N	ame Vend	or Product	Model	FW Version	Drive Axis Numb	er Mapped to Axis	Simulated \leq
V 📷 Motion									>
> 📷 Profiles]
> 🍖 PLCopen	PLCopen								>
Controller Onboard I/O									
📥 EtherCAT									
References									
Fieldbus	m		\sim	~~~~	~~~~	~~~~			<u>}</u>

7.11.1 EtherCAT Devices tab

The EtherCAT Devices tab lists all the EtherCAT devices and provides for the ability to discover and map their use. The interface is slightly different for AKD and AKD2G drives. The AKD2G prompts you to select the mapping by the drive axis.

Sho	w only drives							🔎 Scan De
	Name	Vendor	Product	Model	FW Version	Drive Axis Number	Mapped to Axis	Simulated
	AKD_1	Kollmorgen	AKD EtherCAT Drive (CoE)	AKD-M00306-MCEC-0000	01-20-00-001	1	(None) V	N/A
	AKD 2	Kollmorgen	AKD EtherCAT Drive (CoE)	AKD-P00306-NAEC-0000	01-20-00-001	1	(None) V	N/A
	Coupler 1	Kollmorgen	AKT-ECT-000 EtherCAT Fieldbus coupler	N/A	(Unknown)	N/A	N/A	N/A
	Coupler 2	Kollmorgen	AKT2G-ECT-000-000 EtherCAT Gateway	N/A	(Unknown)	N/A	N/A	N/A
	Al_Slice_1	Kollmorgen	AKT2G-AN-430-000 4Ch. Ana. Input +/-10V Diff., +/-20mA SingleEnded, 16 Bit		05	N/A	N/A	N/A
	AO Slice 1	Kollmorgen	AKT2G-AT-425-000 4Ch. Ana. Output +/-10V, 12bit	AKT2G-AT-42	04	N/A	N/A	N/A
	AO_Slice_2	Kollmorgen	AKT2G-AT-410-000 4Ch. Ana. Output 0-10V, 12bit	AKT2G-AT-41	04	N/A	N/A	N/A
	Al_Slice_2	Kollmorgen	AKT2G-AN-240-000 2Ch. Ana. Input PT100 (RTD)	AKT2G-AN-24	07	N/A	N/A	N/A
	Al_Slice_3	Kollmorgen	AKT2G-AN-400-000 4Ch. Ana. Input Thermocouple (TC)	AKT2G-AN-40	07	N/A	N/A	N/A
	DI_Slice_1	Kollmorgen	AKT2G-DN-002-000 2Ch. +/- Counter 24V, 100kHz	DN-002	04	N/A	N/A	N/A
	DI_Slice_2	Kollmorgen	AKT2G-DNH-008-000 8Ch. Dig. Input 24V, 10µs	N/A	(Unknown)	N/A	N/A	N/A
	DI_Slice_3	Kollmorgen	AKT2G-DN-008-000 8Ch. Dig. Input 24V, 3ms	N/A	(Unknown)	N/A	N/A	N/A
	DO_Slice_1	Kollmorgen	AKT2G-DT-008-000 8-channel digital output terminal 24 V DC, 0.5 A	N/A	(Unknown)	N/A	N/A	N/A
	DI_SafeSlice_1	Kollmorgen	AKT2G-SDI-004-000, 4 Ch. Safety Input 24V	EL1904	06	N/A	N/A	N/A
	DO_SafeSlice_1 AKD2G_1	Kollmorgen Kollmorgen	AKT2G-SDO-004-000, 4 Ch. Safety Output 24V, 0.5A AKD2G Dual Axis FS1 EtherCAT Drive (CoE)	EL2904 AKD2G-SPE-6V06D-A1F3-0000	04 M_02-05-00-000	N/A	N/A	N/A
						1	PLCAxis2 ~	N/A
						2	PLCAxis2 ~	N/A
	Device 1	BBH Products GmbH	scu	SCU-	03.00.00.61	N/A	N/A	N/A
	AKD2G 2	Kollmorgen	AKD2G Dual Axis FS2 EtherCAT Drive (CoE)	AKD2G-SPE-6V06D-A2F3-0000	M 02-05-01-000			

Figure 5-104: EtherCAT Devices Summary Form

First generation AKD drives support one axis per drive, while AKD2G drives support two axes per drive. When configuring drives in the EtherCAT Devices tab, you will need to designate which motion axis applies to each "drive axis", which is the actual axis the drive controls.

Item	Description
Show only drives	This option hides from the list all EtherCAT slaves that are not drives

320

Item	Description				
Scan Devices		erCAT messages to discover the physical nt in the network (see explanation below)			
Name, Vendor, Product, and Model	and model number for each	ndor name, product name (or code number), device is displayed and ordered by the del (when available) includes the extension and for example.			
FW Version	For Kollmorgen drives, the f that:	irmware version is displayed on the conditions			
	 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 is displayed if the Scan Devices routine was successfully run and the device supports reading this informaiton over EtherCAT (i.e. it uses CoE object 0x100A:0).				
	Otherwise (Unknown) ^{is disp}	played.			
	NOTE				
	If a Kollmorgen drive is showing the firmware version as "Unknown", the drive has valid resident firmware (from which to boot), but does not have valid operational firmware. Please download and install the latest operational firmware and reboot the AKD/AKD2G.				
Drive Axis Number	This number designates which axis, of those available on a drive, is being defined. For most drives, this number will just be a "1", but for AKD2Gs, there are two entries, Drive Axis Number 1 and Drive Axis Number 2,				
Mapped to Axis	For each drive, it is displayed if it is:				
 Unassigned: from the drop-down menu, you can choose an not been assigned (it is applicable either for PLCopen or Pipe motion engines). Already mapped to a physical device: the mapping operation using the Scan Devices command. See details in 					
Simulated	Select this option when you want to simulate the device, which means that the device is not used and no communication to this device is performed through the fieldbus.				
	Mapped to an Axis?	Simulated State			
	No	Simulation is not applicable			
	Yes	 If Drive is mapped to a physical drive, then the simulation is Enabled, so you can set state to Yes/No> Display checkbox If Drive is not mapped to a physical drive, Simulation is forced to Yes 			

Item	Description
Save parameters to NV memory	Allows you to save the drives' parameters to the _{NVRAM} of each drive currently selected in the list. This action is enabled only when the Online Mode is activated
	You will be automatically be prompted with the option to save modified drive parameters if this action has not been performed prior to the following circumstances.
	 Exiting Online Configuration Mode Disconnecting from the controller Closing the project Exiting the IDE
Upgrade Firmware	This command triggers a firmware upgrade for the selected drives (you can use CtrI+A shortcut to select all drives).
	For more details, refer to _{FAQ} section.

Table 5-23:	EtherCAT	Devices
-------------	----------	---------

7.11.1.1 Scan Devices

The scan process allows the following tasks:

- Discover the devices and modules physically present in the fieldbus network (see EtherCAT Network Physical View)
- Map them to items under the EtherCAT node of the Project Explorer (see EtherCAT Network -Logical View. K-Bus on left, E-Bus on right)

Note that the order of the devices in the tree is the same as in the real fieldbus network.

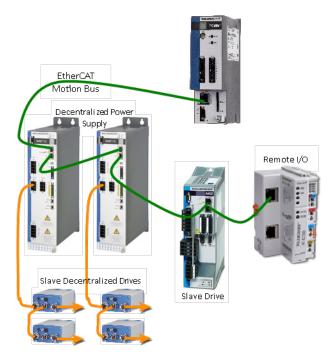


Figure 5-105: EtherCAT Network - Physical View

🛓 📥 EtherCAT		[🗸 🔫 Etł	nerCAT	
AKD_Drive2	AKD Drive			Coupler_1	AKT2G-ECT-000-000
📄 🏭 Coupler_1	KBus Coupler		-	DI_Slice_1	AKT2G-DN-008-000
DISlice_1	8 ch Digital Input			DO_Slice_1	AKT2G-DT-008-000
- 🔯 AKD_1	AKD Drive		-	Al_Slice_1	AKT2G-AN-430-000
E. E Coupler_2	KBus Coupler 8 ch Digital Input		- Î	AO_Slice_1	AKT2G-AT-410-000
DISICE_1	8 ch Digital Input		-	AO Slice 2	AKT2G-AT-425-000
DISlice_3	4 ch Digital Input			ENC_Slice_1	AKT2G-ENC-190-000
DOSlice_1	4 ch Digital Output		-	BRC Slice 1	AKT2G-BRC-000-000
DOSlice_2	8 ch Digital Output		> Ö	AKD2G_1	AKD2G Dual Axis FS3
				-	

Figure 5-106: EtherCAT Network - Logical View. K-Bus on left, E-Bus on right

During the scan operation, all physical devices connected to the EtherCAT network will be discovered and identified. The KAS IDE will list the devices in the order they are connected. Kollmorgen devices, which include AKD and S300/S700 drives, Standard I/O Couplers, and remote I/O terminals (for a list, see "Remote Input/Output Terminals" on page 765) will include detailed information for each device.

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.
0	Excluded	The device has been added manually or has been disassociated from a discovered device.
0	ESI missing	The ESI file is missing.
0	Error	For AKD, shown when there is an error.

Table 5-24: EtherCAT device icon descriptions.

7.11.1.1.1 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. Nor between AKT-DN-008-000 and AKT-DNH-008-000.
- Devices other than those made by Kollmorgen will be identified by the Vendor Name (or ID number) and Product Name (or ID number). If the device is missing an ESI file, then you will need to import the ESI file supplied by the device Vendor. The ESI file is required by the IDE to decode and display the Product Name, Device Description, and other details.
- ESI files for any MDP devices connected ot the network should be added to KAS IDE's ESI library before starting a scan. If the ESI file for an MDP device is not available, then the scan cannot identify the device and cannot discover any physical or logical modules under the device.

7.11.2 EtherCAT Master Settings Tab

This tab includes configurations for the EtherCAT bus master.

Devices	Master	ENI File	ESI Files	FSoE / PDO Conne			
EtherCAT Master Settings							
-							
Cycle Time (µs)							
1000 💌							
Network Usage							
F	rame Size (b	vtes) 182	2				
	Transmit Time		FC				
	ransmit nime	2 (µs) 14.	00				
Band	dwidth Usage	e (%) 1.4	1.46				
				All and the second s			

Figure 5-107: EtherCAT Master Settings

Item	Description		
Cycle Time	Duration of one cycle in microseconds (time = 250, 500, 1000 μs) to define the time base period for scheduling the motion and the PLC programs (for more details on scheduling, see "Tasking Model / Scheduling" on page 459)		
Frame Size	e It is the total size (in bytes) of the EtherCAT frame which is sent cyclically. The size is proportional to the number of EtherCAT slaves (and consequently the PDO data) on the network. The EtherCAT frame size may be between 84 to 1500 bytes.		
Transmit Time	It is the time (in microseconds) that it takes to send a frame		
Bandwidth Usage	width It is an estimation of the percentage of the cycle time used to transmit a frame of deBandwidth value goes up when cycle time decreases (see calculation below)		
Working Counter	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:		
Error Limit	Trigger on 1st Error , 2 Errors in 1000 Frames , 10 Errors in 1000 Frames , 100 Errors in 1000 Frames , or Disabled . If the Disabled option is selected the Master will keep 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 with regards to 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 for additional working counter information.		

Table 5-25: EtherCAT Cycle Settings - Form Description

The three read-only fields display **(unknown)** when the **Use imported file** option is selected (see ENI File Tab). Otherwise, they are recalculated and refreshed each time that:

- A device is added or removed
- A device simulation state changes
- The Use imported file check box is cleared

Bandwidth calculation algorithm

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 (μ sec) = (Frame Size in bytes * 8) bits / 100 * 106 bps

For example:

If Frame Size = 100 bytes then Transmission Time = 100*8 / (100*106) = 8 µsec

If cycle time = 1000 µsec then BW% = 8/1000 = 0.8 %

7.11.3 ENI File Tab

During the compilation, the KAS IDE generates the ENI (EtherCAT Network Information) file based on the EtherCAT devices defined in your project.

Devices	Master	ENI File	ESI Files	FSoE / PDO Connections	1
					1
	Ethe	PrCAT N	letwork	Information (ENI) File	1
	Luix				2
					1
Import /	Export				1
Import E	ENI File	Use impor	ted file		1
Export E					1
Export					1
		. حديد بالله	-		3

Figure 5-108: ENI File tab

Item	Description
Import ENI File	Enables you 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 your project.
	This enables you to include EtherCAT devices in your project that are not natively supported by KAS. For more details, refer to Add Unsupported EtherCAT Device ·
Use imported ENI file	Allows you to specify whether or not to use the imported ENI file. See also the paragraph below.
Export ENI File	Enables you to export the ENI file generated by the $_{\mbox{KAS IDE}}$. You can 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.

Table 5-26: ENI File - Form Description

IMPORTANT

Importing an external ENI file overrides all EtherCAT project device information and configuration settings in the IDE. The following 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.

Using 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 the following 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 EtherCAT Devices before downloading the application to the controller.

7.11.4 ESI Files Tab

This tab lists the available ESI (EtherCAT Slave Information) files and provides the ability to add and remove files.

Devices	Master ENI File ESI Files FSoE / PDO Connections EtherCAT Slave Information (ESI) Files
✓ ВВН	BBH Products GmbH
>	SCU_V3.0.062_080620.xml
× 🔤	Kollmorgen
>	AKD-C-for-KAS-Revision_1.xml
\rightarrow	AKD-C.xml
\rightarrow	AKD-N-for-KAS-Revision_1.xml
\rightarrow	AKD-N-for-KAS.xml
\rightarrow	AKD-for-KAS.xml
\rightarrow	AKD2G-SPE-xxxxD-x1xx-0000.xml
\rightarrow	AKD2G-SPE-xxxxD-x2xx-0000.xml
\rightarrow	AKD2G-SPE-xxxxS-x1xx-0000.xml
>	AKD2G-SPE-xxxx-0000.xml
>	AKD2G-SPE_02-01.xml
>	AKD2G-SPE_02-02.xml
>	AKD2G-SPE_02-03.xml
>	AKD2G-SPE_02-04.xml
>	AKD2G-SPE_02-05-00-000.xml
>	AKD2G.xml
>	AKD_Revision_1.xml
>	Danaher_S300_S400_S600_S700_V8.xml
>	Kollmorgen-AKT2G-ECT-000.xml

Figure 5-109: The ESI Files tab

326

ESI files are sorted by vendor and then the files provided by the vendor. Each file lists the device it supports including

- Device name
- Device description
- Device revision number

ESI files are easily added to the list by clicking the *Add* button and navigating to where the file is saved. The *Delete* button will remove a file from the list but not your hard drive.

🗙 TIP

KAS IDE manages Kollmorgen devices (AKD2G, AKD, AKD-N, etc.) and will automatically install 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. For example, the Kollmorgen ESI files: "AKD-for-KAS.xml" or "AKD-N-for-KAS.xml" should not be manually imported.

⊿	X III	Kollmorg	en		
	Þ	KD-C.xn	nl		
	⊿ A	KD-N-fo	or-KAS.xml		
			Name	Description	Revision
			AKD-N	AKD EtherCAT Drive (CoE)	0x2(2)
	⊿ A	KD-for-l	KAS.xml	·	
			Name	Description	Revision
		10	AKD	AKD EtherCAT Drive (CoE)	0x2(2)

Figure 5-110: 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.

For more information on working with ESI files, see the ESI File Management topic.

① IMPORTANT

WARNING: Replacing an ESI file changes the configuration of any associated device to the new file. You will be prompted to confirm or cancel replacing an ESI file and the associated devices' configurations.

ESI files will be 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 "term_A" (use the version in the project) or "term_B" (use the ESI Internal Library version). Selecting "term_A" will overwrite the ESI Internal Library version.
- Pressing the **Add** button in the ESI Files tab view and:
 - importing an ESI file name that already exists in the KAS Internal Library of ESI Files stored on the PC.
 - importing an ESI file that 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.

7.11.5 FSoE / PDO Connections Tab

The FSoE (FailSafe over EtherCAT) tab displays devices discovered on the EtherCAT network, listing their Input (Tx) and Output (Rx) PDOs. This provides you with a way to connect an Input PDO to an Output PDO.

The KAS Runtime will copy 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 could be useful for any application that need to copy PDO data between EtherCAT devices.

	EtherCAT 🔀	Webserv	ver: 10.50.67.76	×	EtherCAT: AKD2G_1
Devices	Master ENI File ESI Files FSoE / PDO	Connections			
-	Connected PDOs				
Input (Tx)) PDOs		Outp	ut (Rx) PDO	s
> 3	AKD_3 [AKD EtherCAT Drive (CoE)]		>	AKD_3	[AKD EtherCAT Drive (CoE)]
> 0	AKD 4 [AKD EtherCAT Drive (CoE)]		>	AKD 4	[AKD EtherCAT Drive (CoE)]
> 🔡	Coupler_2 [AKT-ECT-000 EtherCAT Fieldbus cou	upler]	>	E Couple	er_2 [AKT-ECT-000 EtherCAT Fieldbus coupler]
✓ <u>■</u>	AKD2G_1 [AKD2G Dual Axis SIL2 EtherCAT Driv	ve (CoE)]	~		G_1 [AKD2G Dual Axis SIL2 EtherCAT Drive (CoE
> (0x1A00 Inputs [160 Bits] AIN1.VALUE [16 Bits] AIN2.VALUE [16 Bits] Position actual value - Axis 1 [32 Bits] Digital inputs - Axis 1 [32 Bits] Statusword - Axis 2 [16 Bits] Position actual value - Axis 2 [32 Bits] 0x1A01 Inputs [96 Bits] 0x1A02 Inputs [160 Bits] 0x1A03 Inputs [96 Bits] 0x1A03 Inputs [96 Bits] 0x1A04 Inputs [96 Bits] 0x1A05 Inputs [96 Bits]		>>	V 0x1603 Phy AO AO Cor Tar Cor	Outputs [32 Bits] Outputs [160 Bits] ysical outputs [32 Bits] UT1 Fieldbus Value [16 Bits] UT2 Fieldbus Value [16 Bits] ntrolword - Axis 1 [16 Bits] get position - Axis 1 [32 Bits] ntrolword - Axis 2 [16 Bits] get position - Axis 2 [32 Bits]
	Input			Out	put BitSize
AKD2G	1.Inputs	>>	AKD2G_1.Output		160 🔞

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

An example of mapping PDOs may be found in the topic Set Up FSoE Master and an AKD2G with SafeMotion Monitor.

How to Unmap FSoE PDOs

- Double-click (or right-click) on EtherCAT in the Project Explorer, and select the FSoE tab.
 Select the desired pair of Connected PDOs from the list at the bottom of the tab.
- 3. Click the Delete Connection button.

NOTE

Inputs and Outputs must have identical Bit sizes in order to be paired.

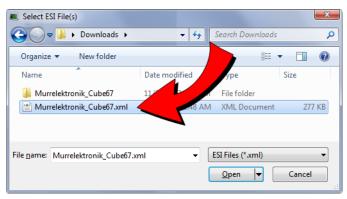
See Also: Safety over EtherCAT

7.11.6 ESI Files with References (MDP)

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 MDP (Modular Device Profile) 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 IDE's ESI library, *only* add the top-level file that contains the references to the other files and sub-directories. The IDE will check the selected ESI file for reference files and automatically add the reference files.

For example, when adding the Murrelektronik Cube67 MDP (modular) device ESI file, select the toplevel ESI file.



Do not select the lower-level module ESI files found in a sub-folder.



IMPORTANT

330

WARNING: Always add the top-level ESI file to the IDE's ESI library. Do not add, delete, or upgrade lower-level module files to the 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 IDE's ESI library, an error message will notify you which sub-files are missing.

7.11.6.1 EtherCAT Device Configuration - MDP Devices

The KAS IDE incorporates the ability to configure and manage EtherCAT MDP (modular device profile) devices. Devices which meet the ETG.5001 standard may be added to the EtherCAT network in the

Project Explorer and modules may be mapped to them. This includes devices such as I/O device modules, safety device modules, or fieldbus gateway modules.

7.11.7 ESI File Management

ESI files which are referenced from a project's EtherCAT devices are stored in the project.

• Upon opening a project the ESI files are copied to the local folder.

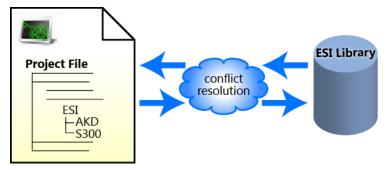


Figure 5-111: Opening — Upon opening a KAS project, the project's ESI files are compared to the internal library. If there are conflicts, you are prompted to resolve them.

• Adding or deleting ESI files from KAS IDE affects the internal library

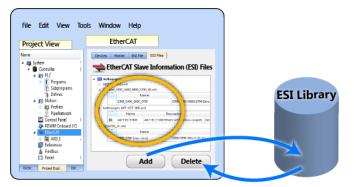


Figure 5-112: Adding/Deleting — Adding or deleting an ESI file from the KAS IDE affects KAS's internal library of ESI files.

• Upon saving a project the ESI files are copied to (project folder)\Controller\ESI.

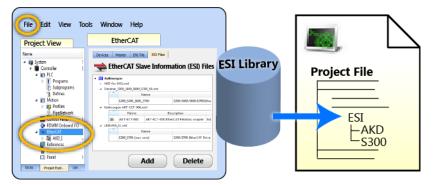


Figure 5-113: Saving — When a KAS project is saved, a copy of the ESI file(s) is included in the project file.

★ TIP

The ESI files are stored along with the project so the project can easily be moved to another location and it will 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.

IMPORTANT

WARNING: Replacing an ESI file changes the configuration of any associated device to the new file. You will be prompted to confirm or cancel replacing an ESI file and the associated devices' configurations.

ESI files will be 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 "**term_A**" (use the version in the project) or "**term_B**" (use the ESI Internal Library version). Selecting "**term_A**" will overwrite the ESI Internal Library version.
- Pressing the Add button in the ESI Files tab view and:
 - importing an ESI file name that already exists in the KAS Internal Library of ESI Files stored on the PC.
 - importing an ESI file that 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.

7.12 EtherCAT Devices

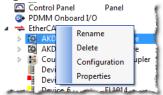
This section provides reference for defining EtherCAT devices. Devices are added by right-clicking the **EtherCAT** entry in the Project Explorer. Devices are seen as branches of the EtherCAT item. Double-clicking a device opens a tab labeled as "**EtherCAT**: **[device name]**". See Configuring EtherCAT for information on the tree entry.

Project Explorer	5 × 5	- EtherC	AT: AKD_1			EtherCAT: AKD_2	
Project View		General Properties I/O Con	figuration PDO Selection/Mapping	PDO Editor	Distributed Clock	CoE Init-Commands	CoE Object-Dictionary
Name	Туре						
🗸 📷 System	Sinope						
V 💼 Controller	[10.50.68.44]	AKD_1 (AKD) General Properties				
> ig PLC							
> 👩 Motion		Information					
📀 Controller Onboard I	/0	Name:	AKD_1				
🗸 🔫 EtherCAT		Description:	AKD EtherCAT Drive (CoE)				
AKD_1	AKD Drive	Vendor:	Kolmorgen				
AKD_2	AKD Drive	Product Code:	0x414844 (4279108)				
> 📓 Coupler_1	KBus Coupler	Revision Number (from ESI file):	0x1A0000 (1703936)				
References		Revision Number (from device):	0x1A0000 (1703936)				
옯 Fieldbus		URL:	Not Available				
Control Panel	Panel	ESI File:	C:\Users\william.gaffga\AppData\Loc	al Kollmorgen (K	AS\Astrolabe\ESI\AKD	-for-KAS.xml	Import ESI File
FaultReportPanel	Panel	EtherCAT Address:	1001				
KVBProject	KVB 2						
		Topology					
		Port A, MII:	EtherCAT Master				
		Port B, MII:	AKD_2 (1002)				
Libraries Project Explorer Dict	ionary						
Information and Loos							A

7.12.1 EtherCAT Device General Properties Tab

This tab provides information about the selected EtherCAT device and the associated ESI file. It also provides the ability to import an ESI file if one is not present. There are two ways to access this tab:

- double-click on an EtherCAT device
- right-click on an EtherCAT device and select Properties.



There are two main sections in the dialog: Information, and Topology.

7.12.1.1 Information

This section details the device's basic information and is not editable. In addition to the device and vendor information is 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. Lastly, this section lists the **EtherCAT Address** for devices and **FSoE Address** for safety devices which have been discovered.

General Properties I/O PDC	Selection/Mapping CoE Init-Commands CoE Object-Dictionary	
DO_SafeSlice_	2 (AKT2G-SDO-004-000) General Properties	
Information		
Name:	DO_SafeSlice_2	
Description:	AKT2G-SDO-004-000, 4 Ch. Safety Output 24V, 0.5A	
Vendor:	Kollmorgen	
Product Code:	0xB583052 (190328914)	
Revision Number (from ESI file):	0x120000 (1179648)	
Revision Number (from device):	0x120000 (1179648)	
URL:	https://www.kollmorgen.com	
ESI File:	C: \Users\Dusty.Schafer\AppData\Local\Kollmorgen\KAS\Astrolabe\ESI\Kollmorgen-AKT2G-ECT-000.xml	Import ESI File
EtherCAT Address:	1006	
FSoE Address(es):	0x2 (2)	
Тороюду		
Port A, E-Bus:	DO_Slice_1 (1005)	
Port B, E-Bus:	DI SafeSlice 2 (1007)	

Why are the Revision Numbers different?

It is possible that 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. If this occurs, your device may not have access to the full capabilities of the latest firmware.

Revision Number (from ESI file):	0x3010000 (50397184)	Ν
Revision Number (from device):	0x3010001 (50397185)	ν

7.12.1.2 Topology

This section lists the device's ports and assignments. There are three possible states for this information.

Condition	Appearance
Physical Devices	Topology Port A, E-Bus: Device_5 (1001) Port B, E-Bus: Device_7 (1003)
Virtual Device	Topology Port A, E-Bus: Not Applicable Port C, E-Bus: Not Applicable Port D, MII: Not Applicable
ESI file is missing	Topology Missing Port Information

7.12.2 Configuration Tab

Double-clicking on the AKD drive or right-clicking on the drive and selecting Configuration opens this tab. The contents of this tab allow you to set the parameters for the AKD drive.

See Configure the AKD2G/AKD Drive for more information.

7.12.3 PDO Selection/Mapping Tab

This tab includes the PDO configurations for an EtherCAT device. Assigned PDOs and their objects are viewable for the Inputs (Tx) and Outputs (Rx). The PDOs become active when the EtherCAT network is initialized to operation mode.

lect Output (Rx) PDG				Input (Tx) PDI	6			
	ls .			Select Input	Tx) PDOs			
0x1602 Outputs				^ ~ 0x1A00 li	puts			
Index Subin	lex Object Name	Size [bit]	PLC Variable	Inde	Subindex	Object Name	Size [bit]	PLC Variable
0x60B8 0	Touch probe function - Axis 1	16	A	0x3780	1	AIN1.VALUE	16	Managed in I/O tab
0x6888 0	Touch probe function - Axis 2	16	<u>A</u>	0x378	1	AIN2.VALUE	16	Managed in I/O tab
0x1603 Outputs				0x604	0	Statusword - Axis 1	16	
0x1603 Outputs 0x1650 Module_1	Controlunad			0x6064	0	Position actual value - Axis 1	32	
DX TOSO MODULE_T	controlword			0x60FI	0	Digital inputs - Axis 1	32	Managed in I/O tab
Index Subin	dex Object Name	Size [b	it] PLC Variable	0x684	0	Statusword - Axis 2	16	
0x2051 0	Axis 1 Release the brake	1		0x686-	0	Position actual value - Axis 2	32	
0x2052 0	Axis 1 Reset the encoder position	on 1		0x1A01 li				
0x6630 0	Axis 1 Restart acknowledge	1		0x1A01 l				
0x6632 0	Axis 1 Error acknowledge	1		0x1A50 M				
0x6640 0	Axis 1 STO command	1		- OKIAJON	odule_1.5tz			
0x6650 1	Axis 1 SS1 Instance 1	1		Inde	Subindex	Object Name	Size [b	it] PLC Variable
0x6650 2	Axis 1 SS1 Instance 2	1		0x206	0	Axis 1 SBA	1	
0x6650 3	Axis 1 SS1 Instance 3	1		0x206	0	Axis 1 SBC	1	
0x6650 3	Axis 1 SLA	1		0x206	0	Axis 1 Temperature warning	1	
0x6653 0	Axis 1 SOS command	1		0x206	0	Axis 1 SLA	1	
	Axis 1 SUS command			0x206	0	Axis 1 SLA	1	
0x6653 0	Axis 1 SS2 Instance 1	1						
0x6653 0 0x6668 0		1		0x661	0	Axis 1 safe position actual value	10 10	

Figure 5-114: The PDO Selection/Mapping tab including 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 About Safety, PDOs, and Variables for more information.
- See How To Set Up a Variable to Monitor Safety Data for step-by-step instructions.

Each assigned PDO is listed by its Index (hex) and Name. The objects associated with each PDO are listed below the name. The objects are identified by their object dictionary **Index** and **Subindex**. The **Object Name** provides a simple description. The **Size** determines the data length. The **PLC Variable** mapped to the PDO objects is also listed.

elect Inp	ut (Tx) PC	OS				
⊿ 0x1A	00 Chan	nel 1				
	Index	Subindex	Object Name	Size [bit]	PLC Variable	
0x6000 1 Input 1						
✓ 0x1A01 Channel 2						
	Index	Subindex	Object Name	Size [bit]	PLC Variable	
	0x6010	1	Input	1		
⊿ 0x1A	02 Chan	nel 3				
	Index	Subindex	Object Name	Size [bit]	PLC Variable	
	0x6020	1	Input	1		
✓ 0x1A03 Channel 4						
	Index	Subindex	Object Name	Size [bit]	PLC Variable	
	0x6030	1	Input	1		

7.12.3.1 Map PLC Variable to PDO Object

PLC variables can be mapped to PDO objects by:

- Double-clicking in a PLC variable cell to open the PLC Variable Selector.
- Right-clicking in a PLC variable cell and select Map or Unmap.

PDO Selecti	PDO Selection/Mapping								
-	PDO Selection for Device_5 (EL2004)								
Output (R	x) PDOs								
Select Ou	itput (Rx) I	PDOs							
⊿ 0x10	500 Chan	nel 1							
	Index	Subindex	Object Name	Size [bit]	PLC Variable				
	0x7000	1	Output	1					
✓ 0x1601 Channel 2									
	Index	Subindex	Object Name	Size [bit]	PLC Variable				

- Drag-and-drop a variable from the Dictionary to a PLC variable cell.
- Multiple objects may be selected by:
 - Shift-clicking a contiguous range of objects.
 - Ctrl-clicking 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 will display only Read Only variables.
- Variables created by clicking the Create PLC Variable button, when accessed from a safety PDO, will automatically be set to Read Only.
- There will be no warning if an assigned variable is changed to Read/Write, but an error will be reported when the project is compiled.

For more details, please see Map Input and Output to Variables.

7.12.3.2 Select Input and Output PDOs

Press the **Select Input (Tx) PDOs** or **Select Output** (**Rx) PDOs** button to choose the Input or Output PDOs. The appropriate dialog box will open.

PLC Variable Selector	×
Select a variable below.	
Name	Туре
i (Global)	
💾 (Retain)	
MCFB_AKDFaultLookup	
🗙 🗊 Main	
MyROVar	BOOL
MotionControl	
ControlPanelUpdate	
💼 AxisLoop	
💼 Estop	
Create PLC Variable	
<< Advanced	
Ok	Cancel

ut (Rx)		-				_	t (Tx) PDOs	-			
ct Outp	out (Rx) I	PDOs				Sele	sct Input (Tx) Pi	208			
0x16F	F Mapp	ing RxPDC	Device Control			1	0x1AFF Mapp	ing TxPDO	Device Status		
	Index	Subindex	Object Name	Size [bit]	PLC Variable		Index	Subindex	Object Name	Size [bit]	0 Vari
c	h/F200	1	Device Control W	16			0xF100	1	Summarized module diag	1	
							0xF100	2	Diagnosis message availa	1	
							0xF100	3	Module bus error	1	
							0xF100	4	I/O configuration changed	1	
							0xF100	5	Gap	3	
							0xF100	6	Undervoltage UI	1	
							0xF100	7	Overvoltage Usys	1	
							0xF100	8	Undervoltage Usys	1	
							0xF100	9	Gap	1	
							0xF100	10	Master configuration error	1	
							0xF100	11	Gap	2	
							0xF100	12	Force mode active	1	
							0xF100	13	Module bus failure	1	

- The upper portion contains checkboxes to select specific PDOs for the EtherCAT network cyclic data.
- The lower portion contains the list of object(s) included within a specific PDO.

7.12.3.3 Viewing the contents of a PDO

Click on a row in the upper portion of the PDO Selector dialog box to view the contents of that PDO. The lower portion will update to list the associated objects. Using the following 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.

PDO Selection:					
	PDO Index	Name			
2 🔲 0x1A00	-	Channel 1			
🗆 0x1A01		Channel 2		1.	Selected PDOs
🗸 🗸 0x1A02		Channel 3			Selected row
▼ 0x1A03		Channel 4			Contents of selected row's
0x1A00 Conte	ints: (3)			5.	PDO.
Index	Subindex	Object Name	Size [bit]		
0x6000	1	Input	1		

NOTE

- 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. Using the following image as an example, choosing PDO 0x1A00 excludes selecting PDO 0x1A04.

PI	00 Index	Name				
🔽 0x1A00		COM TxPDO-Map Inputs Channel 1				
0x1A01		COM TxPDO-Map Inputs Channel 2				
0x1A04 (exc	luded by 0x1A00)	COM TxPDO-Map Inputs Channel 1				
0x1A05		COM TxPDO-Map Inputs Channel 2				
x1A00 Content	s:					
Index	Subindex	Object Name	Size [bit]	1		
0x6000	1	Status_Transmit accepted	1	-		
0x6000	2	Status_Receive request	1	1		
0x6000	3	Status Init accepted	1	_		

• Device vendors determine the PDO(s), content and possible selection exclusivity. This information is defined inside the device vendor's ESI file. Please contact the device vendor for details about a specific device.

7.12.3.4 How To Set Up a Variable to Monitor Safety Data

This example walks you through setting up a variable which can be 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)

While this step is optional, it is quite useful and will save you time and effort.

	E	EtherCAT
evices Master ENI File ESI Files FSoE / PDO Connections		
Connected PDOs		Output (Rx) PDOs
✓	oE)]	Eff Device_1 [SCU] Get AKD2G_1 [AKD2G Dual Axis SIL3 Basic Functions EtherCAT Drive (C) or 1602 Outputs [32 Bits] or 1603 Outputs [160 Bits] or 1603 Outputs [160 Bits] or 1604 Outputs [160 Bits] or 1604 Module_2.Controlword [85 Bits]
Connected PDOs		
Input		Output BitSize
Device_1.0x1A00 Module_1.Controlword	>>	> AKD2G_1.0x1650 Module_1.Controlword 85 0
Device_1.0x1A20 Module_2.Controlword	>>	> AKD2G_1.0x1660 Module_2.Controlword 85 (
AKD2G_1.0x1A50 Module_1.Statusword	>>	> Device_1.0x1600 Module_1.Statusword 204 6
AKD2G_1.0x1A60 Module_2.Statusword	>>	> Device_1.0x1620 Module_2.Statusword 204 (

2. Right-click (or double-click) on 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.

Input ((Tx) PDOs	5				
Select	t Input (T	x) PDOs				
<mark>∼ 0</mark> ×	1A00 Mo	odule_1.Cor	trolword			^
	Index	Subindex	Object Name	Size [bit]	PLC Variable	
	0x6000	1	FSOE Command	8		
	0x6000	3	FSOE Master CRC_0	16		
	0x6000	4	FSOE Master CRC_1	16		
	0x6000	2	FSOE ConnID	16		
	0x6001	1	Axis 1 STO	Map		
	0x6001	1	Axis 1 SS1 Instance 1	1		
	0x6001	1	Axis 1 SS2 Instance 1	1		
	0x6001	1	Axis 1 SOS	1		
\mathcal{N}	\sim	\sim	MARA MAN			NV U

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**.

PLC Variable Selecto	ır X
Select a variable below.	
Name	Туре
iii (Global)	
💾 (Retain)	
📄 Main	
MachineLogic	
Create PLC Variable	
<< Advanced	
Ok	Cancel

4. Define a variable by giving it a name, type, and optionally an initial value, and click **OK**.

💻 Create	PLC Variable	Х
Scope	Main	\sim
Name	SafeWatch_1	
Туре	BOOL	\sim
Dimension	0	
Init Value		
Ok	Cancel	

5. The **PLC Variable Selector** now displays variable. Click OK.

PLC Variable Selecto	r X
Select a variable below.	
Name (Global) (Retain) (Retain) (Retain)	Туре
SafeWatch_1 MachineLogic	BOOL
Create PLC Variable	
<< Advanced	
Ok	Cancel

The variable is now assigned and available to monitor any object in the same Index:Subindex.

KAS IDE User Guide | 7 KAS IDE Reference

	runc_neor	ntrolword		
Index	Subindex	Object Name	Size [bit]	PLC Variable
0x6000	1	FSOE Command	8	
0x6000	3	FSOE Master CRC_0	16	
0x6000	4	FSOE Master CRC_1	16	
0x6000	2	FSOE ConnID	16	
0x6001	1	Axis 1 STO	1	Main/SafeWatch_1
0x6001	1	Axis 1 SS1 Instance 1	1	Main/SafeWatch_1
0x6001	1	Axis 1 SS2 Instance 1	1	Main/SafeWatch_1
0x6001	1	Axis 1 SOS	1	Main/SafeWatch_1
0x6001	1	Axis 1 SSR Instance 1	1	Main/SafeWatch_1
0x6001	1	Axis 1 SDI_p	1	Main/SafeWatch_1
0x6001	1	Axis 1 SDI_n	1	Main/SafeWatch_1
0x6001	0	Axis 1 Acknowledge error	1	

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.

🛋 Edit Variable			×
Name:	SafeWatch_1		
Туре:	BOOL		•
Description:			
Init value:			
User Group:			•
Tag:			
Advanced:	🔽 Read Only	INOUT Dim.:	•
	🗌 External 🗌	Syb.	
	OK	Cancel	Help

7.12.4 PDO Editor Tab

This tab provides the ability to modify editable PDOs. There are two sections, Editable PDOs and PDO Content. See PDO Restrictions and Compiler Errors for information about compiler errors and certain constraints.

table P	DOs				PDO Con	tent			
					0x1600	Contents:		Size :	8 bytes
Input	(Tx)				In	dex	Subindex	Object Name	Size [b
	Index	Name	Size [bytes]	User Modified	0×6040	0		Control word	16
0>	×1A00	Inputs	2	No	0×20A4	0		Latch control	16
0>	×1401	Inputs	0	No	0×60C1	1		Position dema	32
0>	×1A02	Inputs	0	No					
0>	×1A03	Inputs	0	No					
Outp	ut (Rx)								
	Index	Name	Size [bytes]	User Modified					
0>	×1600	Outputs	8	No					
0>	×1601	Outputs	8	No					
0>	×1602	Outputs	0	No					
0.	×1603	Outputs	0	No					

Figure 5-115: The PDO Editor tab

7.12.4.1 Editable PDOs

This section lists all of the PDOs which may be edited. PDOs may be created (Add PDO), modified (Edit PDO), or removed from the project (Delete PDO). Please see the EtherCAT device-specific documentation for more details.

-	AKD_1	I/O (AKD) P	Configuration DO Editor	PDO Selection/	Mapping	PDO Edito
⊿ In	iput (Tx)					
	Inde	x	Name	Size [bytes]	User M	odified
	0x1A00		Inputs	2	No	
	0x1A01		Inputs	0	No	
	0x1A02		Inputs	0	No	
	0x1A03		Inputs	0	No	
⊿ 0	utput (Rx)					
	Inde	x	Name	Size [bytes]	User M	odified
	0x1600		Outputs	8	No	
	0x1601		Outputs	8	No	
	0x1602		Outputs	0	No	
	0x1603		Outputs	0	No	
	Add Input (Tx) PDC	Add Output (R	x) PDO Edit F	200	Delete PDO

The Input (Tx) and Output (Rx) PDOs are grouped separately. Each PDO table contains the following information:

Index	Identification value for the PDO. The range is defined by the EtherCAT specification. RxPDOs: 0x1600 to 0x17FF TxPDOs: 0x1A00 to 0x1BFF
Name	Text label to identify the PDO. The name is provided either by the device vendor or user, and 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 as well as on the sum of the sizes of PDOs. See PDO Restrictions and Compiler Errors for more information.

7.12.4.1.1 Add a Flexible PDO

- 1. Click the Add Input (Tx) PDO or Add Output (Rx) PDO button to open the Add Input or Add Output window.
- 2. Select an **Index** value and enter a **Name** for the PDO. The dialog performs a live check for valid entries.
- 3. Click OK.

+	Add Input (Tx) PDO	? 🔀	<table-cell-rows> Add Input (Tx) PDO</table-cell-rows>				
			PDO details		🗢 Add Input (Tx) PDO		? 🗙
	PDO details		Name:		PDO details		
	Name:	MyPDO	Index (Hexadecimal): 0x	1A00	Name:		
	Index (Hexadecimal): 0x	1BAA	Range [0x1A00 : 0x1BFF]	Index is already in use	Index (Hexadecimal): 0x		
	Range [0x1A00 : 0x1BFF]	Index is valid		ОК	Range [0x1A00 : 0x1BFF]] Index out of range	
						ОК	Cancel
		OK Cancel					

Figure 5-116: Examples of live checking for valid entries.

Objects may be added once the PDO has been created. See Add a CoE Object.

EtherCAT device support for flexible PDOs is optional. Please consult the manufacturer's devicespecific manual for details about support for flexible PDOs and which PDO index values are supported.

7.12.4.1.2 Edit a Flexible PDO

A flexible PDO's Name and Index may be edited after it has been added.

- 1. Select a flexible PDO from the list of Editable PDOs.
- 2. Click the Edit PDO button.
- 3. Make your changes. The dialog performs a live check, enabling the OK button when an entry is valid.
- 4. Click OK.

本 Edit Input (Tx) PDO			? 💌
PDO details			
Name:	MyTxPDO		
Index (Hexadecimal): 0x	1A04		
Range [0x1A00 : 0x1BFF]	Index is valid		
		ок	Cancel

7.12.4.1.3 Delete a Flexible PDO

A flexible PDO, which has previously been added, may be deleted.

- 1. Select a flexible PDO from the list of Editable PDOs.
- 2. Click the **Delete PDO** button.
- 3. Confirm your deletion by clicking **OK**.

		S300_S400_S600_	S700) PDO Editor	
	e PDOs put (Tx)			
	Index	Name	Size [bytes]	User Modified
	0x1A00	MyPDO	4	Yes
	0x1A01	MyOtherPDO	10 (1)	Yes
⊿ Ou	tput (Rx)			
	Index	Name	Size [bytes]	User Modified
	0x1600	APDO	4	Yes
		AnotherPDO	10	Yes

7.12.4.2 PDO Content

This section lists the objects within the currently selected PDO. Objects may 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 will appear in the PDO Content section (right side of dialog box).

-									
litat	ole PDOs				PDO	Content			
					0x1	600 Conten	ts:	Size :	8 bytes
a Ir	iput (Tx)] [Index	Subir	ndex Object Name	Size [bit]
	Index	Name	Size [bytes]	User Modified	0x6	040	0	Control word	16
	0x1A00	Inputs	2	No	0x2	DA4	0	Latch control	16
	0x1A01	Inputs	0	No	0x6	0C1	1	Position dema	. 32
	0x1A02	Inputs	0	No					
	0x1A03	Inputs	0	No					
4 0	utput (Rx)								
	Index	Name	Size [bytes]	User Modified					
	0x1600	Outputs	8	No					
	0x1601	Outputs	8	No					
	0x1602	Outputs	0	No					
	0x1603	Outputs	0	No					

Index	Identification value for the PDO. The range is defined by the EtherCAT specification. RxPDOs: 0x1600 to 0x17FF TxPDOs: 0x1A00 to 0x1BFF
Name	Text label to identify the PDO. The name is provided either by the device vendor or user, and 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 will be highlighted in red if the Editable PDO content size is greater than 8 bytes.

7.12.4.2.1 Add a CoE Object

- 1. Select a PDO in the **Editable PDOs** section.
- 2. Click the Add Object button to show the Add CoE Object to PDO window.
- 3. Select an object from the **CoE Object Dictionary** or manually enter the Name, Index, Subindex and Size data, and click **Ok**.

	Object							
Name	e: Manufact	urer status byte:	1	Index (Hexadecimal): 0x	2002			
Size	(Bits): 8			Subindex (Decimal):	1			
CoE	Object Diction	ary						
	Index	Value		Name	Type Flags			
	0×1002	-	Manufacturer st	atus register	UDINT	TX (RO RO RO)	·	
\triangleright	0×1026	2	OS prompt		USINT	(RO RO RO)		
⊿	0x2002	4	Manufacturer st	atus bytes	USINT	(RO RO RO)		
	Subindex	Value	Name		Туре	Flags		
	1	-	Manufacturer st	atus byte 1	USINT	TX (RO RO RO)		
	2	-	Manufacturer st	atus byte 2	USINT	TX (RO RO RO)		
	3	-	Manufacturer st	atus byte 3	USINT	TX (RO RO RO)		
	4	-	Manufacturer st	atus byte 4	USINT	TX (RO RO RO)	0)	
	0x2050	0	Position actual v	value 2	DINT	TX (RO RO RO)		
	0x2077	0	Current actual v	alue	DINT	TX (RO RO RO)		
	0x20A0	0	Latch1P32		DINT	TX (RO RO RO)		
	0x20A1	0	Latch1N32		DINT TX (RO RO			
	0x20A2	0	Latch position 2		DINT	TX (RO RO RO)		
	0x20A3	0	Latch position 3		DINT	TX (RO RO RO)		
	0x20A5	0	Latch status wor	ď	UINT	TX (RO RO RO)		
	0x20A6	0	Latch position		DINT	TX (RO RO RO)		
	0x20A7	0	Latch position 1		DINT	TX (RO RO RO)		
	0×3417	-	REGEN.POWERF	ILTERED	UDINT	TX (RO RO RO)	-	

NOTE

The Flags indicate whether an object can be selected for an Input (Tx) or an Output (Rx) PDO. If the Flags are "-- --" then the object is not eligible for the PDOs. Selecting the wrong object type will generate an error.

7.12.4.2.2 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.
- 4. In the **Rename** pop up, enter the new name.
- 5. Click **Ok**.

344



7.12.4.2.3 Delete a CoE Object

- 1. Select a PDO from the Editable PDOs list.
- 2. Select the object to be deleted from the PDO Content.
- 3. Click the **Delete Object** button.
- 4. Confirm the deletion.

table	PDOs				PD) Content			
					0x1	600 Content	ts:	Size :	8 bytes
Inpu	ıt (Tx)					Index	Subindex	Object Name	Size [bit
	Index	Name	Size [bytes]	User Modified	0×1	5040	0 (2)	Control word	16
-	0×1A00	Inputs	2	No	0×3	20A4	0	Latch control	16
	0×1A01	Inputs	0	No	0×1	50C1	1	Position dema	32
- 1	0×1A02	Inputs	0	No					
1	0x1A03	Inputs	0	No					
⊔ Outp	put (Rx)								
Γ	Index	Name	Size [bytes]	User Modified					
	0×1600 🚺	Outputs	8	No					
	0×1601	Outputs	8	No					
- 1	0×1602	Outputs	0	No					
	0×1603	Outputs	0	No					

7.12.4.3 PDO Restrictions and Compiler Errors

7.12.4.3.1 Restrictions for AKD Devices

The KAS IDE compiler enforces the following 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 will be 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.
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.

Compiler errors will be generated if these restrictions are not met. 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 \ensuremath{\mathtt{Rx}} PDOs that are assigned is not even: 29
```

7.12.4.3.2 Restrictions for AKD2G Devices

The KAS IDE compiler enforces the following 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].
- There is a limitation to the number of PDO objects, based on the the Sample Rate / Cycle Time.

Sample Rate	Cycle Time	Max. PDOs
4kHz	250µs	14 Rx, 14 Tx
2kHz	>500µs	30 Rx, 30 Tx

7.12.4.3.3 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
```

7.12.4.4 Redundant PDO Entries

It is possible within 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 in regards to identification.

In the example below, the entry defined by index 0×6040 , subindex 0, appears in the Rx PDOs 0×1600 and 0×1604 . Similarly, the entry 0×3470 , subindex 4 appears in two different PDOs.

neral Prop	erties	I/O	Configuration	PDO Se	lection/Ma	ipping PD	O Editor	Distri	buted Clo	tk CoE I	nit-Commands	CoE
	A1/		KD) PDO S	'alasti	on /\ 4	nning						
	AL	.D_1 (A	KD) PDO S	electi	OUTAM	apping						
ltput (R×) PDO	s						-Input (Tx) PDOs			
elect Out	:put (F	x) PDOs						Select	Input (T>) PDOs		
0x1600) Out	puts							Index	Subindex	Object Na	me
In	dex	Subindex	Object Na	me	Size [bit]	PLC Varial	ole		0×3470	4	AOUT_INPUT	VAL
0×2	0A4	0	Latch control v	word 1	.6	4		⊿ 0x1	.B23 Inp	uts		
0×6	040	0	Control word	1	.6	4						
0×6	0C1	1	Position dema	nd v 3	2	<u>A</u>			Index	Subindex	Object N	ame
			1						0×2050	0	Position actua	al valu.
0x1601		•							0x20A5	0	Latch status v	vord
0x1604	1 my_	Rx_PDO							0x20A6	0	Latch position	n
In	dex	Subindex	Object Name	Size [bit] PL	.C Variable			0x3470	4	AIN.VALUE	
0×6	040	0	Control word	16	4				0×6041	0	Status word	
									0~6063	0	Position actua	سادت اد

• A redundant Rx PDO entry will generate a compiler error. For example, the redundant Rx PDO illustrated above would generate the following 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 will fail with the error:

Failed to export ENI file, check the logs for more details.

• A redundant Tx PDO entry will generate a compiler warning, such as the example below:

```
EtherCAT: Warning : In AKD_1, PDO object index 0x3470, subIndex 4 is redundant. It is in the following PDO(s): 0x1A04, 0x1B23
```

• A redundant Tx PDO entry will not prevent a successful compilation nor will it generate an error when exporting the ENI file.

7.12.4.5 About Safety, PDOs, and Variables

This topic discusses 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. Mapping a variable to an object is easily done by any of the following methods.

- Double-clicking on the object and then selecting or creating a variable
- Right-clicking on the object and selecting Map and then selecting or creating a variable
- Dragging an existing variable from the Dictionary to the object

Variables assigned to a safety object must be Read Only. Variables that are Read/Write will not show up in dialogs, nor will drag-and-drop work.

NOTE

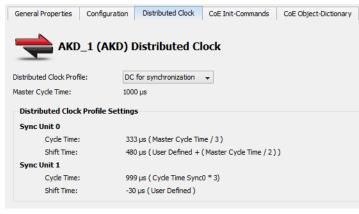
Once a variable is assigned, there is no further validation until the project is compiled. That means a variable could be changed to Read/Write and still be assigned. When you compile the project an error will be generated.

it (Rx) PDO						Input (Tx) PDC	s				
t Output (R	x) PDOs					Select Input (Tx) PDOs				
1602 Out	puts				^	~ 0x1A00 lr	iputs				
Index	Subindex	Object Name 5	Size [bit]	PLC Variable		Index	Subindex	Object Name	Size [bit]	PLC Variable	
0x60B8 0	1	Touch probe function - Axis 1 1	6	A		0x3780	1	AIN1.VALUE	16	Managed in I/O tab	
0x6888		Touch probe function - Axis 2		<u> </u>		0x3781	1	AIN2.VALUE	16	The Managed in I/O tab	
					_	0x6041	0	Statusword - Axis 1	16	-	
1603 Out					_	0x6064	0	Position actual value - Axis 1	2		
1650 Mod	lule_1.Con	trolword				0x60FE	0	Digital inputs - Axis 1	2	Managed in I/O tab	
Index	Subindex	Object Name	Size [b	t] PLC Variable		0x6841	0	Statusword - Axis 2	16		
0x2051	0	Axis 1 Release the brake	1			0x6864	0	Position actual value - Axis 2	2		
0x2052	D	Axis 1 Reset the encoder positio	n 1		i						
0x6630	D	Axis 1 Restart acknowledge	1			> 0x1A01 Inputs > 0x1A02 Inputs					
0x6632	0	Axis 1 Error acknowledge	1			> 0x1A02 la					
0x6640	D	Axis 1 STO command	1			* UXTASU N	lodule_1.5tz	itusword			
0x6650	1	Axis 1 SS1 Instance 1	1			Inde	Subindex	Object Name	Size (b		
0x6650	2	Axis 1 SS1 Instance 2	1			0x206	0	Axis 1 SBA	1		
0x6650	3	Axis 1 SS1 Instance 3	1		1	0x206	0	Axis 1 SBC	1		
0x6653	0	Axis 1 SLA	1			0x206	0	Axis 1 Temperature warning	1		
0x6668	D	Axis 1 SOS command	1			0x206	0	Axis 1 SLA	1		
0x6670	1	Axis 1 SS2 Instance 1	1			0x206	0	Axis 1 SLA	1		
0x6670	2	Axis 1 SS2 Instance 2	1			0x661	0	Axis 1 safe position actual value	ie 16		
0x6680	1	Axis 1 SSR Instance 1	1			0x661	0	Axis 1 safe velocity actual valu	e 16		

See How To Set Up a Variable to Monitor Safety Data for step-by-step instructions.

7.12.5 Distributed Clock tab

The contents of this tab allow you to change settings related to the Distributed Clock for all Kollmorgen and third-party devices (both discovered and manually added).



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 chosen such that it coincides with other input slaves in the EtherCAT system.
Master Cycle Time	Base interval in microseconds, which will be used by the master. This is changed and automatically updated by changing the Cycle Time value on the EtherCAT Master Settings Tab tab.

Element	Description
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 will be available under the following scenarios:

- The Sync 0 or Sync 1 parameter is not present
- Distributed Clock is not supported by the device
- The ESI file is missing.

7.12.5.1 Oversampling devices

Some EtherCAT devices have oversampling features. An oversampling device is typically able to record (input) or provide (output) signals at a higher rate than the EtherCAT cycle time. This rate is called the *oversampling factor*. For example, with 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 in order to achieve the higher rate. Each of these PDO objects corresponds to one sample. For example, an output device with an oversampling factor of 4 will typically have 4 PDO objects: Output 1, Output 2, Output 3 and Output 4.

The oversampling factor is tied to the Distributed Clock Profile. A warning is displayed next to the DC profile selection box in the Distributed Clocks tab when a device has oversampling features, as seen below.

Distributed Clock Profile: Master Cycle Time:	2 Ch 10 times oversampling	
Distributed Clock Profile Se		
Sync Unit 0	3	í.
Cycle Time:	100 µs (Master Cycle Time / 10)	ŝ
Shift Time:	0 µs (User Defined)	c
Sync Unit 1	1	1
Cycle Time:	1000 µs (Master Cycle Time * 1)	l
Shift Time:	0 µs (User Defined)	Ì
Changing the Distributed Cloo See the PDO Selection/Mappi	k Profile will automatically modify the PDO selection and clear the mapped PLC variables that are no longer valid. ng tab for details.	

Figure 5-117: Example of a device with oversampling.

Changing the Distributed Clock Profile will automatically change the PDO selection. All PDOs corresponding to the selected Distributed Clock Profile and its according oversampling factor will be selected. These maps will be discarded if one or several PLC variables were mapped to a PDO that is no longer selected.

7.12.6 CoE Init Commands tab

This tab displays the EtherCAT device's CoE Init commands. The Init commands are grouped based on the EtherCAT transition state. 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 .

📥 AKD	_1 (AKD)	CoE Init	-Commands		
Pre Operation	al -> Safe Ope	rational			
Index	Subindex	Value	Comment	Direction	Sourc
0x6060	0	0x07	Opmode	Read	ESI File
0x60C2	1	0x01	Cycle time	Read	ESI File
0x60C2	2	0xFD	Cycle exp	Read	ESI File
0x3460	1	0x00	Latching engine 0 config to FI0, CAP0.TRIGGER=0, 0x3460-1:=0 (1 byte)	Read	ESI File
0x3460	2	0x01	Latching engine 1 config to FI1, CAP1.TRIGGER=1, 0x3460-2:=1 (1 byte)	Read	ESI File
0x60FE	2	0x30000	Digital Outputs Mask, 0x60fe:=0x30000 (4 bytes)	Read	ESI File
0x36E6	0	0x01	Set FBUS.PARAM02 to activate synchronization with the interrupt	Read	ESI File
0x36E8	0	0x01	Set FBUS.PARAM04 to disabled the drive on a motion error	Read	ESI File
0x3506	0	0x01	Set DRV.HWENMODE to disabled the rising edge of the hardware ena	Read	ESI File
0x35CA	0	1048576	Set UNIT.PIN to set gear IN to the correct unit conversion.	Write	ESI File
0x365F	0	0x00	Set UNIT.VROTARY to set the velocity units to RPM.	Read	ESI File
0x3460	3	0x02	Capture mode to distributed clock time (DCT), CAP0.MODE=2, 0x3460	Read	ESI File
0x3460	4	0x02	Capture mode to distributed clock time (DCT), CAP1.MODE=2, 0x3460	Read	ESI File
0x50E2	0	0x3E8	Set IL.KBUSFF (1.0 = 1000) UINT32	Read	ESI File
0x3498	0	01	Set FBUS.PROTECTION (available only since FW 01-07-03-000)	Read	ESI File
Safe Operatio	nal -> Operati	onal			
Index	Subindex	Value	Comment	Direction	Sourc
0x6040	0	0x80	Control word reset (I)	Read	ESI File
0x6040	0	0x00	Control word reset (II)	Read	ESI File

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 will not be displayed.

Field	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. There are two possible values.
	ESI File denotes that the command comes from the ESI fileUser denotes that the command is created by the user.

Function	Description
Add	Clicking Add opens a dialog which allows you to create a new Init Command. See Adding CoE Commands for more information.

Function	Description	
Edit	entry and click	lows you to edit an existing Init Command's attributes. Select an this button or double-click an entry. The dialog allows you to edit the including the Value , Comment , and Transition .
		mmand
	Init Command	
	Index: Subindex:	0x3460
	Direction:	⊥ Write
	- Init Command Data	
	Value (Decimal):	0
	Type:	USINT
	Comment:	engine 0 config to FI0, CAP0.TRIGGER=0, 0x3460-1:=0 (1 byte)
	Transition:	Pre Operational -> Safe Operational 🗸
		OK Cancel
Delete	Delete the sele	ected Init Command row.

7.12.6.1 Adding CoE Commands

This command allows you to construct a new Init Command by supplying Index, Subindex, Value, Comment, Transition, and Direction attributes. The dialog lists the device's object dictionary and the objects are filtered based on the **Direction** currently selected, e.g. if "Write" is selected then the Read-only objects are not shown.

	Command x (Hexadecim	al):	0x			
	ndex (Decima		0			
	tion:	y.	Write 👻			•
CoE	Object-Dict					
	Index	Value	Name	Туре	Flags	
	0x0002	-	DummySINT	SINT	RX (WO)	
	0x1003	0	Pre-defined error	USINT	(RW RW RW)	
	0x1005	128	COB-ID SYNC	UDINT	(RW RW RW)	
	0x1006	0	Communication c	UDINT	(RW RW RW)	
	0x100C	0	Guard time	UINT	(RW RW RW)	
-						-

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 you specified are already used) then you are presented with a dialog which allows you to specify the **Value**, a **Comment**, and the **Transition**.

- The Transition may be **Pre Operational -> Safe Operational** or **Safe Operational -> Operational**.
- If the **Type** is "STRING (20)" or "STRING (50)", you may only enter up to 20 or 50 characters.

Clicking **OK** adds the command to the appropriate transition group in the CoE Init Commands tab and setes the Source attribute to **User**.

🗢 Add/Edit CoE Init-C	ommand	? ×
Init Command		
Index:	0x0002	
Subindex:	0	
Direction:	Write	
Init Command Dat	1	
Value (Decimal):	0	
Type:	SINT	
Comment:	DummySINT	
Transition:	Pre Operational -> Safe Operational	•
	ОК	Cancel

If the Init Command data type is an array of bytes, then 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.

Init Command	
Index:	0x2000
Subindex:	3
Direction:	Read
Value (HexBinary): Type:	00 01 ARRAY [02] OF BYTE
Comment:	Reply
Transition:	Pre Operational -> Safe Operational
	OK Cancel

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) presents you with a dialog which allows you to specify a **Value**, **Comment**, and **Transition**.

- The Value must be entered in HexBinary format as shown below.
- The Transition may be Pre Operational -> Safe Operational or Safe Operational -> Operational.

Clicking **OK** adds the command to the appropriate transition group in the CoE Init Commands tab and sets the Source attribute to **User**.

Init Command	
Index:	0x3000
Subindex:	0
Direction:	Write
- Init Command Data Value (HexBinary):	00 01 02 03
Type:	Raw data
Comment:	
Transition:	Pre Operational -> Safe Operational 🗸
	OK Cancel

7.12.7 CoE Object Dictionary Tab

This tab displays the 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 fields in the table are described below. 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

NOTE

This tab is grayed out if the ESI file does not have CoE object information, or if the file is unavailable.

General Properties		Distributed Clock		CoE Init-Commands		CoE Object-Dictionary		
Coupler_3 (AKT-ECT-000) CoE Object-Dictional								
-		Jei_3		LC1-000)	COL	objec	CDICCIONAL	y
	Index	Value		Vame	Ţ	ype	Flags	
\triangleright	0x6000	8	AKT-DNx-	008-000_In	USINT		(RO RO RO	D)
۵	0x7010	8	AKT-DT-0	08-000_Out	USINT		(RO RO RO	D)
	Subindex	Value		Name	т	ype	Flags	
	1	-	K1 Ausga	ng	BOOL		RX (RO RO F	10
	2	-	K2 Ausga	ng	BOOL		RX (RO RO F	0
	3	-	K3 Ausga	ng	BOOL		RX (RO RO F	10
	4	-	K4 Ausga	ng	BOOL		RX (RO RO F	0
	5	-	K5 Ausga	ng	BOOL		RX (RO RO F	10
	6	-	K6 Ausga	ng	BOOL		RX (RO RO F	0)
	7	-	K7 Ausga	ng	BOOL		RX (RO RO F	10
	8	-	K8 Ausga	ng	BOOL		RX (RO RO F	0

CoE objects can be composed of a simple data type or a complex type. In case of a complex data type, each simple data type composing it has a SubIndex . As shown in the example below, 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)
I	Subindex	Name	Value	Туре	Flags
	1	Max. number of messages	32	USINT	(RO RO RO)
I	2	Latest message	0	USINT	(RO RO RO)
	3	Latest acknowledged message	0	USINT	(RW RW RW)
I	4	New active message	0	USINT	TX (RO RO RO)
	5	Control bits	0	UINT	(RW RW RW)

Field	Description
Index / SubIndex	CoE Index or SubIndex number
Name	Name of the CoE object
Value	Default value of the CoE object
Туре	Data type of the CoE object

Field	Descrip	otion
Flags	This column contains 5 values in the format XX YY (AA BB CC) , each representing an access option.	
	The valu	ues 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.	
	ΥY	Specifies if the CoE objecct can be mapped as TxPDO. It is represented as T x if it can be mapped.
	The values for AA, BB, and CC provide the CiE Access type. The valu be read only (RO), read-write (RW), or write only (WO)	
	AA BB CC	Access type of the object in PREOP state Access type of the object in SAFEOP state Access type of the object in OP state
Module	Name o	f the module that is associated with the CoE Object

8 Understanding KAS

This chapter gives explanation about the most important **concepts** that you need to understand to use KAS.

IMPORTANT

To take full advantage of KAS functions, a basic understanding of automation (programming languages and motion control) is required.

8.1 IEC 61131-3

8.1.1 Introduction

To create programs for the implementation of the PLC part of your application, the KAS IDE complies with IEC 61131-3. This standard currently defines five programming languages for programmable control systems.

The KAS IDE implements this standard to provide you with well-defined and well-known programming languages.

8.1.2 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, which helps to prevent errors early on in the programming phase. This avoids for example 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).

Different kinds of Data types

Common data types are Boolean, Integer, Real, Byte, Word, Date, Time_of_Day, and String. Based on these, you can define your own personal data types, known as derived data types. In this way you can define an analog input channel as a data type, and re-use it.

List of Data types

Below are the available basic data types:

Types	Description	Values	Prefixes
BOOL	Boolean (bit)	FALSE or TRUE - stored in 1 byte	
SINT	Small signed integer in 8 bits	-128 to +127	SINT#
USINT	Small unsigned integer in 8 bits	0 to +255	USINT#
BYTE	<i>Same as USINT</i>		
INT	Signed integer in 16 bits	-32768 to +32767	INT#
UINT	Unsigned integer in 16 bits	0 to +65535	UINT#
WORD	<i>Same as UINT</i>		
DINT	Signed double precision integer in 32 bits	-2147483648 to +2147483647	
UDINT	Unsigned integer in 32 bits	0 to +4294967295	UDINT#
DWORD	<i>Same as UDINT</i>		

Types	Description	Values	Prefixes
LINT	Long signed integer in 64 bits		LINT#
ULINT	Long unsigned integer in 64 bits		ULINT#
LWORD	Same as ULINT		
REAL [‡]	Single precision floating point stored in 32 bits	-3.4E38 to 3.4E38 and -3.4E-38 to 3.4E-38 (6 to 7 significant digits of accuracy)	
LREAL [‡]	Double precision floating point stored in 64 bits	-1.7E308 to 1.7E308 and -1.7E- 308 to 1.7E-308 (14 to 15 significant digits of accuracy)	LREAL#
TIME	Time data type is used to specify a time variable - accuracy is 1ms. See TIME for more information.	0ms to 24h	T# or TIME#
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	

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#3.141592653589793238.

IMPORTANT

^{‡‡} REAL is restrictive, but because it is the default, it is recommended to explicitly declare your real constants with the **LREAL#** prefix.

🗙 TIP

You can use **2#,8#** or **16#** prefixes to specify an integer in binary, octal or hexadecimal basis respectively.

8.1.2.1 Structures

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. In that case, corresponding members of all declared variables having this structure type will be initialized with the initial value of the member.

To specify a member of a structured variable in PLC languages, use the following notation:

```
VariableName.MemberName
```

8.1.2.1.1 Limitation

If a member of a structure is an instance of another structure, the nested structure must be declared **BEFORE** in the list.

8.1.2.2 Arrays

356

You can declare arrays for internal variables by specifying dimension(s).

To declare an array, enter the number of elements in the **Dim.** column of the **Dictionary** (see procedure here).

16 Name	Туре	Dim.	Attrib.
NewVar	BOOL		
😑 🗋 Mac	hineLogic	2,10,	4 🚍
NewVar	BOOL	[01,09,0	03]

For a multi-dimensional array (note that arrays have at most three dimensions), enter the number of elements for each dimension separated by commas (for example **2,10,4** is a 3 dimensional array, the first dimension has 2 elements, the second dimension has 10 elements, and the third dimension has 4 elements).

() IMPORTANT

All indexes are 0 based. For example, in the case of a single dimension array, the first element is always identified by *ArrayName*[0].

8.1.3 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.

8.1.3.1 About Retain Variables

A retain variable is a PLC variable which:

- is non-volatile: stored in non-volatile controller memory (AKD PDMM or PCMM). When using 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 is started the retained values are used. Additionally, retains are auto-saved every two seconds. This applies to both the AKD PDMM and PCMM models.
- Variables stored in non-volatile memory that have changed will be updated every two seconds. Unchanged variables are not updated.

Such a variable is used to store application specific data, like for instance to count a cutting-edge cycle in order to stop for its blade replacement after a specific number of iterations.

- On an application "Start", KAS initializes the retain variables with the value stored in the NVRAM.
- On an application "Cold Start", KAS initializes the retain variables with their default value.
- 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 default value.

Two parameters are checked to identify if the declaration changed:

- The number of variables of each type
- The length of a STRING type variables

① IMPORTANT

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 will be logged and the data will not be stored in non-volatile memory. See NVRAM for more information.

For the KAS Runtime Simulator, the retained variables are saved in a file in your project repository.

NOTE

The following actions will reset retained value(s) to their Init value(s):

- Changing the type of a retain variable
- Changing the length of a string retain variable
- Changing the size of an array variable
- Changing any element of a structure variable

8.1.3.2 Working with Variables

All variables used in programs must be first declared in the variable editor. Each variable belongs to a group and must be identified by a unique name within its group.

8.1.3.2.1 Groups

A group is a set of variables. A group either refers to a physical class of variables, or identifies the variables local to a program or user-defined function block. Below are the possible groups:

Groups	Description
GLOBAL	Internal variables known by all programs
RETAIN	Non volatile internal variables known by all programs
%I	Channels of an input board - variables with same data type linked to a physical input device
%Q	Channels of an output board - variables with same data type linked to a physical output device
PROGRAMXXX	All internal variables local to a program (the name of the group is the name of the program)
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)

8.1.3.2.2 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 the later case, 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.

Refer to the list of available data types for more information. Refer to the section describing function blocks for further information about how to use a function instance.

Additionally, you can specify dimension(s) for an internal variable, in order to declare an array.

8.1.3.2.3 Naming a variable

A variable must be identified by a unique name within its parent group. The variable name cannot be a reserved keyword of the programming languages and cannot have the same name as a standard 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 ("_") mark, followed by letters, digits or underscore marks. It is not allowed to put two consecutive underscores within a variable name. Naming is case-insensitive. Two names with different cases are considered as the same.

%IX	1 byte input - BOOL or SINT
%QX	1 byte output - BOOL or SINT
%IW	2 bytes input - INT
%QW	2 bytes output - INT
%ID	4 bytes input - DINT or REAL
%QD	4 bytes input - DINT or REAL
%IL	8 bytes input - LINT or LEAL
%QL	8 bytes output - LINT or LEAL
%IS	STRING input
%QS	STRING output

8.1.3.2.4 Attributes of a variable

Physical I/Os are marked as either "**Input**" or "**Output**". Each internal variable can be configured as Read/Write or **Read Only**. Read Only variables can be mapped to Outputs, but not to Inputs. This is because Inputs can change state and a Read Only variable would not be able to change its value to match the input state.

Parameters of User-Defined Function Blocks are marked as either INor OUT.

8.1.3.3 Retain Variables

What is a retain variable?

A retain variable is a PLC variable which:

- is non-volatile: stored in non-volatile controller memory (AKD PDMM or PCMM). When using 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 is started the retained values are used. Additionally, retains are auto-saved every two seconds. This applies to both the AKD PDMM and PCMM models.
- Variables stored in non-volatile memory that have changed will be updated every two seconds. Unchanged variables are not updated.

Such a variable is used to store application specific data, like for instance to count a cutting-edge cycle in order to stop for its blade replacement after a specific number of iterations.

- On an application "Start", KAS initializes the retain variables with the value stored in the NVRAM.
- On an application "Cold Start", KAS initializes the retain variables with their default value.
- 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 default value.

Two parameters are checked to identify if the declaration changed:

- The number of variables of each type
- The length of a STRING type variables

IMPORTANT

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 will be logged and the data will not be stored in non-volatile memory. See NVRAM for more information.

For the KAS Runtime Simulator, the retained variables are saved in a file in your project repository.

NOTE

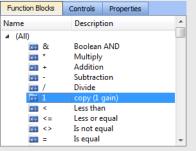
The following actions will reset retained value(s) to their Init value(s):

- Changing the type of a retain variable
- Changing the length of a string retain variable
- Changing the size of an array variable
- Changing any element of a structure variable

🗙 TIP

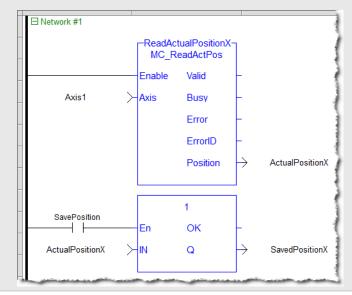
Retained variables should not be used as the output of Function Blocks. Doing so may cause problems with retaining the value if the FB is executed with the Enable input off. The output structure can cause it to be conditionally updated to zero or the old value.

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.



Work-around example

In the following image ReadActualPositionX is continually reading position feedback from an axis, and the copy block is saving a specific value to a retain variable. ActualPositionX is the intermediate variable and SavedPositionX is the retain variable.



8.1.4 Constant Expressions

360

Constant expressions can be used in all languages for assigning a variable with a value. All constant expressions have a well-defined data type according to their semantics. If you program an operation between variables and constant expressions having inconsistent data types, it will lead to syntactic errors when the program is compiled. Below are the syntactic rules for constant expressions according to possible data types:

Туре	Prefix	Description
BOOL		Boolean . There are only two possible Boolean constant expressions. They are reserved keywords $^{\rm TRUE}$ and $^{\rm FALSE}$.
SINT	SINT#	Small (8 bit) Integer . Small integer constant expressions are valid integer values (between -128 and 127) and must be prefixed with $SINT#$. All integer expressions having no prefix are considered as DINT integers.
USINT/BYTE	USINT#	Unsigned 8 bit Integer . Unsigned small integer constant expressions are valid integer values (between 0 and 255) and must be prefixed with $USINT#$. All integer expressions having no prefix are considered as DINT integers.
INT	INT#	16 bit Integer . 16 bit integer constant expressions are valid integer values (between -32768 and 32767) and must be prefixed with $INT#$. All integer expressions having no prefix are considered as DINT integers.
UINT/WORD	UINT#	Unsigned 16 bit Integer . Unsigned 16 bit integer constant expressions are valid integer values (between 0 and +65535) and must be prefixed with $UINT#$. All integer expressions having no prefix are considered as DINT integers.
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.
		You can use 2# , 8# or 16# prefixes to specify an integer in binary, octal or hexadecimal basis respectively.
UDINT/DWORD	UDINT#	Unsigned 32 bit Integer . Unsigned 32 bit integer constant expressions are valid integer values (between 0 and 4294967295) and must be prefixed with UDINT#. All integer expressions having no prefix are considered as DINT integers.
LINT	LINT#	Long (64 bit) Integer . Long integer constant expressions are valid integer values and must be prefixed with LINT#. All integer expressions having no prefix are considered as DINT integers.
ULINT/LWORD	ULINT#	Unsigned 64-bit integer constant expressions are valid integer values. All integer expressions having no prefix are considered as DINT integers.
REAL		Single precision floating point value . Real constant expressions must be valid numbers, and must include a dot ("."). If you need to enter a real expression having an integer value, add ".0" at the end of the number. You can use "F" or "E" separators for specifying the exponent in case of a scientific representation. REAL is the default precision for floating points: such expressions do not require a prefix.
		REAL is restrictive, but because it is the default, it is recommended to explicitly declare your real constants with the LREAL# prefix. REAL constants are limited to 6-7 digits of accuracy. Any digits after these significant digits will be lost, leading to a loss of precision.

Туре	Prefix	Description	
LREAL	LREAL#	expression You can use	cision floating point value . Real constant expressions must nbers, must include a dot ("."). If you need to enter a real having an integer value, add " .0 " at the end of the number. " F " or " E " separators for specifying the exponent in case of epresentation.
			tants are limited to 14-15 digits of accuracy. Any digits after icant digits will be lost, leading to a loss of precision.
TIME	T# or TIME#	be less than T#. They ar of minutes	y . Time-constant expressions represent durations that must 24 hours. Expressions must be prefixed by either TIME# or e expressed as a number of hours followed by " h ", a number followed by " m ", a number of seconds followed by " s ", and a milliseconds followed by " ms ".
		respecte You cann	r of units (hour, minutes, seconds, milliseconds) must be d. ot insert blank characters in the time expression. ust be at least one valid unit letter in the expression.
STRING		quote mark You can use	string. String expressions must be written between single s. The length of the string cannot exceed 255 characters. the following sequences to represent a special or not- naracter within a string:
		Sequence	Description
		\$\$	a "\$" character
		\$'	a single quote
		\$T	a tab stop (ASCII code 9)
		\$R	a carriage return character (ASCII code 13)
		\$L	a line feed character (ASCII code 10)
		\$N	carriage return plus line feed characters (ASCII codes 13 and 10)
		\$P	a page break character (ASCII code 12)
		\$xx	any character (xx is the ASCII code expressed on two hexadecimal digits

Table 6-1: List of Prefixes for Constant expressions

8.1.4.1 Examples

Below are some examples of valid constant expressions:

TRUE	TRUE Boolean expression
FALSE	FALSE Boolean expression
SINT#127	small integer
INT#2000	16 bit integer
123456	DINT (32 bit) integer
16#abcd	DINT integer in hexadecimal basis
8#34712	DINT integer in octal basis
2#1000100	DINT integer in binary basis
LINT#1	long (64 bit) integer having the value "1"

0.0	0 expressed as a REAL number
1.002E3	1002 expressed as a REAL number in scientist format
LREAL#1E-200	Double precision real number
T#23h59m59s999ms	maximum TIME value
TIME#0s	null TIME value
T#1h123ms	TIME value with some units missing
'hello'	character string
'name\$Tage'	character string with two words separated by a tab
'I\$'m here'	character string with a quote inside (I'm here)
'x\$00y'	character string with two characters separated by a null character (ASCII code 0)

Below are some examples of typical errors in constant expressions

BooVar := 1;	0 and 1 cannot be used for Booleans
1a2b	basis prefix ("16#") omitted
1E-200	"LREAL#" prefix omitted for a double precision float
T#12	Time unit missing
'I'm here'	quote within a string with "\$" mark omitted
hello	quotes omitted around a character string

Additionally, there are pre-defined constants. See Use the Defines List for information about Internal and user-defined Defines.

8.1.5 Program Organization Units

Within IEC 61131-3, the Functions, Function Blocks, and Programs are called Program Organization Units (POU).

In addition to the IEC standard, you can write you own code: sub-program or UDFB.

Туреѕ	IEC 61131-3	Written by end-user
Basic functions (has no memory)	Functions	Programs / Sub-programs
Instantiated functions (keep track of the past)	Function Blocks (FB)	User-Defined Function Blocks (UDFB)

8.1.5.1 Difference between Functions and Function Blocks

- Functions are expected to complete in one cycle
- Function Blocks can take several cycles to complete

8.1.5.1.1 Description of FB operation

Rather than halt the application, waiting for operations to complete, the FB typically gives control back to the application but does not set its **Done** output.

8.1.5.1.2 Examples of Operations Overrunning the Cycle Duration

- A motion command to move from one location to another can take several cycles to complete.
- Same for operations like reading/writing to files or reading and writing over TCP/IP can also take several cycles to complete.

8.1.5.1.3 Operation Sequence

- 1. When a 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
- 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

8.1.5.2 Functions

IEC has defined standard functions and also allows you to create your own functions (called userdefined 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 the following 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
```

8.1.5.3 Function Blocks

364

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.

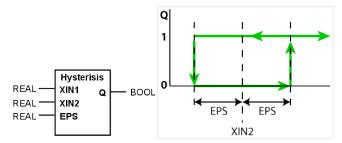
They contain data as well as an algorithm, so they can keep track of the past (which is one of the differences from Functions). They have a well-defined interface and hidden internals, like an IC or a black box. The user only sees the interface, being the inputs and outputs. The code itself is hidden.

Function Blocks can be used in any of the IEC languages. Note that in an SFC program, function blocks can be part of a step or transition created in FFLD, ST, IL and FBD.

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 also additional function blocks that can come from libraries produced by you or other suppliers (e.g. a temperature controlloop or PID).

Example of function blocks



The function block is based on the programming language function block Diagram and has the name Hysterisis. 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.

★ TIP

Input names are not very usable. Please use meaningful names.

Internally, the FB contains the following 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 the Structured Text 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, it 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.

8.1.5.4 Programs

With the above-mentioned 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.

8.1.5.4.1 Sub-programs

The list of programs is completed with "Sub-programs". Sub-programs are written in FBD, FFLD, ST or IL languages, and can be called by the programs of the application. 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 also Application Software Structure - Definitions.

8.1.5.4.2 Program Guidelines

An application is a list of programs. Programs are executed sequentially within the target cycle, according to the following 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, ST or IL between two SFC programs). Each program is entered using a language chosen when the program is created. Possible languages are Sequential Function Chart (SFC), Function Block Diagram (FBD), Free Form Ladder Diagram (FFLD), Structured Text (ST) or Instruction List (IL). The maximum number of programs in an application is 32767.

8.1.5.4.2.1 Program Name

366

Programs must have unique names. The name cannot be a reserved keyword of the programming languages and cannot have the same name as a standard or "C" Function or function block. A program must not 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.

8.1.5.4.2.2 Child SFC Programs

You can define a hierarchy of SFC programs, entered as a tree in the list of programs. A child program is controlled within action blocks of the parent SFC program.

🗙 TIP

Even if you do not want to split your FFLD program, at least separate FFLD from SFC. Simply make a sub-program in FFLD called from the SFC step, and keep only the state machine in the SFC program. This makes everything simpler and more comfortable for editing and debugging.

8.1.5.4.3 Program Limitations

When creating your application you have to consider the following important limitations.

For 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

For **FFLD** programs:

- Width of any network is limited to 255 columns
- Height of any network is limited to 255 rows

For any program, sub-program or UDFB written in **other** languages:

- Jump limit is 64kB For example, in a Free Form Ladder program, if you create a UDFB or program which is over 64kB and then decide to 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

8.1.5.5 User-Defined Function Blocks

The list of programs is completed with "User-Defined Function Blocks" (UDFBs). UDFBs are described using SFC, FBD, FFLD, ST or IL languages, and 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.

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. Note that the called UDFB must be declared <u>before</u> 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.

8.1.6 Programming Languages

Within 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 two textual and three graphical versions:

Textual:

- Instruction List (IL)
- Structured Text (ST)

Graphical

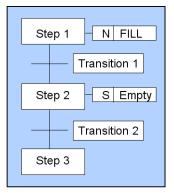
- Sequential Function Chart (SFC)
- Free Form Ladder Diagram (FFLD)
- Function Block Diagram (FBD)

All five languages are interlinked: they provide a common suite. The choice of programming language depends on:

- the programmer's background
- the problem at hand
- the level of describing the problem
- the structure of the control system
- the interface to other people / departments

8.1.6.1 Sequential Function Chart (SFC)

SFC describes graphically the sequential behavior of a control program. It is derived from Petri Nets.



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.

Alternative and Parallel Sequences

368

You can use alternative sequences and even parallel sequences, like those commonly required in batch applications. For example, one sequence is used for the primary process, and the second for monitoring the overall operating constraints.

As shown in the following picture, parallel sequences are also possible:

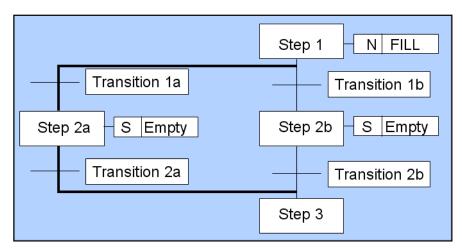


Figure 6-1: Example of a Parallel Sequence in SFC

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.

8.1.6.2 Structured Text (ST)

ST is a very 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. It can be used for the definition of complex function blocks, which can be used within any of the other languages.

8.1.6.3 Function Block Diagram (FBD)

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.

8.1.6.4 Free Form Ladder Diagram (FFLD)

FFLD is based on the graphical presentation of Relay Ladder Logic.

8.1.6.5 Instruction List (IL)

IL is the European counterpart of FFLD. As textual language, it looks like Assembler.

8.1.7 Alias Definitions

The compiler supports the definition of aliases (see usage in Use the Defines List). An alias is a unique identifier that can be used in programs to replace another text. Definitions are typically used to replace a constant expression and facilitate the maintenance of programs.

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 according to the following syntax:

#define Identifier Equivalence (* comments *)

Below are some 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. Below is an example:

```
#define PI 3.14
#define TWOPI (PI * 2.0)
```

Note that a definition can be empty, for 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.

8.1.8 Handling Exceptions

The compiler enables you to write your own exception programs for handling particular system events. The following exceptions can be handled:

- Startup Exceptions (before the first cycle)
- Shutdown Exceptions (after the last cycle)
- Division By Zero Exceptions

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. You then have the selection between the following possibilities:

- Return without any special call. In that case the standard handling will be 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. Just put a breakpoint in your handler. When stopped, the call stack will show you the location of the division in the source code of the program.

8.1.8.1 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 the following line:

#OnStartup ProgramName

NOTE

Warning: The program is executed before all other programs within the fisrt cycle. This implies that the cycle timing can be longer during the first cycle. You cannot put breakpoints in the Startup program.

8.1.8.2 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 will handle the exception. It cannot be an SFC program.
- 2. In the editor of global defines, insert the following line:

#OnShutdown ProgramName

NOTE

Warning: You cannot put breakpoints in the Shutdown program.

8.1.8.3 Division By Zero Exceptions

You can write your own exception program for handling the "Division by zero" exception. Below is the procedure you must follow for setting an exception handler:

- 1. Create a new sub-program without any parameter that will handle the exception
- 2. In the editor of global defines, insert the following line:

#OnDivZero SubProgramName

8.2 Motion Concepts

8.2.1 Introducing Motion

8.2.1.1 Motion Control Main Functions

To ensure accurate positioning and movement, motion control consists of the two following main parts:

- Setpoint generation
- Regulation

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.

Regulation

Even using the very best drives, you cannot maintain accurate positioning without a feedback loop. The regulation consists of following the generated position settings using classical feed-forward or feedback control-loops (by means of PID). 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.

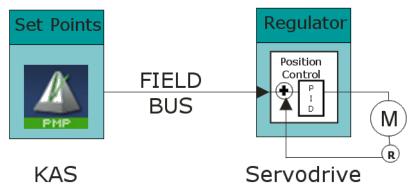


Figure 6-2: Regulation with Remote Drive

8.2.1.2 Single and Multi-Axis Motion

In **Single-Axis**, as shown in the figures above, 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, like a physical motor (called master) or an internal profile generator (called virtual master) as shown in Multi-Axis Driven by a Virtual Master below.

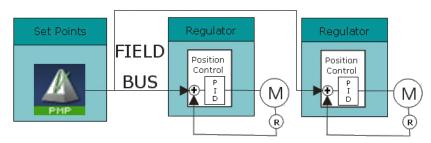
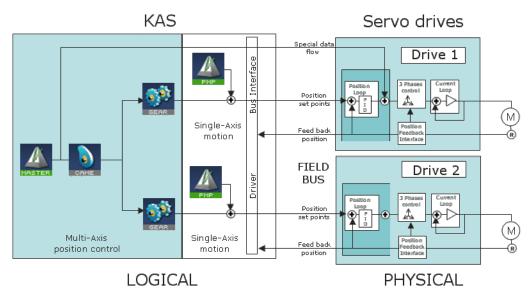


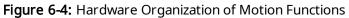
Figure 6-3: Multi-Axis Driven by a Virtual Master

8.2.1.3 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.



One of the possible configurations is represented in the figure below.



The scope of Kollmorgen Automation Suite is to manage all the logical parts of the motion control and to 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.

Therefore, we do no longer consider regulation and the physical world in the following paragraphs. Only setpoint generation are taken into account.

8.2.1.4 Motion Profile

In motion control, a common need is to move a system from one steady position to another (point-topoint motion). Following the fastest possible motion within an allowed maximum value for speed, acceleration, and jerk, results in a third-order motion profile as illustrated below:

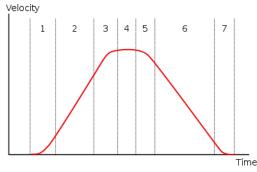


Figure 6-5: Third-order motion profile

The motion profile consists of up to seven phases defined by the following:

- 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

374

- constant maximum deceleration (zero jerk)
- deceleration decrease, approaching the desired position at zero velocity, with maximum positive jerk

If the initial and final positions are sufficiently close together, the maximum acceleration or maximum velocity may never be reached.

8.2.1.5 KAS Touch Probe & Fast Input Overview

This page provides links to KAS documentation sections which are helpful in configuring and using the Touch Probe feature, also known as "fast inputs" or "capture engine".

8.2.1.5.1 Configuration

8.2.1.5.1.1 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

8.2.1.5.1.2 PipeNetwork Configuration

- MLAxisCfgFastIn
- MLTrigSetEdge

8.2.1.5.1.3 PLCopen Configuration

- MC_TouchProbe
- MC_Reference
- MC_MachRegist

8.2.1.5.2 List of Function Blocks and Pipe Blocks

8.2.1.5.2.1 PLCopen Function Blocks

I/O

MC_TouchProbe

Reference

• MC_Reference

Registration

- MC_MachRegist
- MC_MarkRegist
- MC_StopRegist

8.2.1.5.2.2 PipeNetwork Blocks

Motion Library - Axis

- MLAxisCfgFastIn
- MLAxisIsTrigged
- MLAxisRstFastIn

Motion Library - Trigger

- MLTrigClearFlag
- MLTrigInit
- MLTrigIsTrigged
- MLTrigReadDelay
- MLTrigReadPos
- MLTrigReadTime
- MLTrigSetEdge
- MLTrigWriteDelay

8.2.2 Pipe Network or PLCopen

Using KAS there are two ways to generate motion functions and motion profiles: with Pipe Network or PLCopen.

Pipe Network

The Pipe Network enables you 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) and includes the definition of specific transformer motion profiles. This structure is then controlled from the PLC application using dedicated function blocks in the Motion Library.

To be able to use pipes correctly, it is necessary to first consider some definitions.

PLCopen (see PLCopen Web site)

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.

8.2.2.1 Motion Engine Differences

The following table outlines some of the main feature differences between the Pipe Network and the PLCopen motion engines. It also provides their associated function blocks.

Торіс	Pipe Network	PLCopen
Function block format	Begins with ML ex: MLAxisRel	Begins with MC_ ex: MC_MoveRelative
Does Function block requires instantiation?	No. Except for MLAxisStop	Most require it
Method to start execution	Most are level triggered	Most are edge triggered
Motion execution status, for function block executing motion	Use MLMotionStatus function block	Each function block includes a standard set of outputs for motion status
Function block standard input format	Requires additional function blocks to define motion parameters (speed, accel, decel, etc.)	Includes standard set of inputs to define motion (speed, accel, decel, etc.)
Axis setup method	Includes in the Pipe Network Axis block properties	Part of Axis definition screen in the Project tree



PLC open

notior

Торіс	Pipe Network	PLCopen
How the Axis name is setup?	Automatically done as part of Pipe Network Axis block properties	Create an instance of a Axis_Ref variable structure in the dictionary, then assign an axis number to it in a PLC program (for procedure, Create PLCopen Axis)
Is there additional motion editor?	Yes (Pipe Network editor)	No
Motion buffering	Execution of multiple motion commands in a row is handled by the programmer	Function blocks have built in buffering modes
Motion jerk reduction	Primarily available by adding cams to the Pipe Network	Function blocks have jerk reduction input

Table 6-2: Differences between the Pipe Network and PLCopen

8.2.3 Pipe Network Concept

To introduce the Pipe Network concept, we can use a mechanical analogy.

In the figure below, 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.

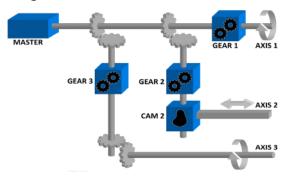


Figure 6-6: Mechanical System

The Pipe Network in the figure below corresponds to the mechanical system described above. The pipe concept is a one-to-one translation of a mechanical system into the logical world.

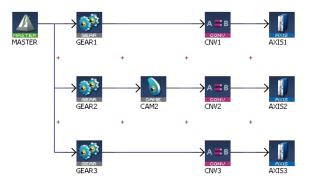


Figure 6-7: Pipe Network Structure

In our Pipe Network, the analogy is as follows:

- 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

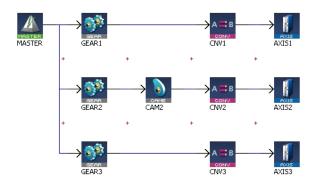
The Pipe Network concept allows motion engineers to define in a very natural way the physical relationships between the different axes of their machine.

This powerful modular approach provides a solution for almost any multi-axis requirements. It also remains open for new, additional functions that can be required in the future.

8.2.3.1 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 as shown in 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. In the application below there are 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, but you must compile your 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, as the parameters are entered into set-up screens.
 WPORTANT

Pipe Network code is generated automatically by the compiler, you should not try to modify it.

- In the programs, you can 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 is used for more than just coordinated motion. It contains a full library of single-axis motion commands for sections of an application where an axis operates independently.

8.2.3.2 Pipe

A pipe is a set of Pipe Blocks linked together (where position flows from one Pipe Block to the next). The general structure of a pipe is quite simple:

- 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



Figure 6-8: Typical Pipe Structure

★ TIP

To avoid jerk in the pipe network (which ultimately may cause a jerk in motor motion when a cam block is applied to the upstream pipe network positions) 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.

NOTE

More about the different kinds of Pipe Blocks are discussed in Adding Motion.

8.2.3.3 Pipe Block

Pipes are built using logical entities called Pipe Blocks.

A Pipe Block is an object whose purpose is to modify a flow of values with strict time constraints. Pipe Blocks normally have both input and output flows of values.

Based on their functions, there are four kinds of Pipe Blocks:

Function	Description
Input	Works as generator of values:
(source)	 sample external source objects or create a discrete flow of values as an input to the pipe
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
Output (convertor)	Block that can end a pipe:convert the incoming values from user units to correct system units for the destination objects
Destination	Simply models a physical axis of the machine

The following table provides a short description of each Pipe Block:

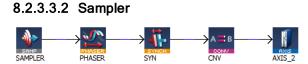
Function	Pipe Block	Description
Input	Master	Virtual master generating values (position) at each cycle
Input	Sampler	Samples external value (encoder, resolver, PLC variable etc.)
Transformation		
Mathematical	Derivator	Applies a derivation on the input data flow
Mathematical	Integrator	Integrates the input data flow
Mathematical	Adder	Adds two data flows
Event-driven	Synchronizer	Starts and stops a sub-pipe in a controlled way
Event-driven	Delay	Delay the data flow during some cycles
Event-driven	Comparator	Monitor the input data flow and detects the crossing of a particular value
Event-driven	Trigger	Computes the local pipe value from the timestamp of a Fast Input event
Modification	Cam	Applies a cam table (also called Cam Profile) to the input data flow
Modification	Gear	Applies a gearing ration on the input data flow
Modification	Phaser	Applies a phase offset to the input
Output	Convertor	Converts input data flow to a position and forwards it to an axis
Destination	Axis	Models a physical axis

Table 6-3: Pipe Network - List of Pipe Blocks

8.2.3.3.1 Master



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.



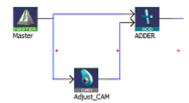
Use a Sampler Pipe Block to read an external encoder as an input signal into the Pipe Network or to directly read the actual position of another axis.

8.2.3.3.3 Gear

Use a Gearing Pipe Block to perform electronic gearing. The Gear Pipe Block allows gear ratios and the slope of the gear change to be initially set, then changed from within the application program.

8.2.3.3.4 Cam

Use a Cam Pipe Block to optimize the motion profile. Use an Adder block with a Cam block to dynamically change the distance moved during each period (or modulo) of motion.

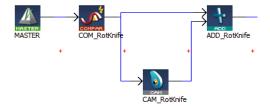


Cam Profiles are created using the cam creation tool.

8.2.3.3.5 Comparator

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.

The following example shows the changing of the offset move by changing the amplitude (or offset) of the Cam Pipe Block.



In a PLC application program, the MLCompWriteRef function block is used to arm the comparator block and MLCompCheck function block is used to check the position. By using condition statements in a user program, specific actions (such as 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.

8.2.3.3.6 Trigger



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.

8.2.3.3.7 Delay

Use a Delay Pipe Block to delay the flow of position through a Pipe Network. One potential use is to 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.

8.2.3.3.8 Phaser

Use a Phaser Pipe Block to perform a dynamic phase adjustment inside the Pipe Network. This block can be used to phase-advance or phase-retard a position as required to synchronize different motion elements on a machine.

8.2.3.3.9 Synchronizer

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.

8.2.3.3.10 Axis

Models the link from the Pipe Network to a physical axis.

8.2.3.3.11 Changing Information Flow from Position to Velocity

You can change the Pipe Network flow of information from position to velocity by using the **Convertor** Pipe Block. This Pipe Block is normally set up to receive position, so it must be changed to receive the expected input signal type as shown below:



Change the mode of Converter block to SPEED (and not POSITION mode).

8.2.3.4 Axis Pipe Block

Making the link between the logical and physical worlds, the Axis pipe block manages the data on positions.

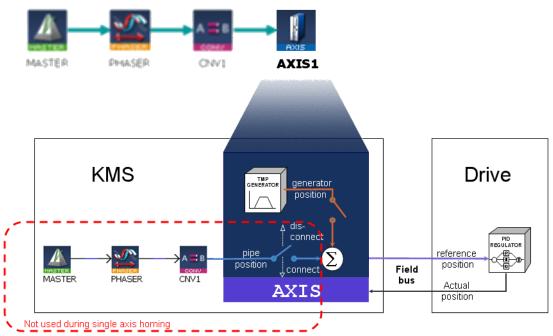


Figure 6-9: Axis Pipe Block Positions

8.2.3.4.1 About Associated Data on Positions

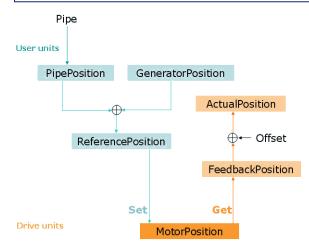
The following data are illustrated in the figure below.

NOTE

All positions are in user units with Modulo applied if active, unless specified.

Position / Offset	Description
ActualPosition	Actual refers to the actual position of the underlying Drive. It is the current position of the drive in user units. It is the sum of the feedback value (Position actual value) returned from the communication link to the drive, the Power ON Delta Offset , and any zero-offset due to an MLWritePos function (MLAxisWritePipPos, MLAxisWritePos). Normally the value of power on delta offset is zero.
CurrentPosition	Current position is the actual command value being sent to the drive. It is an unsigned 32-bit integer value (fraction = zero). When in the power on condition this value is the command value that represents the target value in the communication link (Position demand value). It is not in user units, but in Drive units of 2**20 units per revolution of the drive.
FeedbackPosition	Feedback Position is the "Position actual value" read from the drive. FeedbackPos relates to the TxPDO value of 'Actual position value'

Position / Offset	Description
GeneratorPosition	Generator position is the summation of all previous commands to the Axis internal trapezoidal motion generator. It is also a collector of uncompensated motion due to MLAxisWritePos being used to modify actual position via the zero offset value and the adjustment in commanded value to insure no steps in the Current position command. It also accumulates changes in pipe position due to activate and deactivation of the pipe and convertor output to pipe position of the axis.
MotorPosition	Motor position relates to the RxPDO value of 'Position demand value'
	MotorPosition = CurrentPos + PowerOnDeltaOffset
PipePosition	The output of the convertor block is written into the PipePosition value whenever the convertor block is connected to the axis and the pipe is active.
Power ON Delta Offset	A change was made a long time ago to allow absolute feedback to be passed into the axis rather than always starting at zero actual position. Units are in Drive units of 2**20 units per revolution. On Drive Power On this value is set to be the difference between the " ActualPosition value" and the "Position demand value" last sent to the drive. It is then added to the Current position value when the "Position demand value" is updated. It is read in User Units without periodicity applied.
ReferencePosition	Reference position is the summation of PipePosition and GeneratorPosition
	ReferencePosition = Pipe Position + Generator Position
Zero Offset	Affected by the MLAxisWritePos function to adjust the actual position to the desired value of the command by setting zero offset to the difference between the desired and actual position, and applying the change to modify the generator position so that the reference position tracks the change in reference.



User Units (abreviated UU) are the dimensions that will be 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. Some examples for this field are:

User Units	User Units per Revolution Value	Notes
Degrees	360.0	
Gradians	400.0	
Radians	3.14159265358979324	
Inches	15.7079632679489662	This equals pi*5 which is the circumference of a 5" diameter drum

KAS standard units for velocity and acceleration are UU/sec and UU/sec2.

8.2.3.4.2 Functions That View Axis Block Positions

MLAxisReadActPos	Returns the Actual Position	
MLAxisCmdPos	Returns the Reference Position	
MLAxisFBackPos	Returns the Feedback Position	
MLAxisGenPos	Returns the Generator Position	
MLAxisPipePos	Returns the PipePosition	

8.2.3.4.3 Functions That Change Axis Positions

MLAxisAdd	This function adds a relative distance to the current target Generator Position.
MLAxisRel	This function adds a relative distance to the current Generator Position.
MLAxisAbs	This function sets a new target Generator Position.
MLAxisMoveVel	This function sets the generator position moving at a programmed speed.
MLAxisStop	This function stops any current Generator Motion. It also causes the axis to start ignoring any changes in Pipe position to be added into the reference position. It decelerates, if moving, at a programmed rate.
MLAxisReAlign	Causes the Axis to move by a programmed amount without changing the Reference Position following an MLAxisStop. Also allows the Pipe Position to be used following an MLAxisStop.
MLAxisWritePos	If convertor is not connected, Zero Pipe Position and Pipe Offset. If convertor is connected (pipe active also), the pipe position and offset are left alone. The actual position is then set equal to the target position, and the Zero Offset is adjusted for no motion. The Reference position and Generator Positions are then realigned so that the new reference position creates no step in motion, with the lag between reference position and actual position being absorbed in the generator position.
MLAxisWritePipPos	Changes the pipe position to be the new value. May cause step motion.
MLCNVConnect	Initializes the pipe position to the Convertor block output value, and adjusts the axis Pipe Offset so that no jump in motion is generated.
MLCNVDisconnect	Stops sending the convertor output to the Pipe Position, and disconnects the convertor from the axis.
MLPipeAct	Starts calculating Pipe data and if the convertor block is connected to the axis it will reconnect the convertor and start updating the pipe position with any changes.
MLPipeDeact	Stops sending Pipe data to the Axis Block Pipe Position and disconnects the convertor output from the axis.

8.2.3.4.4 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
- The values are "aligned": ReferencePosition = Pipe Position + Generator Position

8.2.3.4.5 Axis Connection to a Pipe

A call to the MLPN_CONNECT Function or the MLCNVConnectMLCNVConnect function block is required to get motion generated in the pipe to the Axis

- Pipe Offset is calculated as follows: Pipe Offset = Pipe Position Reference Position
- The values are "aligned": Reference Position = Pipe Position + Generator Position

8.2.3.4.6 Realigning 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

8.2.3.4.7 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 Reference Position equals Pipe Position + Generator Position

8.2.3.4.8 Homing

Homing is the process of moving the motor to a known physical reference point on the machine.

Drive Homing: The AKD contains various pre-configured homing modes that avoid creating code. These home modes are drive-controlled and selected using the AKDHome function block .

Controller Homing: This homing type requires code in the application or UDFBs to perform the homing move.

Each axis is homed using MLAxis function blocks only (the Pipe Network is not used). Typically homing is done with MLAxisRel and MLAxisAbs to make motion and MLAxisWritePos to set a position offset.

8.2.3.4.9 Single-Axis Operation

This includes motion done on an individual axis: 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.

8.2.3.4.10 Multi-Axis Operation

388

For multi-axis applications, automatic operation requires motion synchronization between two or more axes and the Pipe Network is required to achieve the synchronization. To start up the Pipe Network the following two functions must be executed in an application program:

```
PipeNetwork(MLPN_ACTIVATE):
PipeNetwork(MLPN_CONNECT);
```

Multi-axis synchronized motion is then accomplished using a motion block associated with one of the three input Pipe Blocks:

- Master: MLMasterRun, MLMasterRel, and MLMasterAbs
- PMP: MLPmpAbs, MLPmpRel
- Sampler: MLSmpConnect, MLSmpConECAT, MLSmpConPNAxis, MLSmpConPLCAxis

8.2.3.4.11 Monitoring an axis

There are function blocks to monitor the performance and status of an axis. The key function blocks are as follows:

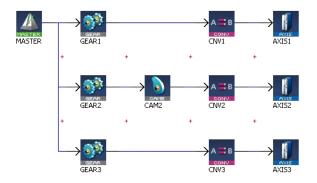
- MLAxisCmdPos The commanded position to the servo drive
- MLAxisReadActPos The actual position of the axis
- MLAxisStatus The status of the axis: enabled/disabled , bus connection, Pipe Network connection, drive executing an axis stop function, drive finished a stop
- MLAxisReadGenStatus The status of the Axis generator: acceleration, run, deceleration, change designation point, single step
- MLAxisGenIsRdy Is Axis generator ready

8.2.3.5 Executing Motion

Two types of Pipe Blocks are used to command motion in a Pipe Network: Axis block and Input block.

- Axis block starts motion directly on one axis.
- Input blocks start motion that affect all axes that are connected downstream in a Pipe Network. Input blocks can be one of three types:
 - Master Trapezoidal motion
 - PMP Parabolic Motion
 - Sampler Externally generated motion from another axis or external encoder

In the following example, executing MLAxisMoveVel, MLAxisAbs and MLAxisRel Functions can be used to cause motion on a particular axis. Whereas MLMstRun, MLMstAbs and MLMstRel functions cause motion on Axis1, Axis2 and Axis3.



For information on error management, Error Management.

For explanations on restarting the motion, refer to Restarting Motion with PipeNetwork

8.2.3.6 Pipe Block Lifetime

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).

Usage

As long as the pipe remains active, its values are cyclically calculated. Functions can be performed and events can be created.

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).

8.2.3.7 Motion State Machine

The Motion State Machine is driven by the IEC 61131-3 application with the help of dedicated function blocks.

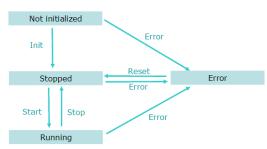


Figure 6-10: Motion State Machine

Each arrow represents a transition from one State to another.

8.2.3.8 Phase Execution in the Pipe Network

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.

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 zero. The phase of execution between two Pipe Networks cannot be specified by the user and depend on the pipe activation time of the application execution.

8.2.3.9 Use Motion Function Block for Pipe Network

Use motion library function blocks in your PLC application program to interface to the Pipe Network (to see the procedure Design Motion with Pipe Network).

ML function blocks are used to:

- 1. Create and initialize the Pipe Network
- 2. Perform motion at a single-axis or multi-axis level
- 3. Read information from points in the Pipe Network

For 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
- 4. Modify how the blocks work in the Pipe Network

For 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

8.2.3.9.1 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.

8.2.3.9.2 Motion Init

During initialization, the IEC 61131-3 application can create (by means of the MLMotionInit function) 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, in order to avoid memory allocation problems while running the application.

8.2.3.9.3 Motion Start

The Start method (MLMotionStart function) initializes the motion engine, motion bus driver, and initializes 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.

8.2.3.9.4 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:

392

- 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

For more details on Q output, refer to What is the difference between Q and OK?

8.2.3.10 Function - General rules

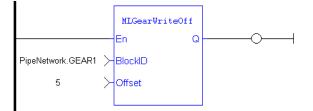
This section outlines rules for using ML function in the Pipe Network context.

8.2.3.10.1 Languages

Function that interact with the Pipe Network start with ML (for example MLAxisRel, MLPrfWriteOScale, or MLMstRel). These function can be used in all four of the 61131-3 PLC languages.

8.2.3.10.2 BlockID Inputs

The BlockID input is a DINT ID. It is the second input to a Pipe Network function when using FFLD:



The BlockID input is the first one if programming in Structured Text:

MLGearRatSlp(BlockID (*DINT*));

This input identifies the block in the Pipe Network that the function interacts with, and if using the graphical Pipe Network Editor the used variable starts with **PipeNetwork.xxx** (except if it is acting on a CAM profile, in which case the input is named ProfileID and the variable starts with **Profiles.xxx**).

🗙 TIP

As a general rule, when selecting a Pipe Block as the BlockID for a ML function, choose a Pipe Block with the same type which is in the name of the ML function.

For 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; and MLPrfxxx functions need a Profile entered for ProfileID, etc.

8.2.3.10.3 Output status

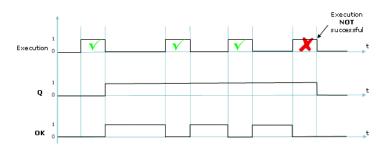
Most Pipe Network function have a default Boolean output labeled either Q or OK:



8.2.3.10.4 What is the difference between Q and OK?

OK returns true when function successfully executes.

Q output is initially set to 0 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. It remains to 0 until a successful execution resumes



8.2.3.10.5 When Q is set to True?

Some function change the Q output from low to high immediately after it starts executing, but others (including most functions that command motion) wait to change the output until the function has completely finished executing.

You need to check the description for each individual function block to be sure how its Q output is behaving.

8.2.3.10.6 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. For 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.

8.2.3.10.7 Missing input parameters

All inputs to a function must be entered in order for code to compile.

8.2.3.10.8 Position versus distance

Position is a value defined within a coordinate system.

DeltaPosition is a relative measure related to technical units. It is the difference between two positions.

8.2.3.10.9 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. But if a value is never set in a program the parameter entered during initialization is used.

For 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.

8.2.4 PLCopen®

The 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. The PLCopen function blocks for Motion Control, is based on IEC 61131-3 Function Blocks concept with the following factors in consideration:

- Simplicity ease of use for the application program builder and installation & maintenance
- Efficiency in the number of function blocks, for efficiency in design (and understanding)
- Consistency conforming to the IEC 61131-3 standard
- Universality hardware independent
- Flexibility future extensions / range of applications
- Completeness

KAS supports PLCopen motion in the following program formats: FFLD, SFC, ST,IL and FBD. PLCopen blocks in KAS start with "MC" (example: MC_MoveAbsolute). MC blocks are an alternative to using the ML Motion function blocks (example: MLAxisAbs) and associated Pipe Networks in many applications. Using MC Motion function blocks does not require a separate motion editor. Users who are familiar with PLCopen are automatically familiar with PLCopen inside the KAS IDE.

8.2.4.1 PLCopen Function Blocks

The following function block (FB) library is designed for the purpose of controlling one or more servo axes using the IEC 61131-3 PLCopen standard (for more details on FB, refer to Function Blocks).

To offer flexibility, ease of use and reusability, the library consists of command-oriented function blocks that have a reference to the axis, e.g. the abstract data type **Axis**

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 registration 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

MC_MoveAbsolute	performs a single-axis move to a specified endpoint position.
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_MoveAdditive	commands a controlled motion of a specified relative distance. Can also be used to interrupt a motion currently being performed. In this case the MotionAdditive FB causes the speed, acceleration, and deceleration of the motion already running to be changed to the parameters specified in the MC_MoveAdditive FB. If the MC_ MoveAdditive FB is activated in Continuous Mode, the specified relative distance is added to the actual position (at the time of execution).
MC_MoveSuperimposed	commands a controlled motion of a specified relative distance additional to an existing motion. The existing Motion is not interrupted, but is superimposed by the additional motion.
MC_MoveVelocity	commands a never-ending controlled motion (jog) at a specified velocity.
MC_Stop	commands a controlled motion stop and transfers the axis to the "Stopping" state. It aborts any ongoing function block execution. When the Done output is set, the state transfers to StandStill. While the axis is in Stopping state, no other FB can perform any motion on the same axis.
MC_Power	controls the power stage: enable(on) and disable (off).
MC_ReadStatus	returns Axis status details with respect to the motion currently in progress.
MC_ReadAxisErr	indicates Drive-related errors.
MC_ResetError	makes the transition from the state ErrorStop to StandStill by resetting all internal axis-related errors and clearing pending commands – it does not affect the output of the FB instances.
MC_ReadParameter & MC_ReadBoolParameter	return the value of a Drive parameter. The returned value has to be converted to Real if required. If not possible, the vendor has to provide a supplier-dependent FB for it.
MC_WriteParameter & MC_WriteBoolParameter	modify the value of a Drive parameter.
MC_ReadActualPosition	returns the value of the actual position.
MC_MachRegist	performs Mark-to-Machine registration
MC_MarkRegist	performs Mark-to-Mark registration
MC_StopRegist	turns off registration
MC_CamTblSelect	selects the Cam tables by setting the pointers to the relevant tables.
MC_CamIn	engages the Cam.
MC_CamOut	disengages the slave from the master axis immediately in a cam block.
MC_GearIn	commands a ratio between the VELOCITY of the slave and master axis.
MC_GearOut	disengages the slave from the master axis.
MC_AddSuperAxis	adds a superimposed axis to a specified axis.
MC_RemoveSuperAxis	removes a superimposed axis from an axis.

8.2.4.2 PLCopen Function Blocks - Overview

8.2.4.2.1 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, described below, define the transition from the active move to the next move.

8.2.4.2.2 Buffer Modes

Some of the FBs have an input called BufferMode. With this input, the FB can either work in a Nonbuffered mode (default behavior) or in a Buffered mode. The difference between those modes is when they start their action:

- A command in a non-buffered mode acts immediately, even if this interrupts another motion
- A command in a buffered mode waits untill the current FB sets its **Done** output (or **InPosition**, or **InVelocity**,..).

There are six buffer modes that can be specified at the BufferMode input of the function blocks.	

Buffer mode	Value	Short name	Description
MC_BUFFER_ MODE_ ABORT	0	Abort	A move that specifies Abort aborts the active move, removes the next move from the queue, and immediately becomes the active move and begins executing
MC_BUFFER_ MODE_	1	Buffer	One of three events can happen with a move that specifies Buffer:
BUFFERED			 Case 1. If there is no active move, this move immediately becomes the active move and begin executing. Case 2. If there is an active move but no next move queued, this move is queued as the next move, and 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.
MC_BUFFER_ MODE_ BLENDING_ PREVIOUS	2	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
MC_BUFFER_ MODE_ BLENDING_ NEXT	3	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 that it reaches this move's programmed velocity at the time the active move reaches its endpoint
MC_BUFFER_ MODE_ BLENDING_ LOW	4	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
MC_BUFFER_ MODE_ BLENDING_ HIGH	5	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

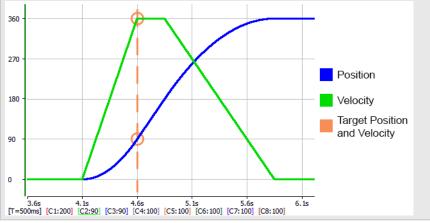
The MC_BUFFER_MODE_BLENDING_NEXT and MC_BUFFER_MODE_BLENDING_HIGH buffer

modes may increase the acceleration to meet the target position at the specified velocity. The acceleration will be automatically re-calculated to reach the target position at the specified velocity if the specified acceleration is too small to reach the target velocity within the specified distance.

For example, suppose the following 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 will automatically calculate and apply an acceleration of 720 to reach the target position at the specified velocity:



8.2.4.2.3 S-curve and Trapezoidal Acceleration/Deceleration

8.2.4.2.3.1 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/deceleration reached during changes in velocity. The Deceleration input is unused. The Jerk input specifies the constant rate of change of acceleration and deceleration used to cause a smooth transition to and from maximum acceleration.

The Small Jerk Acceleration below 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/deceleration when transitioning from one velocity to another.

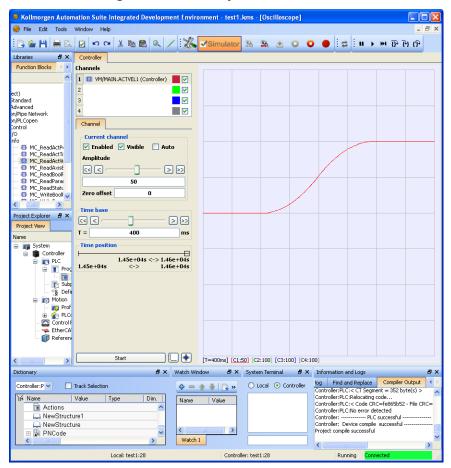


Figure 6-11: Small Jerk Acceleration

The Large Jerk Acceleration below 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/deceleration when transitioning from one velocity to another.

KAS IDE User Guide | 8 Understanding KAS

🗧 Kollmorgen Auton	nation Suite Integrated Developme	ent Environment - test	1.kms - [Oscilloscope]		
🌼 File Edit Tools	Window Help				_ 8 ×
: 🕞 🍲 💾 🗮 🖳	🖉 🔊 🛪 🐰 📾 🛃 🗸	Simulato	r 🏦 🏝 🛓 🔘	• ● ! # ! u	▶ ₩ ᠿ ᠿ ᠿ
Libraries 🗗 🗙	Controller				
Function Blocks	Channels				
<u>^</u>	1 💷 VM/MAIN.ACTVEL1 (Controller)				
ect)	2				
Standard	3				
Advanced on/Pipe Network	4				
n/PLCopen 🔤	Channel				
Control JO	-Current channel				
nfo	Enabled Visible Aut				
💷 MC_ReadActPi 💷 MC_ReadActT(.0			
MC_ReadActW MC_ReadAxisE	Amplitude				
- 🕀 MC_ReadBoolF					
🕄 MC_ReadPara 🕄 MC_ReadStati	50				
	Zero offset 0			/	
< >	∼ Time base				
Project Explorer 🛛 🗗 🗙		200			
Project View					
Name	T = 400	ms			
🖃 📷 System 📄 💼 Controller	Time position				
	1.44e+04s <-> 1.44				
🖨 🛐 Prog	1.44e+04s <-> 1.44	fe+04s			
💽 Subp					
🔤 🛐 Defi					
🖃 📷 Motion					
🖃 📷 Prof 🕀 🏠 PLCo					
- 🔄 Control F					
< >	Start	[T=400ms] [C	<u>1:50]</u> [C2:100] [C3:100] [C	24:100]	
Dictionary	8 ×	Watch Window 🗗 >	System Terminal 🗗	× Information and Log:	s 🗗 🗙
Controller:P 😽 📃	Track Selection	\$ = ↑ ↓], »	🔿 Local 💿 Controlle	er log Find and Repl	
A Name	Value Type Dim.			Controller:PLC: < CT : Controller:PLC:Reloca	Segment = 352 byte(s) >
Actions	value Type Dim.	Name Value		Controller:PLC: < Cod	le CRC=fe865b52 - File CRC=
A NewStruct	ure1			Controller:PLC:No err Controller:	or detected - PLC successful
RewStruct		<		Controller: Device co	mpile_successful
🕀 🔂 PNCode	×	Watch 1		Project compile succe	ssiui 🗸
				<	<u>></u>
	Local: test1:28	Contr	oller: test1:28	Running	Connected

Figure 6-12: Large Jerk Acceleration

8.2.4.2.3.2 Trapezoidal

If the Jerk input of a motion function block is zero, trapezoidal acceleration/deceleration is used. The Acceleration input specifies the linear acceleration rate. The Deceleration input specifies the linear deceleration rate.

The Trapezoidal Acceleration below is a velocity plot of the acceleration of a move when trapezoidal acceleration is used (Jerk = 0).

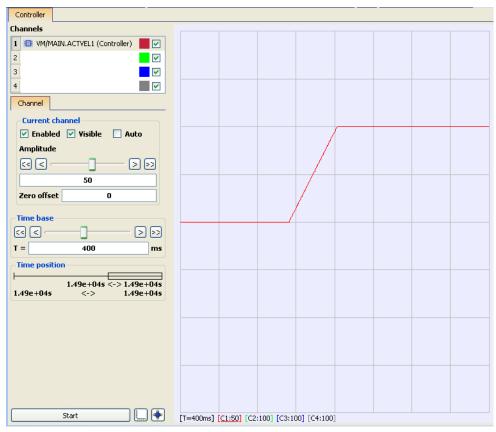


Figure 6-13: Trapezoidal Acceleration

8.2.4.2.4 Selection of Acceleration and Jerk Parameters for Function Blocks

Definition

Acceleration is the first derivative of velocity, or the rate of change of velocity. The Acceleration rate therefore specifies how quickly an axis may change its velocity.

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 therefore controls how abrupt the axis begins and ends the acceleration and deceleration curves.

See also Motion Profile.

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 will attain its programmed velocity. The following are generalizations that can be made about acceleration, jerk and their relationships to each other.

- The higher the acceleration rate, the faster the axis will obtain programmed velocity
- The higher the jerk rate, the more responsive the axis will be to changes in command
- Excessive jerk typically, more noticeably contributes to harsh acceleration 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, while higher jerks sharpen the beginning and end of acceleration
- Typically, Jerk > Acceleration, Acceleration > Velocity

Methods

There are several methodologies to determine proper acceleration and jerk values. These methodologies allow you to calculate parameters given different desired profiles. Once parameters are calculated, you can then modify them as desired to obtain the results you want. Acceleration and Jerk values are subject to the limits of ratios as explained below.

1/3,1/3,1/3 time, given velocity and time. 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 time, accelerate 1/3 of the time and ramp acceleration down 1/3 of the time. Time is the desired amount of time to reach desired velocity. Note, this is the time to change velocity, not the time to complete the move.

```
Acceleration = (3 * Velocity)/(2 * time)
Jerk = 3* Acceleration / time
```

1/3,1/3,1/3 velocity, given velocity and time. 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 and 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. Note, this is the time to change velocity, not the time to complete the move.

```
Acceleration = (5 * Velocity) / (3 * time)
Jerk = (3 * Acceleration ^2) / (2 * velocity)
```

Calculate Jerk, given Velocity, acceleration and time. If you already know the maximum acceleration of the axis, and want to simply calculate a Jerk given the velocity and time, you can use the following equation. Note, this is the time to change velocity, not the time to complete the move.

```
Jerk = (2 * Acceleration) / ( time - ( velocity / (2 * acceleration)))
```

8.2.4.2.4.1 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 would be 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.

8.2.4.2.5 Profile Generator

Each servo axis has three Profile Generators which has its own queue. The three Profile Generators are: Normal, Superimposed, and Phasing.

- Normal handles all single-axis and master-slave moves
- Superimposed handles MC_MoveSuperimp moves exclusively
- Phasing handles MC_Phasing phase shifts exclusively

The three Profile Generators allow these types of moves to execute simultaneously.

8.2.4.2.6 AXIS_REF Structure

The PLCopen specification indicates a data structure to be used for identifying the axis at a function block input. AXIS_REF contains two members:

Member	Туре	Description
AXIS_NUM	INT	The axis number
AXIS_ENGINE	UINT	Reserved. Do not modify this variable or rely on its value.

For more details on Axis Number, About Axis Name and Number

You have to create and initialize this data structure in your application.

8.2.4.2.7 Axis Parameters

The table below is a list of Boolean parameters currently supported. These parameters are read and written by the function blocks MC_ReadBoolPar and MC_WriteBoolPar.

Parameter	ID	Name	R/W	Update Rate Type	Description
MC_AXIS_PARAM_IN_ POSITION	1011	Axis In-Position	Read Only	Controller	True if the axis has no active or next move queued, the command delta is 0, and the actual position is within the in- position bandwidth of the command position. False otherwise, Boolean.
MC_AXIS_PARAM_ DRIVE_WARNING	1013	Drive Warning	Read Only	EtherCAT Cyclic	Drive Warning Status. ¹
MC_AXIS_PARAM_ REGIST_GOOD	1025	Good Registration Mark Occurred	Read Only	EtherCAT Non-Cyclic	True indicates that a good registration mark was encountered. This Boolean will be automatically reset after it has been read. ²
MC_AXIS_PARAM_ REGIST_BAD	1026	Bad Registration Mark Occurred	Read Only	EtherCAT Non-Cyclic	True indicates that a bad registration mark was encountered. This Boolean will be automatically reset after it has been read. 2
MC_AXIS_PARAM_FI_ OCCURRED	1027	Fast Input Occurred	Read Only	EtherCAT Non-Cyclic	This parameter is deprecated. The recommended replacement is to use 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 will automatically reset after it has been read. ³

Parameter	ID	Name	R/W	Update Rate Type	Description
MC_AXIS_PARAM_ APPLY_ SUPERIMPOSED_ DISTANCE	1033	Apply Superimposed Distance	Read / Write	Controller	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 <i>Position</i> input to MC_MoveAbsolute. <i>Example of False</i>
					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_MoveSuperimp calls at the time of the initial MC_ MoveAbsolute call. The final reported position of the axis will be equal to the <i>Position</i> input to MC_MoveAbsolute. If MC_MoveSuperimp is called during MC_MoveAbsolute, then the final position will include the superimposed distance.
					Example of True Param 1033 = True MC_MoveSuperimp move 1 +50 MC_MoveSuperimp move 2 -35 MC_MoveAbsolute 125 End Position

Parameter	ID	Name	R/W	Update Rate Type	Description
					= 125
					The default value for this parameter is False.

The table below is a list of non-Boolean parameters currently supported. These parameters are read and written by the function blocks MC_ReadParam and MC_WriteParam.

Parameter	ID	Name	R/W	Update Rate Type	Description
MC_AXIS_PARAM_ CMD_POS	1	Command Position	Read Only	Controller	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.
MC_AXIS_PARAM_ ACT_VEL	10	Actual Velocity	Read Only	EtherCAT Cyclic	Axis actual velocity, User unit/sec
MC_AXIS_PARAM_ CMD_VEL	11	Command Velocity	Read Only	Controller	Axis command velocity – includes any command deltas from superimposed axes, User unit/sec
MC_AXIS_PARAM_ PHASE_SHIFT	1000	Phase Shift	Read Only	Controller	The amount of phase shift applied by MC_ Phasing, in slave axis' user units
MC_AXIS_PARAM_ SUPERIMPOSED_ DISTANCE	1001	Superimposed Distance	Read Only	Controller	The cumulative distance traveled via MC_ MoveSuperimp moves, user units
MC_AXIS_PARAM_ MASTER_OFFSET	1002	Master Offset	Read / Write	Controller	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.
MC_AXIS_PARAM_ SLAVE_OFFSET	1003	Slave Offset	Read / Write	Controller	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.

KAS IDE User Guide | 8 Understanding KAS

Parameter	ID	Name	R/W	Update Rate Type	Description
MC_AXIS_PARAM_ MOVE_TYPE_ACTIVE	1004	Active Move Type	Read Only	Controller	The active move type (see table in Move Types)
MC_AXIS_PARAM_ MOVE_TYPE_NEXT	1005	Next Move Type	Read Only	Controller	The queued (next) move type (see table in Move Types)
MC_AXIS_PARAM_ POSITION_ERROR	1006	Position Error	Read Only	EtherCAT Cyclic	Position error in user units
MC_AXIS_PARAM_ FEEDBACK_LAST	1007	Raw Feedback	Read Only	EtherCAT Cyclic	Raw Feedback position in user units. Keeps track of the actual position based on the physical feedback device connected to the drive and will not be offset as a result of calling MC_ SetPos.
MC_AXIS_PARAM_ ROLLOVER_POSITION	1008	Rollover	Read / Write	Controller	The axis rollover position 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
MC_AXIS_PARAM_ VELCOMP_FACTOR	1009	Velocity Compensation Factor	Read / Write	Controller	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 for more information. ¹
MC_AXIS_PARAM_ VELCOMP_FILTER	1010	Velocity Compensation Filter	Read / Write	Controller	The number of updates in which to apply a change in velocity compensation. See Velocity Compensation for more information. ¹ 4
MC_AXIS_PARAM_IN_ POSITION_BAND	1012	Axis In-Position Bandwidth	Read / Write	Controller	The bandwidth about the command position to determine the state of the in-position flag. ²
MC_AXIS_PARAM_ DRIVE_STATUS	1014	Drive Status	Read Only	EtherCAT Cyclic	Drive Status Word (Similar to MLAxisStatus) Some status bits are set only at program startup.

Parameter	ID	Name	R/W	Update Rate Type	Description
MC_AXIS_PARAM_UU_ FB_RATIO_NUM	1015	User Unit to feedback unit ratio numerator	Read Only	Static	The numerator of the user-units-to-feedback- units ratio (UU/FBU Ratio). See About the User Units to Feedback Units Ratio below. 1
MC_AXIS_PARAM_ TORQUE_ACTUAL	1016	Actual Torque	Read Only	EtherCAT Cyclic	The actual torque being delivered by the drive, expressed in thousandths of max torque. See Example of reading the current to units of Amps.
MC_AXIS_PARAM_ BUS_ADDRESS	1017	Drive Address	Read Only	Static	Drive address value to be used in EtherCAT fieldbus functions as drive address. Before using in fieldbus functions, this value needs to be converted to integer by using a convert any to DINT function. ¹
MC_AXIS_PARAM_ SENSOR_DELAY	1018	Sensor Delay	Read / Write	Controller	Compensation for Physical sensor delay for MC_TouchProbe (FunctionBlock), _{in} microseconds. ¹
MC_AXIS_PARAM_ INTERP_CMD_POS	1019	Interpolated Command Position	Read Only	Controller	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.
MC_AXIS_PARAM_ INTERP_CMD_VEL	1020	Interpolated Command Velocity	Read Only	Controller	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.

Parameter	ID	Name	R/W	Update Rate Type	Description
MC_AXIS_PARAM_ REGIST_COMP	1021	Registration Compensation	Read Only	Controller	The latest calculated registration compensation value. This value is updated each time a good registration mark is encountered. This value is in User Units.
MC_AXIS_PARAM_ REGIST_DIST	1022	Distance Between the Last Two Good Registration Marks	Read Only	EtherCAT Non-Cyclic	Distance between the last two good registration marks. This value is in User Units. ²
MC_AXIS_PARAM_ REGIST_GOOD_CNT	1023	Number of Consecutive Good Registration Marks	Read / Write	EtherCAT Non-Cyclic	Number of consecutive good registration marks. This value is incremented each time a good registration mark is encountered and 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. ²
MC_AXIS_PARAM_ REGIST_BAD_CNT	1024	Number of Consecutive Bad Registration Marks	Read / Write	EtherCAT Non-Cyclic	Number of consecutive bad registration marks. This value is incremented each time a bad registration mark is encountered and 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. ²
MC_AXIS_PARAM_UU_ FB_RATIO_DEN	1028	User Unit to feedback unit ratio denominator	Read Only	Static	The denominator of the user-units-to-feedback- units ratio (UU/FBU Ratio). See About the User Units to Feedback Units Ratio below.
MC_AXIS_PARAM_CM_ ACT_CMD_POS	1029	Coordinated Motion Applied Command Position	Read Only	Controller	Amount of motion actually applied to the PLCopen axis drive by Coordinated Motion commands.

410

Parameter	ID	Name	R/W	Update Rate Type	Description
MC_AXIS_PARAM_CM_ CMD_POS	1030	Coordinated Motion Command Position	Read Only	Controller	Amount of motion requested of a PLCopen axis by the Coordinated Motion commands.
MC_AXIS_PARAM_ INGEAR_BANDWIDTH	1031	"In Gear" bandwidth	Read/Write	Controller	The bandwidth about the target slave velocity in which the slave axis will lock onto the master axis and the "InGear" output will turn on for the MC_ GearIn function block; User unit/sec (Default value 0.1 User units/sec).
MC_AXIS_PARAM_ DRIVE_AXIS_NUMBER	1032	Drive Axis Number	Read Only	Static	One-based number that specifies the axis on the drive
MC_AXIS_PARAM_UU_ FB_RATIO	1034	User unit to feedback unit ratio	Read Only	Static	The user units to feedback units ratio for the axis. See About the User Units to Feedback Units Ratio below.

- 1. This is a configuration parameter.
- 2. There is some delay is acquiring fast input information from drives as well as 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. Please see Tuning Controller Performance for EtherCAT Communication Latency for more details.
- 3. There is some delay is acquiring fast input information from drives as well as 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. Please see Tuning Controller Performance for EtherCAT Communication Latency for more details.
- 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. Please see Tuning Controller Performance for EtherCAT Communication Latency for more details.

Update Rate Type	Description
EtherCAT Cyclic	Update rate depends on the EtherCAT (link to ECAT update page EtherCAT Master Settings Tab) and KAS application program (link to Define the PLC Cycle) update rates
EtherCAT Non- Cyclic	Update rate depends on the update rate of reading the parameter through Ethercat (Link to times to read non-cyclic parameters [? does the time come from ECATReadSDO FB or from the MCReadParam DriveReadParam FB EtherCAT Library]) and KAS application program update rate
Controller	Update rate depends on KAS application program update rate (link to Define the PLC Cycle)
Static	Values do not change after axis is created.

About the User Units to Feedback Units Ratio

Parameters 1015 and 1028 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, which is used to command the drive per the ratio.

For a drive where the drive interface units are set to 1048576 units per revolution, the value of the *UU-per-revolution* ratio can be calculated as follows:

```
UU per revolution = MC_AXIS_PARAM_UU_FB_RATIO_NUM * 1048576 / MC_AXIS_
PARAM UU FB RATIO DEN
```

Example

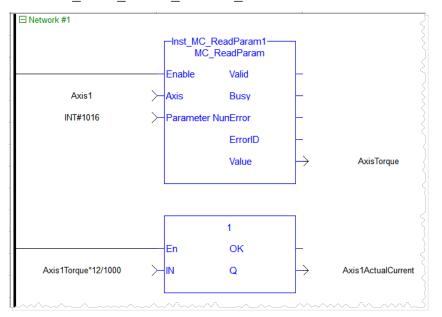
For a drive:

- A ratio of 360 UU / 1048576 FBU will generate 360 UU per revolution of the drive motor and a 0.000343323 UU-to-FBU ratio.
- A ratio of 11379 UU / 1898996404 FBU will generate 6.283185 UU per revolution of the drive motor and a 5.99211e-6 UU-to-FBU ratio.

Example of reading the current to units of Amps.

After reading the current via MC_ReadParam, the following equation converts current to amps for a 12 amp peak drive:

MC_AXIS_PARAM_TORQUE_ACTUAL * Drive Peak Current Rating / 1000



8.2.4.2.8 Axes Group Parameters

The table below is a list of Boolean parameters currently supported. These parameters are read and written by the function blocks MC_GrpReadBoolPar and MC_GrpWriteBoolPar.

Parameter	ID	Name	R/W	Description
MC_GRP_ PARAM_ IGNORE_ AXIS_ ESTOP	1000	Ignore Axis EStop	Read/Write	Controls whether the axes group will continue performing motion if one of the member axes is experiencing an Estop. A value of FALSE indicates that the axes group should enter an Estop state if a member axis enters the Estop state. A value of TRUE indicates that the axes group should continue performing motion.
MC_GRP_ PARAM_ AXIS_ ESTOP_ ACTIVE	1001	Axis EStop Active	Read Only	This Read-only parameter will be asserted 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 will be set to FALSE.

The table below is a list of non-Boolean parameters currently supported. These parameters are read and written by the function blocks MC_GrpReadParam and MC_GrpWriteParam.

Parameter	ID	Name	R/W	Description
MC_GRP_ PARAM_ CIRCLE_ TOLERANCE		Command Velocity	Read/Write	This parameter controls how large an error will be acceptable when constructing circles using circle mode MC_CIRC_MODE_CENTER. A value of zero will default to the 1 part in 100,000 tolerance. For more information see Precision Requirements for Circular Move Input Parameters.

8.2.4.2.9 Axis Positions Data

CommandPosition is the command position (without modulo and in the axis' User Units) that is sent to the drive interface to command an axis. This position is tied to the Status output of the MC_Power function block:

- When the Status = 1 the command position is a combination of the Normal, Superimposed and Phasing commands
- When the Status = 0 the command position tracks the Actual Position

CoordinatedMotionCmdPos is the command position generated by the Coordinated Motion Interpolator when interpolating coordinated motion moves.

NormalCmdPos is the command position generated by the Normal Interpolator when interpolating a single axis move or a slave move.

PhaseCmd Pos is the command position generated by the Phasing Interpolator when interpolating a MC_Phasing master phase shift. It is also incorporated in the generation of the NormalCmdPos when interpolating a slave move.

SuperimposedCmdPos is the command position generated by the Superimposed Interpolator when interpolating a MC_Superimp move

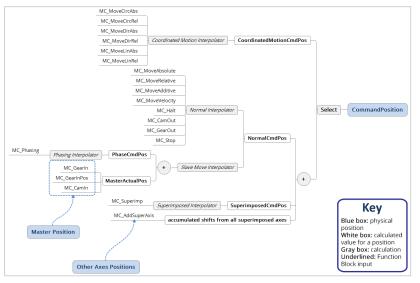


Figure 6-14: Graphic of how PLCopen axis position data is calculated.

Actual Position is the position of the axis read from the drive interface, which is read from the feedback device,

Group Actual Position is the set of positions of an axes group read from the axis interface,

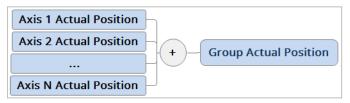
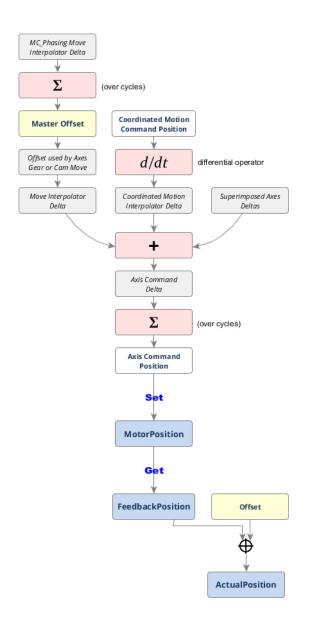


Figure 6-15: How Actual Position is calculated.



8.2.4.2.10 Axis Velocity Data

The following velocity data are related to PLCopen Axis

- ActualVelocity: 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)
- **Command Velocity**: the rate of change in command position. This velocity is tied ot the Status output of the MC_Power function block. Since command velocity is the rate change of command position:
 - When the **Status = 1** the command velocity is the rate of change of the command position sent to the drive interface to command the axis.
 - When the Status = 0 the command velocity tracks the Actual Velocity

8.2.4.2.11 Velocity Compensation

Velocity compensation is a value that is added to the master actual position in order to predict where the master will be when the new slave command position is sent to the slave. In other words, 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.



The above diagrams represent a 1:-1 gearing.

- On the left, we have 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.
- On the right, we have 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 will cause 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** and **Velocity Compensation Filter**, described in more detail below. Both parameters are properties of master axes. 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.

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 will be applied. The larger the value of *VelocityCompensationFactor*, the more prediction takes place and the more prediction error there will be. For physical axes, the value for VelocityCompensationFactor will normally be 1 or 2. Some feedback devices or drives may require larger values. For virtual masters, the value will usually be 0. Virtual masters do not need compensation since their position is always known by the controller.

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 *VelocityCompensationFilter* is the number of motion cycles over which a previously calculated compensation is applied. This helps smooth noisy systems. The default value for VelocityCompensationFilter is 1, which means that 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 can be 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.

8.2.4.2.12 Move Types

MoveID	Description	Related FB
0	No move	
1	Distance move	, and
2	Position move	
3	Velocity move	
4	Halt move	
5	Gear-in move	
6	Gear-out move	
7	Reference move	
8	Stop move	
9	Gear-in pos. move	
10	Cam profile move	
11	Cam-out move	

8.2.4.2.13 Rollover

The Rollover Position is specified in user units in the PLCopen Axis Data dialog. When this value is non-zero, the axis' position is reset to zero when it reaches the rollover position.

For example, if the rollover position is 360 and the axis is traveling in the positive direction, the axis position counts up until it reaches 360 where it resets to 0 and then continues counting up from there.

If the axis is traveling in the negative direction, the axis position counts down until it reaches 0, where it resets to 360 and then continues counting down from there.

Refer to MC_MoveAbsolute's's description for an explanation of its operation when Rollover Position is nonzero.

When the Rollover Position is zero, rollover is not in effect and the axis position continues to count up when traveling in the positive direction and count down when traveling in the negative direction.

8.2.4.2.14 PLCopen Function Block ErrorID Output

These are the possible errors that could be returned at the ErrorID output of the function blocks.

ErrorID	Description
0	no error
1	queue full
2	abort mode required
3	invalid axis
4	One of four 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 an axis error exists, the axis is in the stopping state or 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 For example, if a MC_ReadParm of FollowingError (1006) is programmed on a simulator axis for which 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 for more details.
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
30	Name is already used
31	The number of axes is too small
32	The number of axes is too large

ErrorID	Description
33	Invalid update rate
34	Invalid axis group
35	Invalid kinematic parameter (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	Colinear 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. As an 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.
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

420

ErrorID	Description
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 two scenarios:
	 User application code was written in a way that the function block was called more than once in the same cycle PLC program is being 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.
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 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 20000 data points
108	Profile is currently is use
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 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.

8.2.4.3 PLCopen Function Blocks - General Rules

The general rules for PLCopen are:

- Input parameters
- Missing input parameters
- Output Exclusivity
- Output Status
- Sign Rules
- Error Handling Behavior
- Behavior of Done Output
- Behavior of CommandAborted Output
- Behavior of Busy Output
- Inputs Exceed Application Limits
- Output 'Active'

8.2.4.3.1 Input parameters

Unless specified otherwise in the function block's description, the input parameters are read with the rising edge of the Execute input.

The input parameters can be as follows:

• Function Blocks with **Execute**

These FBs will initiate execution on the rising edge of 'Execute'. Execution will continue 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 within the same cycle will return an error code 76.

• Function Blocks with Enable

These FBs will continuously be executed every PLC cycle, as long as the Enable remains high.

• Function with **En**

This is very similar to ENABLE on Function blocks. But, as already explained in Difference between Functions and Function Blocks, functions are expected to complete in one cycle.

8.2.4.3.2 Missing input parameters

If any input parameter of a function block is missing (**open**), the compiler generates an error.

8.2.4.3.3 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.

NOTE

The output functionality 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. Unlike most other motion function blocks, once done, the MC_GearOut will remain 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 may be true at the same time.

8.2.4.3.4 Output Status

The **Done**, **Error**, **ErrorID** and **CommandAborted** outputs are reset with the next rising edge of **Execute**.

If an instance of 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.

8.2.4.3.5 Sign Rules

Velocity, **Acceleration**, **Deceleration** and **Jerk** are always positive values. **Position** and **Distance** can be positive or negative.

8.2.4.3.6 Error Handling Behavior

Two outputs deal with errors that can occur while executing a function block. These outputs are defined as follows:

- **Error**: the rising edge of **Error** informs you that an error occurred during the execution of the function block
- **ErrorID**: Error number.

Done, **InVelocity**, **InGear**, and **InSync** mean successful completion so these signals are logically exclusive to **Error**.

Instance errors do not always result in an axis error. Some bring the axis to StandStill(.

8.2.4.3.7 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, the following 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, there are three possible outcomes:

- 1. It completes successfully. At that time, the **Done** output goes high.
- 2. It is aborted prior to completing by a subsequent motion command. At that time, the **CommandAborted** output goes high.
- 3. It encounters an error prior to completing or an invalid input is specified. At that time, the **Error** output goes high.

These outputs stays 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.

8.2.4.3.8 Behavior of CommandAborted Output

CommandAborted is set when a commanded motion is interrupted by another motion command. The reset-behavior of **CommandAborted** is like that of **Done**. When **CommandAborted** occurs, the other output signals such as **InVelocity** are reset.

8.2.4.3.9 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 **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.

8.2.4.3.10 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.

8.2.4.3.11 Output 'Active'

The **Active** output is set at the moment the function block takes control of the motion of the respective axis.

8.2.4.3.11.1 Coordinated Motion

Use the following table to determine the state of the Busy and Active outputs when transitioning from one move to the next.

Function Block Output	First Move	Second Move
Busy	False	True
Active	False	False
Done	True ¹	False

8.2.4.3.12 List of Input Parameters

The input parameters are listed as follows:

- Function Blocks with **Execute**
- Function Blocks with **Enable**
- Function with **En**

8.2.4.3.12.1 List of PLCopen function blocks with Execute

These FBs will initiate execution on the rising edge of 'Execute'. Execution will continue 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 within the same cycle will return an error code 76.

Function Block	Description
MC_MoveAbsolute MC_MoveRelative MC_MoveAdditive MC_MoveSuperimp MC_MoveVelocity MC_Halt MC_CamIn MC_CamOut MC_GearIn MC_GearOut MC_GearInPos	A positive transition of this input requests to queue the move
MC_Phasing	A positive transition of this input requests to queue the phase shift move
MC_SyncSlaves MC_TouchProbe MC_AbortTrigger MC_SetPosition	A positive transition of this input causes this function block to execute
MC_WriteBoolPar MC_WriteParam	A positive transition of this input writes the specified parameter

¹This output will be false if the move is interrupted by a MC_GrpHalt, MC_GrpStop, or if the second move's buffer mode is "Aborting".

Function Block	Description
MC_Reference	A positive transition of this input requests to queue the reference move and arm the reference trigger event(s)
MC_CamTblSelect	A positive transition of this input reads and initializes the specified profile
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

8.2.4.3.12.2 List of PLCopen function blocks with Enable

These FBs will continuously be executed every PLC cycle, as long as the Enable remains high.

Function Block	Description
MC_ReadBoolPar MC_ReadParam	When this input is high, the specified parameter is read
MC_SetOverride	When this input is high, the override factors is written
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_ReadStatus	When this input is high, the function block outputs is updated
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

8.2.4.3.12.3 List of PLCopen functions with input parameter En

This is very similar to ENABLE on Function blocks. But, as already explained in Difference between Functions and Function Blocks, functions are expected to complete in one cycle.

Function	Description
MC_CreatePLCAxisMC_ CreatePLCAxis	When this input is high, a PLCopen axis is created
MC_InitAxis	When this input is high, the specified axis is initialized
MC_EStop	When this input is high, an E-stop is generated for the specified axis
MC_ResetError	When this input is high, the specified axis's errors is reset
MC_AddSuperAxis	When this input is high, the specified axis is added to the superimposed axis list of the receiving axis.
MC_RemSuperAxis	When this input is high, the specified axis is removed from the superimposed axis list for the specified receiving axis.

8.2.4.4 PLCopen State Machine

The following diagram normatively defines according to PLCopen the behavior of the axis at a highlevel 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 (see diagram below). A change of state is reflected immediately when issuing the corresponding motion command (please note that the response time of 'immediately' is system dependent).

There are seven states defined:

- 1. Stand Still
- 2. Homing
- 3. Discrete Motion
- 4. Continuous Motion
- 5. Synchronized Motion
- 6. Stopping
- 7. Error Stop

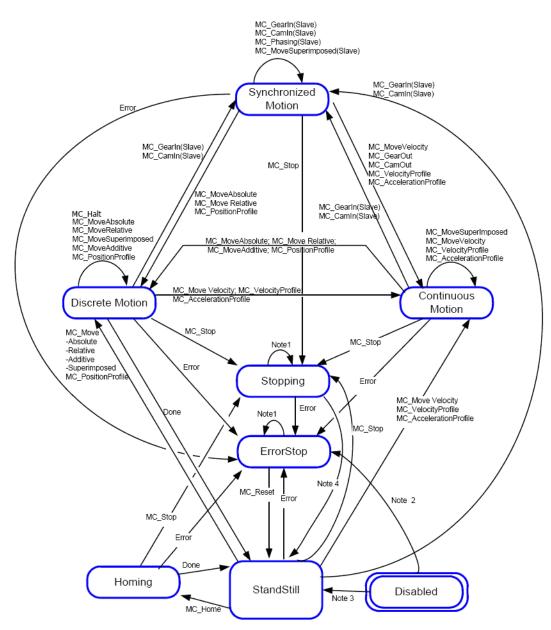


Figure 6-16: Motion State Machine (PLCopen)

NOTE

Note 1: In this state **ErrorStop** or **Stopping**, all function blocks can be called, although they are not executed, except MC_Reset and Error which generate the transition to **StandStill** or **ErrorStop** respectively

Note 2:MC_Power FB is called with Enable=TRUE and there is an error in the Axis **Note 3**: MC_Power FB is called with Enable=TRUE and there is **no** error in the Axis **Note 4**: MC_Stop.Done **and not** MC_Stop.Execute

A normal procedure would start in **StandStill**. In this state the power can be switched on per axis (via the command MC_Power). Also, you can access the **Homing** state (via the issue of the command Home per axis), which after normal completion returns to **StandStill**. From here you can 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) Reset command, you can return to **StandStill**, from which the machine can be moved to an operational state again.

Please note that the States define the functionality of the function blocks. Function Blocks which are not listed in the State Diagram do not affect the state of the axis, meaning that, whenever they are called, the state does not change. They are:

- MC_ReadStatus
- MC_ReadAxisErr
- MC_ReadParameter
- MC_ReadBoolParameter
- MC_WriteParameter
- MC_WriteBoolParameter
- MC_ReadActualPosition
- MC_CamTableSelect

State **Disabled**

The **Disabled** state describes the initial state of the axis. In this state, the movement of the axis is not influenced by the FBs. The axis feedback is operational.

If the MC_PowerFB is called with Enable=TRUE while being in **Disabled**, this either leads to **Standstill** if there is no error inside the axis, or to **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 function block controls an axis while the MC_Power FB with Enable=FALSE is called, the motion generating function block is aborted (CommandAborted).

Disable means power off without error.

State ErrorStop

The intention of the **ErrorStop** state is that 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. These axis' errors can also be 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**.

From StandStill to Stopping

Calling the FB MC_Stop in state **StandStill** changes the state to **Stopping** and back to **Standstill** when "Execute = FALSE". The state **Stopping** is kept as long as the input "Execute" is true. The "Done" output is set when the stop ramp is finished.

StandStill is power on without an error.

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. For instance, 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.

8.2.4.5 Programming a Dual Core Controller

A dual-core controller operates slightly differently in order to take advantage of the extra performance provided by a multiple-core CPU. Consequently, there are some differences in programming a dual-core controller from programming a single-core controller. This page lists those differences.

8.2.4.5.1 Cached Data

428

A dual-core controller gains most of its performance by running PLC code and motion code at the same time. In order to avoid race conditions, function blocks that read motion data will report the data that the motion code used or calculated during the last cycle.

List of function blocks that return cached data:

- MC_AddSuperAxis
- MC_ClearFaults
- MC_EStop
- MC_GrpReadBoolPar
- MC_GrpReadParam
- MC_GrpReadStatus
- MC_ReadActPos
- MC_ReadActVel
- MC_GrpReadCmdPos
- MC_GrpReadCmdVel
- MC_GrpReadActAcc
- MC_GrpReadActPos
- MC_GrpReadActVel
- MC_GrpReadError
- MC_GrpWriteBoolPar
- MC_GrpWriteParam
- MC_MoveContVel
- MC_Power
- MC_ReadAxisErr
- MC_ReadBoolPar
- MC_ReadParam
- MC_ReadStatus
- MC_RemSuperAxis
- MC_ResetError
- MC_SetPos
- MC_StopRegist
- MC_WriteParam
- MLAxisAddTq
- MLAxisCmdPos
- MLAxisFBackPos
- MLAxisGenEN
- MLAxisGenIsRdy
- MLAxisGenPos
- MLAxisPipePos
- MLAxisReadActPos
- MLAxisReadGenStatus
- MLAxisReadTq
- MLAxisReadVel
- MLAxisReAlgnRdy
- MLAxisStatus
- MLAxisStop
- MLAxisTimeStamp
- MLBlkIsReady
- MLBlkReadOutVal
- MLCompCheck
- MLIntWriteOutVal
- MLMotionStatus
- MLPhaReadActPhase
- MLPmpStatus
- MLSyncWriteDeltaS
- MLTrigClearFlag

- MLTrigIsTrigged
- MLTrigReadPos
- MLTrigReadTime

8.2.4.5.2 Calling Function Blocks Multiple Times in the Same Cycle

A given function block can usually be called multiple times without causing problems when using a single-core controller. For example, it is possible to call MC_MoveVelocity multiple times to start velocity moves on multiple axes. You would need to set the **Execute** input to **False** between every call to start a new velocity move. In doing so, you lose the ability to monitor the status of the velocity moves because you changed the axis parameter for each call, but if you were not concerned with monitoring the status of the velocity moves, you could 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 dualcore controller. To do so 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 (error 76), the **Error** and **ErrorID** outputs will report no error on the first cycle. The Error and ErrorID outputs will report their status on 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 (error 76) will be returned immediately.

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_MoveCircular
- MC_MoveContVel
- MC_MoveDirect
- MC_MoveLinear
- MC_MoveRelative
- MC_MoveSuperimp
- MC_MoveVelocity
- MC_Phasing

430

MC_Reference

- MC_RemAxisFromGrp
- MC_SetPos

- MC_Stop
 MC_SyncSlaves
 MC_TouchProbe
 MC_UngroupAllAxes

8.3 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 can be 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.

Slave device input and output data definitions:

- Outputs are written by the master and read by the slave device
- Inputs written by the slave device and read by the master

EtherCATProcessImage

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 (SDO).

References

- EtherCAT Specification V1.0 refer to http://www.ethercat.org (in Member Area Downloads)
- Büttner, H.; Janssen, D.; Rostan, M. (2003), EtherCAT the Ethernet fieldbus, (PDF), PC Control Magazine 3: 14-19



8.3.1 Functional Principle

Typical 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 therefore takes a different approach, called "processing on the fly" (for more details, refer to EtherCAT Implementation).

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. Similarly, 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.

8.3.2 EtherCAT Features

Summary

432

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 and, thanks to Ethernet and Internet technologies, optimum vertical integration. With EtherCAT, the costly Ethernet star topology can be replaced with a simple line structure - no expensive infrastructure components are required. Optionally, EtherCAT can also 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.

EtherCAT is versatile: Master to Slave, Slave to Slave and Master to Master Communication is supported (see figure below). 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, and outstanding real-time characteristics at low costs.

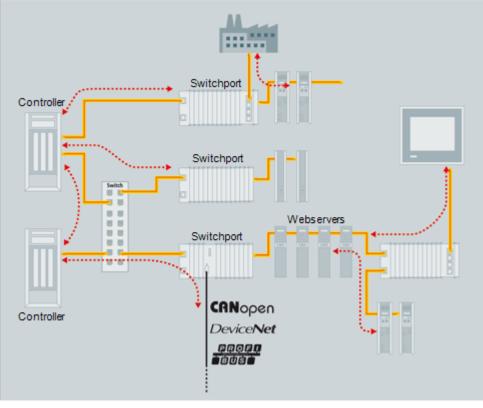


Figure 6-17: Versatile Network Architecture

8.3.2.1 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 up to 4 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 also enables any control with Ethernet protocol stack to address EtherCAT systems.

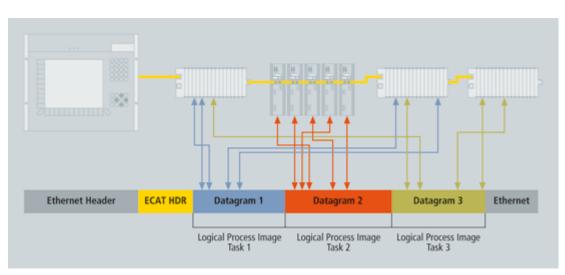


Figure 6-18: Process Data is Inserted in Telegrams

KAS supports **CANopen over EtherCAT** (CoE). It enables the advantages of EtherCAT in terms of transfer characteristics to be combined with proven, profile-specific drive functions.

KAS also uses **File Access over EtherCAT** (FoE) protocols to provide options for efficiently exchanging firmware via the bus (see Several Device Profiles and Protocols can coexist).

8.3.2.2 Topology

Using full-duplex Ethernet physical layers, the 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, such as line, tree or star. The bus or line structure known from the fieldbuses thus also becomes 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, and 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 (remember standard slave devices don't have a MAC address) attached to one switch port together forms an EtherCAT segment, which 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 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.

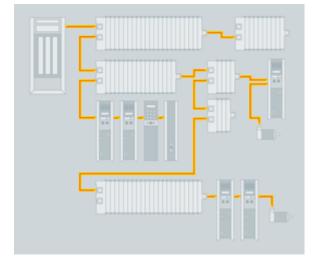


Figure 6-19: Flexible Topology: Line, Tree or Star

KAS controllers support line, tree, and star topologies. As the KAScontrollers (PCMM and AKD PDMM) have a singe EtherCAT port, ring topologies and cable redundancy is not supported.

8.3.2.3 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.

This mechanism also leads to very low jitter of significantly less than 1 µs. Even if the communication cycle jitters, it is still compliant with the IEEE 1588 Precision Time Protocol standard.

Therefore, 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 can then 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.

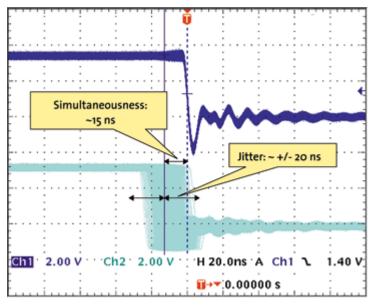


Figure 6-20: Synchronicity and Simultaneousness

Scope view of two distributed devices with 300 nodes and 120 m of cable between them.

8.3.2.4 Performance

KAS IDE User Guide | 8 Understanding KAS

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	100 µs
input and output data each	
1 Fieldbus Master-Gateway	150 µs
(1486 Bytes Input and	
1486 Bytes Output Data)	

Table 6-4: 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. The Distributed Clocks technique enables the axes to be synchronized with a deviation of significantly less than 1 microsecond. And even at this pace, there is more than sufficient bandwidth for asynchronous communications such as TCP/IP, parameter download or diagnostic data upload.

8.3.2.5 Safety over EtherCAT

The protocol enhancement called Safety over EtherCAT (FSoE) 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.

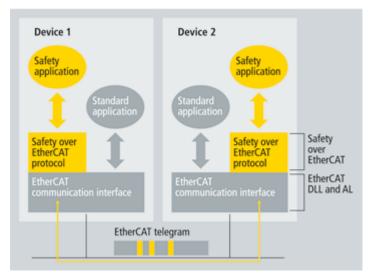


Figure 6-21: Safety over EtherCAT Software Architecture

436

8.3.2.6 Gateways

Gateway devices are available for the integration of existing fieldbus components (e.g., CANopen, DeviceNet) into EtherCAT networks. Also, 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. The EtherCAT network is fully transparent for the Ethernet device, and 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.

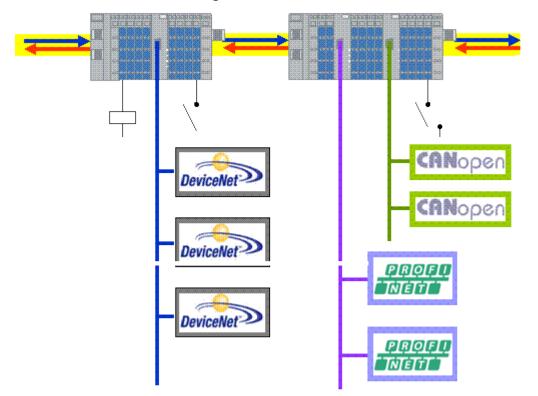


Figure 6-22: Fieldbus Gateway

8.3.2.7 Device profiles

The 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. Therefore, no EtherCAT-specific device profiles have been developed for these device classes. Instead, simple interfaces for existing device profiles are offered. This greatly assists users and device manufacturers alike during the change from existing fieldbuses to EtherCAT.

CANopen over EtherCAT (CoE)

CANopen device and application profiles are available for a wide range of device classes and applications, ranging from 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 thus 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.

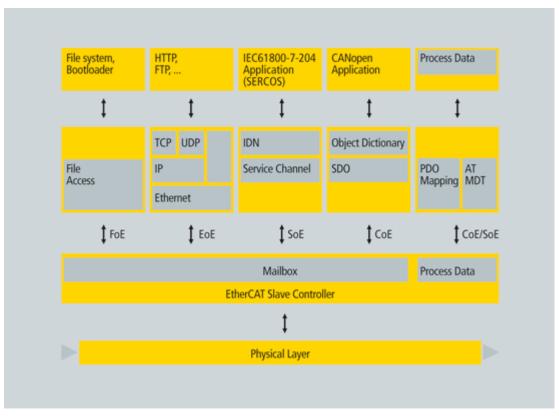


Figure 6-23: Several Device Profiles and Protocols can coexist

8.3.2.8 File Access over EtherCAT (FoE)

This very simple protocol, similar to TFTP, enables access to any data structure in the device. Therefore, standardized firmware upload to devices is possible, regardless of whether or not they support TCP/IP.

8.3.3 EtherCAT Implementation

The EtherCAT Technology 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. Thus EtherCAT uses standard Ethernet MACs, where they really make sense: in the master device. EtherCAT slave controllers are also used where such dedicated chips really make sense: in the slave device, where they handle the process data protocol in hardware and provide maximum real-time performance regardless of the local processing power or software quality.

8.3.3.1 Master Configuration

EtherCAT communicates a maximum of 1486 bytes of distributed process data with just one Ethernet frame. Therefore, 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, so 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: so 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.

For 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 (as shown in Master-Implementation with one Process Image). 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.

Ethernet Header ECAT E		EtherC	EtherCAT Datagram					
DA	SA	Туре	Frame HDR	EtherCAT HDR	Data	CTR	Pad.	FCS
(6)	(6)	(2)	(2)	(10)	(01486)	(2)	(032)	(4)
		\sim				\neg	<u> </u>	
	cons	stant He	ader		(mapped) process data	Working Counter: constant	and CF genera Ethern	RC ted by

Figure 6-24: Master-Implementation with one Process Image

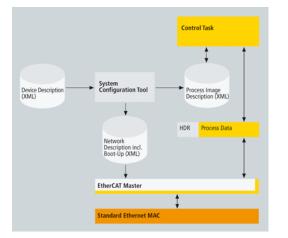


Figure 6-25: Structure of EtherCAT Master Implementation

8.3.3.1.1 ESI and ENI 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.

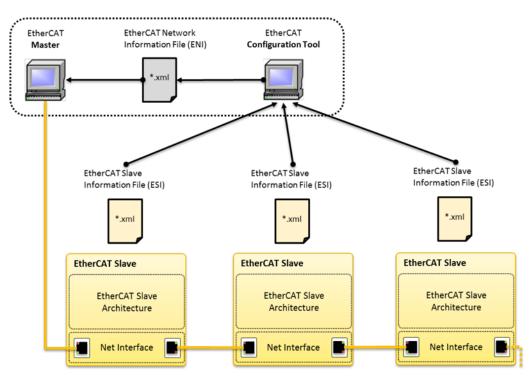


Figure 6-26: EtherCAT Network Architecture1

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 EtherCAT Network Information (ENI) schema version 1.3 (May 20, 2009). For more details, see the ETG.2100 specification at www.ethercat.org.

8.3.3.2 Slave Configuration

A cost-effective EtherCAT slave controller (ESC) is used in the slave devices. With EtherCAT the slave does not need a microcontroller at all. Simple devices that get by 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.

EtherCAT Slave Controller

440

1Image courtesy of EtherCAT.org, http://www.ethercat.org/pdf/english/etg2200_v2i0i1_slaveimplementationguide.pdf

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 particularly for devices with small process data quantity, such as 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 (as shown in Slave Hardware: FPGA with direct I/O).

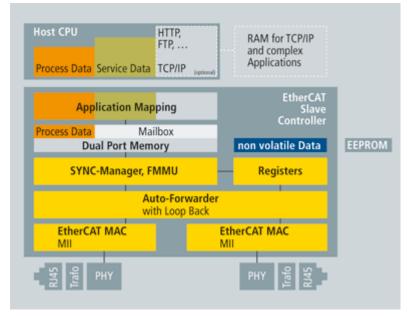


Figure 6-27: Slave Hardware: FPGA with Host CPU

8 I/O 8 I/O 8 I/O	
I/O Application EtherCAT Slave Controller	
Dual Port Memory non volatile Data	EEPROM
SYNC-Manager, FMMU Registers	
Auto-Forwarder with Loop Back	
EtherCAT MAC EtherCAT MAC MII	
RJ45 Trafo Trafo RJ45	

Figure 6-28: Slave Hardware: FPGA with direct I/O

8.3.3.3 EtherCAT State Machine

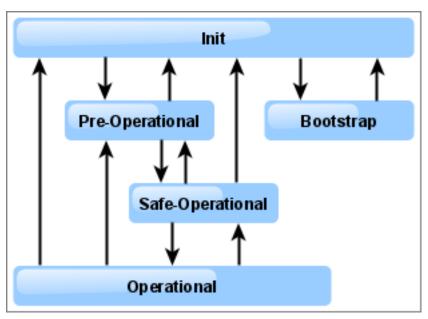


Figure 6-29: EtherCAT State Machine

Where the transitions are:

- from Init to Pre-Operational (Pre-Op): Master configures the Sync Manager channels for Mailbox communication
- from Pre-Op to Safe-Operational (Safe-Op): Master configures parameter using the Mailbox
- from Safe-Op to Operational (Op): Master sends valid Outputs

The different states are:

- Init No communication on the Application Layer Master has access to the DL-Information registers
- **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)
- Operational (Op) Inputs and Outputs are valid
- **Bootstrap** Recommended if firmware updates are necessary No Process Data communication Communication via Mailbox on Application Layer Only FoE protocol available

8.3.3.4 PDOs for AKD, AKD2G, AKD-N, and S300/S700

The KAS Motion Engine interacts with the AKD, AKD-N, and S300/S700 drives through CANopen objects in the selected PDOs. Some of the ML and MC function blocks require specific CANopen objects in the PDO(s).

The default AKD and AKD-N PDO selection includes all of the CANopen objects needed by the motion engine and function blocks. The default S300/S700 PDOs include only the minimal CANopen objects required by the KAS motion engine.

The following tables identify which CANopen objects are required by the motion engine and function blocks, and whether they are available in the default PDO for AKD, AKD-N, or S300/S700, and their corresponding AKD drive parameter There are two types of PDOs:

- **RxPDO** from Controller to Drive
- TxPDO from Drive to Controller

IMPORTANT

Check these tables to make sure the objects needed by the features your application uses are included in the AKD, AKD-N, and S300/S700 PDO selection.

8.3.3.4.1 From Controller to Drive (RxPDO)

AKD Drive CANOpen Objects Available in the Default PDO

Index - subindex	Object Name		Associated ML FB	Associated MC FB	Associated AKD Parameter				
0x20A4 - 0 or 0x2802 - 0	Latch control word	No	MLAxisCfgFastIn, MLAxisTimeStamp, all Trigger MLTrig FB	MC_ TouchProbe, MC_ AbortTrigger	CAP0.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 ¹	IL.BUSFF				
0x60C1 - 1 or 0x6062 - 0	Position demand value	Yes	Related to Axis pipe bloc positions (for more details, About Associated Data on Positions)	ReadParam	PL.CMD				
0x60FE - 1	Digital outputs (used by Onboard I/O mappings)	No	N/A	N/A	DOUTx.STATE				
AKD2G Drive	AKD2G Drive CANOpen Objects Available in the Default PDO								
Index - subindex	Object Name R	equired A	ssociated ML FB A Fl	ssociated MC 3	Associated AKD Parameter				
0x3601:1	Physical N	o N	VA N	/A	DOUT#.STATE				

1means Not Applicable

Outputs (used by Onboard I/O mappings)

Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated AKD Parameter
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 / 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 /0x6870:0	Target Position Axis 1 / 2	Yes	Needed for motion engine to work properly; Related to Axis pipe block positions. For more details see About Associated Data on Positions	Needed for motion engine axis to work properly: MC_ ReadParam (ID 1)	
0x60B8:0 / 0x68B8:0	Touchprobe function Axis 1 / 2	No	MLAxisCfgFastIn, MLAxisTimeStamp, all Trigger MLTrig FB	MC_TouchProbe, MC_ AbortTrigger, MC_Reference, MC_MachRegist, MC_MarkRegist, MC_StopRegist	

NOTE

Axis 1 objects begin with 0x60, Axis 2 objects begin with 0x68.

AKD-N Drive CANOpen Objects Available in the Default PDO								
Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated AKD Parameter			
0x20A4 - 0 or 0x2802 - 0	Latch control word	No	MLAxisCfgFastIn, MLAxisTimeStamp, all Trigger MLTrig FB	MC_ TouchProbe, MC_ AbortTrigger	CAP0.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 ¹	IL.BUSFF			

1means Not Applicable

Index - subindex	Object Na	me Re	equired	Associated ML FB	Associated MC FB	Associated AKD Parameter
0x60C1 - 1 or 0x6062 - 0	Position demand va	Ye alue	25	Related to Axis pipe block positions (for more details, About Associated Data on Positions)	MC_ ReadParam (ID 1)	PL.CMD
0x60FE - 1	Digital outputs (used by Onboard I/O mappings)		No N/A		N/A	DOUTx.STATE
S300 and S700 L	Drive CANO	pen Obj	ects Ava	ilable in the Default PDO		
Index - subindex	Object Name	Require	ed Ass	ociated ML FB	Associated MC FB	Associated AKD Parameter
0x20A4 - 0 or 0x2802 - 0	Latch control word	No	MLA	xxisCfgFastIn, xxisTimeStamp, all Trigger rig FB	MC_ TouchProbe, MC_ AbortTrigger	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	MLA	xisAddTq	N/A ¹	IL.BUSFF
0x60C1 - 1 or 0x6062 - 0	Position demand value	Yes	posi <mark>Abo</mark>	ted to Axis pipe block tions (for more details, ut Associated Data on tions)	MC_ ReadParam (ID 1)	PL.CMD

8.3.3.4.2 From Drive to Controller (TxPDO)

AKD Drive CANOpen Objects Available in the Default PDO

Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated AKD Parameter
0x6041 - 0	CANopen Status Word	Yes	N/A	N/A	N/A
0x6063 - 0 or 0x6064 - 0	Position actual value	No	MLAxisFBackPos, MLAxisReadActPos	MC_ReadActPos	PL.FB

1means Not Applicable

Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated AKD Parameter
0x2050 - 0 or 0x35C9 - 0	Position actual value 2	No	MLAxisRead2ndFB NOTE This FB is deprecated. We recommend reading the actual position of the axis which is assigned to the Sampler with this CoE object.	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)
0x606C - 0	Velocity actual value	No	MLAxisReadVel	MC_ReadActVel	VL.FB
0x6077 - 0	Torque actual value	No	MLAxisReadTq	MC_ReadParam (ID 1016)	IL.FB
0x20A5 - 0 or 0x2901 - 0	Latch status word	No	MLAxisCfgFastIn, MLAxisTimeStamp, all Trigger MLTrig FB	MC_ TouchProbe _, MC_ AbortTrigger	CAPx.STATE
0x20A6 - 0 or 0x2902 - 0	Latch position	No	MLAxisCfgFastIn, MLAxisTimeStamp, all Trigger MLTrig FB	MC_ TouchProbe, MC_ AbortTrigger	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
0x3470-4	AIN.VALUE (used by Onboard I/O mappings)	No	N/A	N/A	AIN.VALUE
0x60F4	Following error	No	MLAxisReadFEUU	MC_ReadParam (ID 1006)	PL.ERR

446

Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated AKD Parameter
0x3780:1	AIN1.VALUE	No	N/A	N/A	
0x3781:1	AIN2.VALUE	No	N/A	N/A	
0x6041:0 / 0x6841:0	Statusword - 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	
0x6064:0 / 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 / 0x6877:0	Torque Actual Value - Axis 1 / 2	No	MLAxisReadTq	MC_ReadParam (ID 1016)	
0x60B9:0 / 0x68B9:0	Touch Probe Status Axis 1 / 2	No	MLAxisCfgFastIn, MLAxisTimeStamp, all Trigger MLTrig FB	MC_TouchProbe, MC_AbortTrigger, MC_Reference. MC_MachRegist, MC_MarkRegist, MC_StopRegist	
0x60D1:0 / 0x68D1:0	Touch Probe 1 Time Stamp positive value Axis 1 / 2	No	MLAxisCfgFastIn, MLAxisTimeStamp, all Trigger MLTrig FB	MC_TouchProbe, MC_AbortTrigger, MC_Reference. MC_MachRegist, MC_MarkRegist, MC_StopRegist	
0x60D4:0 / 0x68D4:0	Touch Probe 2 Time Stamp positive value Axis 1 / 2	No	MLAxisCfgFastIn, MLAxisTimeStamp, all Trigger MLTrig FB	MC_TouchProbe, MC_AbortTrigger, MC_Reference. MC_MachRegist, MC_MarkRegist, MC_StopRegist	
0x60E4:3	3rd additional position actual Value Axis 1	No	MLSmpConECAT	For a digitizing axis	
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 / 0x68F4:0	Following error actual value Axis 1 / 2	No	MLAxisReadFEUU	MC_ReadParam (ID 1006)	

AKD2G Drive CANOpen Objects Available in the Default PDO

NOTE

Axis 1 objects begin with 0x60, Axis 2 objects begin with 0x68.

Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated AKD Parameter
0x6041 - 0	CANopen Status 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 (ID 1016)	IL.FB
0x20A5 - 0 or 0x2901 - 0	Latch status word	No	MLAxisCfgFastIn _, MLAxisTimeStamp _{, all} Trigger MLTrig FB	MC_ TouchProbe, MC_ AbortTrigger	CAPx.STATE
0x20A6 - 0 or 0x2902 - 0	Latch position	No	MLAxisCfgFastIn, MLAxisTimeStamp, _{all} Trigger MLTrig FB	MC_ TouchProbe _, MC_ AbortTrigger	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
0x60F4	Following error	No	MLAxisReadFEUU	MC_ ReadParam (ID 1006)	PL.ERR

S300 and S700 Drive CANOpen Objects Available in the Default PDO

Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated AKD Parameter
0x6041 - 0	CANopen Status Word	Yes	N/A	N/A	N/A
0x6063 - 0 or 0x6064 - 0	Position actual value	No	MLAxisFBackPos, MLAxisReadActPos	MC_ReadActPos	PL.FB

Index - subindex	Object Name	Required	Associated ML FB	Associated MC FB	Associated AKD Parameter
0x2050 - 0 or 0x35C9 - 0	Position actual value 2	No	MLAxisRead2ndFB NOTE This FB is deprecated. We recommend reading the actual position of the axis which is assigned to the Sampler with this CoE object.	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 (ID 1016)	IL.FB
0x20A5 - 0 or 0x2901 - 0	Latch status word	No	MLAxisCfgFastIn, MLAxisTimeStamp, all Trigger MLTrig FB	MC_TouchProbe, MC_AbortTrigger	CAPx.STATE
0x20A6 - 0 or 0x2902 - 0	Latch position	No	MLAxisCfgFastIn, MLAxisTimeStamp, all Trigger MLTrig FB	MC_TouchProbe _, MC_AbortTrigger	CAPx.T (for time) CAPx.PLFB (for position)
0x60F4	Following error	No	MLAxisReadFEUU	MC_ReadParam (ID 1006)	PL.ERR

8.3.3.4.3 Examples

Below are three examples where the PDO object is passed as an argument in the function block.

```
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.

```
MLCNVConnectEx(PipeNetwork.CNV1, PipeNetwork.AXIS1, EC_ADDITIVE_TORQUE_
VALUE, 0 );
```

The argument is a constant based on the object index.

```
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

Please note that 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.

8.3.4 CANopen

8.3.4.1 CANopen Status Machine

The states of the status machine can be revealed by using the CANopen Status Word.

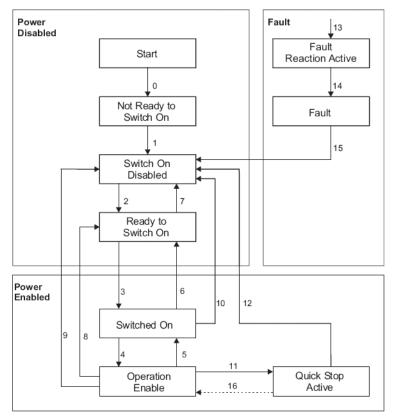


Figure 6-30: CANopen Status Machine

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.

Status description

Status	Description
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
Switch on disabled	The drive cannot be enabled via the EtherCAT interface; because for example there is no connection to a power source
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.
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.
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.

Status	Description
Quick stop active	The drive has been stopped with the quick stop ramp; output stage is enabled; motion functions are not enabled.
Fault reaction active	A fault has occurred and the drive is stopped with the emergency stop ramp
Fault	A fault is active, and the drive has been stopped and disabled

Table 6-5: Status Description

Transitions of the status machine

The drive device supports the transitions and actions as listed in the table below. The event initiates the transition. The transition is terminated after the action has been performed.

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 and 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, and 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, and 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.
12	Automatic transition when the quick stop function is completed and quick stop option code is 1, 2, 3 or 4, or disable voltage command received from control device (depends on the quick stop option code)	The drive function has to be disabled, and 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.

Transition	Event	Action
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	Enable operation command from control device, if the quick stop option code is 5, 6, 7, or 8	The drive function has to be enabled.

Table 6-6: Transition Events and Actions

8.3.4.2 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. Such commands are built up from the logical combination of the bits in the control word and external signals (e.g. enable output stage).

Bits definition of the 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
11	reserved
12	reserved
13	Manufacturer-specific
14	Manufacturer-specific
15	Manufacturer-specific

Table 6-7: Bit Assignment in Control Word

The commands are coded as given in the table below.

Command	Bits of the control word					Transitions	
Command	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0		
Shutdown	0	Х	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 (Note)	
Disable voltage	0	Х	Х	0	Х	7,9,10,12	
Quick stop	0	Х	0	1	Х	7,10,11	
Disable operation	0	0	1	1	1	5	
Enable operation	0	1	1	1	1	4,16	
Fault reset	up	Х	Х	Х	Х	15	

Table 6-8: Command Coding

Note: automatic transition to Enable operation state after executing SWITCHED ON state functionality.

Bits marked by an X are irrelevant.

8.3.4.3 CANopen Status 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.

Bits definition of the 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)
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)

Table 6-9: Bit Assignment in Status Word

The bit combinations coding the following states are listed in the table below.

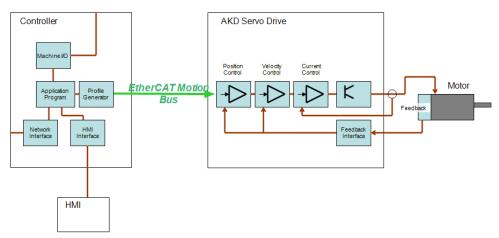
Status word MSB <i>(1512) (11 8) (7 4) (3 0)</i> LSB	State
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Table 6-10: State Coding

Bits marked by an X are irrelevant

8.4 AKD Drive

The **servo loops** in a KAS system are located within the AKD Drive. The **profile generator** used for all the motion in your application is located in the controller (AKD PDMM or PCMM).



8.4.1 AKD Drive Configuration

8.4.1.1 Connection Modes

The AKD drive parameters can be configured when the KAS IDE is connected to the controller and operating in one of two functional modes:

- Online Configuration Mode
- Project Running on the Controller

Disconnected

If the controller is not operating in one of the above modes, then the AKD(s) configurations are not accessible. The Configuration tab will indicate the AKD is Disconnected:

General Properties I/O	Configuration	PDO Selection/Mapping	Distributed Clock	CoE Init-Commands	CoE Object-Dictionary	
000						
Vollmorgen Device	e (AKD_1)	Disconne Drives are not availa		Ø	Learn more about this topic	
		1. Connect the IE 2. Compile and d	w, please follow these DE to your controller us ownload the applicatio configuration mode.	ing the connect button on to the target.	n the toolbar. 🎇	ш
		OR 3b. Start the appl	ication. 💟			•

Online Configuration Mode

456

In Online Configuration mode parameters are updated directly to the AKD2G or AKD with EtherCAT is 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.

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. When you modify the value of a parameter, 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.

NOTE

The AKD drive leave Online Configuration mode and start an the application if you start an application while the drive is in that mode.

NOTE

Important! Before leaving Online Configuration mode (either by clicking the Online Configuration button or clicking the Start button when the device is in Online Configuration mode) you should ensure the following conditions exist.

- 1. Drive is not enabled
- 2. Service motion is not enabled
- 3. Auto-tuning is not enabled

Recommendation: Clicking on the Disable button will force the above scenario.

Disable | Stop | 1 - Field Bus + | 2 - Position Mode + | Disable & Clear Faults | Save To Drive | Disconnect

8.4.1.2 AKD Configuration According to EtherCAT State

The drive configuration can only take place when EtherCAT is in the following state: Pre-Op or Op.

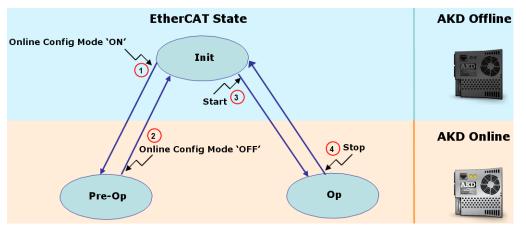


Figure 6-31: AKD Configuration According to EtherCAT State

Action	Name	Description
1 Online Configuration Mode		Online Configuration Mode (to see where you can access this button in the KAS IDE, see "Online Configuration Mode" on page 252)
"ON"	Sets the EtherCAT fieldbus to the Pre-Op state and allows AKD configuration (for procedure, see "Configure the AKD2G/AKD Drive" on page 183)	
2	2 Online Configuration Mode	Sets the EtherCAT fieldbus to the Init state and does not allow AKD configuration.
"OFF"	All AKD drives are disconnected from the IDE.	

Action	Name	Description
3	Start the Project	Sets the EtherCAT fieldbus to the Operational state ¹ (to see where you can access this button, see "Menus and Toolbar Overview" on page 241) This will start the application program running and allows AKD configuration (with some restrictions for the views: Service Motion and Performance Servo Tuner)
4	Stop the Project	This step stops the application from running, sets the EtherCAT fieldbus to the Init state, and does not allow AKD configuration. All AKD drives are disconnected from the IDE.

Table 6-11: AKD Drive - List of Actions

458

¹Depending 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.

8.5 Tasking Model / Scheduling

In the KAS Runtime, both the Motion and Programmable Logic Controller (PLC) Programs are run every cycle. The cycle update time is set when configuring the EtherCAT motion bus (see EtherCAT Master Settings Tab).

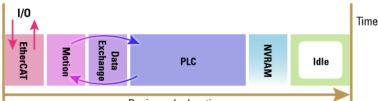
The cycle time becomes effective only when the Motion Engine is started (i.e. when the PLC code initializes the Motion by calling the MLMotionIni function block), and the application runs on a controller.

The time base remains much longer than the cycle time as long as the Motion Engine is **not** yet started, or if the application runs on the KAS Simulator (for more details, When the application runs on the KAS Simulator, the PLC execution rate is approx. 10 milliseconds. KAS Simulator cannot execute the PLC programs faster because Windows is not able to handle timing less than 10ms..). In these cases, the PLC execution rate is approx. 10 milliseconds.

8.5.1 Priority Between Motion and PLC

The Motion computation is always executed each cycle, and occurs before executing the PLC programs application. The figure below shows the execution in the following 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 (for more details, EtherCAT Processing Time)
- 4. PLC programs are executed
- 5. NVRAM variables are saved (for more details, NVRAM Processing Time)



Basic cycle duration

Figure 6-32: Priority Between Motion and PLC

NOTE

The Motion time (see figure above) 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.

8.5.1.1 EtherCAT Processing Time

The EtherCAT frame is executed at the beginning of the cycle. During this period, all the values related to EtherCAT (PDO) are exchanged, including:

- Inputs are read
- Outputs are set

Based on the I/O mapping to PLC variables, the I/Os are updated before they are effectively used during the PLC period.

As a consequence, when the PLC variables set an Output, it is updated during the EtherCAT frame of the next cycle.

8.5.1.1.1 About Variation during the EtherCAT Processing

The EtherCAT period is subject to time variation along the cycles due to the following reasons:

- Some EtherCAT function blocks are using the asynchronous SDO communication, which is not deterministic.
- Some EtherCAT slave devices support mailbox protocols.
 The master cyclically reads the mailbox of the EtherCAT slaves (polling of mailbox is performed every 50 cycles and is spread on several cycles depending on the number of EtherCAT slaves)

See also the FAQ about SDO communication.

8.5.1.2 NVRAM Processing Time

Due to a slow processing when saving the Retain Variables to the NVRAM, this action is not performed each cycle. The save operation is performed in the background every 20 seconds (frequency increases to each 2 seconds when the application is running).

When executed during a cycle, it occurs after the PLC period.

8.5.1.3 What happens when a PLC Program is overrunning the Cycle Duration

Large application can require more than one cycle to completely execute all the PLC programs.

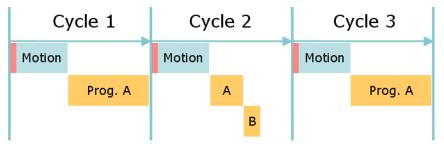


Figure 6-33: Application Overrunning the Basic Cycle

The figure above shows 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

NOTE

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.

IMPORTANT

If Outputs are set when a program runs over several basic cycles, unexpected and potentially dangerous effects can happen.

NOTE

460

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.

See Also: Priority Between PLC Programs

8.5.2 Priority Between PLC Programs

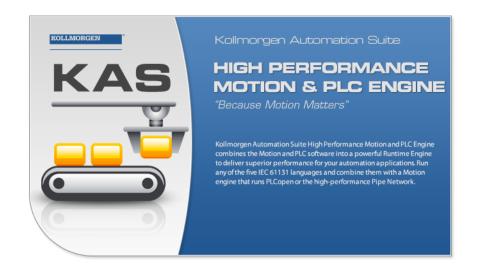
In turn, PLC programs are assigned a priority. At times of heavy demand for processing time, the operating system serves programs with higher priority first.

For more details, see how to:

• Set the PLC cycling

See Also: Priority Between Motion and PLC

9 Using the KAS Simulator



9.1 Start KAS Simulator

The **KAS Simulator** is launched from within the KAS IDE. Click on the KAS Simulator button in the toolbar.



Clicking the button will perform the following:

- 1. Change the KAS project to use KAS Simulator.
- 2. Launch the KAS Simulator.

★ TIP

Each version of KAS IDE has a matching version of Simulator which it will open. Only one instance of KAS Simulator may be run at a time.

3. Connect the KAS IDE to the KAS Simulator.

The first time KAS Simulator is run it will attempt to open some TCP/IP ports to allow communication. Your system's firewall will detect this and prompt for an action. Allow the KAS Simulator to open the ports by selecting Unblock (Windows XP) or Allow Access (Windows 7).

Windows Sec		II has blocked some features of this program
V		e features of Kollmorgen Automation Suite on all public, private
and domain netw	orks. Name:	Colmorgen Automation Stite
	Publisher:	Kollmorgen
	Path:	C: \program files\kollmorgen\kollmorgen automation suite \sinope simulator\bin\kassimulator.exe
Allow Kollmorgen	Automation Suite	to communicate on these networks:
Domain ne	tworks, such as a	a workplace network
Private ne	tworks, such as r	ny home or work network
		ose in airports and coffee shops (not recommended ten have little or no security)
What are the risk	s of allowing a pr	ogram through a firewall?
		Allow access Cancel

462

Figure 6-34: Firewall alert dialog.

NOTE

Simulator uses port 80 for the web server. This is mandatory for proper communication. Before starting Simulator, please close any application, such as VOIP, Skype, or IIS, that may use port 80. If another service is using port 80, you will receive a prompt to close the application and retry Simulator. See Diagnosing What Software is Using Port 80 for more information.

🗙 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 in connected to Simulator, owns Simulator.

9.1.1 Diagnosing What Software is Using Port 80

If you are experiencing trouble determining what software is using port 80:

1. Run netstat -o from a command prompt. This will output a list of ports and the process ID using the port. In this example, process 4000 is using port 80.

```
C:\Users\Admin>netstat -o
Active Connections
 Proto Local Address
                             Foreign Address State
PID
                             blugro5relay:2492
 TCP
       xxx.xxx.xxx.xxx:2492
                                                 ESTABLISHED
5232
 TCP
        xxx.xxx.xxx.80
                             173:http
                                                  ESTABLISHED
4000
 TCP
        xxx.xxx.xxx.80
                             173:http
                                                  ESTABLISHED
4000
 TCP
        xxx.xxx.xxx:53405 cs115p1:5050
                                                 ESTABLISHED
1688
 TCP
        xxx.xxx.xxx:53416 bos-m001c-rdr2:https ESTABLISHED
1688
       xxx.xxx.xxx.xxx:53418 chat-d03b-rdr2:https ESTABLISHED
 TCP
1688
 TCP
        xxx.xxx.xxx.xxx:53428 mtnradsvk1200:52230
                                                 ESTABLISHED
2076
 TCP
       xxx.xxx.xxx.xxx:53442 pb-in-f125:5222
                                                ESTABLISHED
4868
C:\Users\Admin>
```

- 2. Open Windows Task Manager, switch to the Processes tab.
- 3. Click View > Select Columns... and ensure that PID (Process Identifier) is selected.
- 4. Sort the Windows Task Manager by the PID column to easily find the name of the process which is using port 80.

9.1.2 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.

📇 Kollmorgen Automation Su	uite Simulator - [Axes]	
💌 File Help		
Axes	Custom IO Editor	
DriveAX	IS 1	DriveAXIS 2
265.33(0		003.12(0.00)
Log		₽ ×
Filter *		•
Time	Source Leve	9
1 🔎 2/24/2012 1:15:03 PM	(153) Unknown DEBUG	Loading buses and IOs drivers
2 🔎 2/24/2012 1:15:03 PM	(206) Unknown DEBUG	Initialisation done !!!
٠		Þ
		IP0: 10.50.67.88

Figure 6-35: KAS Runtime Log Window

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 KAS Simulator log window

9.2 Stop KAS Simulator

The KAS Simulator can be closed in several ways:

- from within the KAS IDE by clicking on the KAS Simulator button in the toolbar.
- by selecting File > Exit.
- by clicking on the Close button at the top right of the KAS Simulator.



Clicking the KAS Simulator button will perform the following actions.

- 1. Stop the project if it is running.
- 2. Disconnect the KAS IDE from the KAS Simulator.
- 3. Close the KAS Simulator.
- 4. Change the project back to using a PxMM as the controller.

Selecting File > Exit or the Close button will stop the project and disconnect the KAS IDE from the KAS Simulator, but the project will 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 Application.

★ 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 in connected to Simulator, owns Simulator.

9.3 KAS Simulator Axes Tab

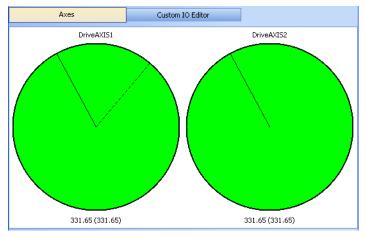


Figure 6-36: Axes Tab

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 the following format:

```
Range value (Modulo value according to the periodicity)
```

As shown on the figure below, the **Error** command (in the contextual menu of the axis tab) is used to simulate an error on an axis (then you can see the impact on the HMI and implement counter-measures if necessary).

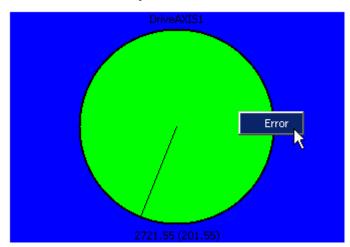


Figure 6-37: Set Axis in Error Mode

The drive becomes Red when it is set to Error . Motion State Machine

To deselect an axis already selected (blue rectangle), click on the white surrounded outside border of the axis tab.

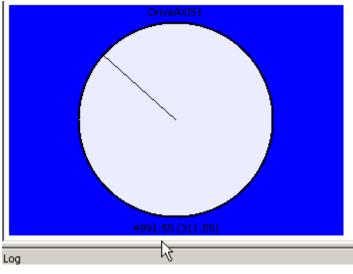


Figure 6-38: Deselect an Axis

9.4 Describing KAS Simulator Graphical User Interface

9.4.1 Windows Overview

9.4.1.1 Main window

KAS Simulator main window contains:

- The menu bar (see call out 1)
- The workspace⁽²⁾
- The Log window 3

In addition, the workspace displays the Axes.

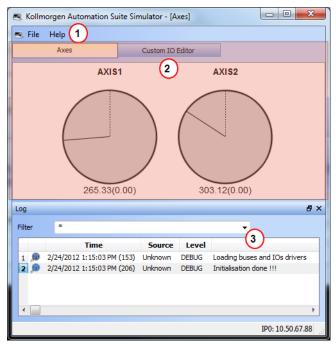


Figure 6-39: KAS Simulator Main Window

9.4.1.2 KAS Simulator log window

This Log window shows all log messages related to the KAS Simulator. Error and warning messages issued from the operating system, as well as Printf Function instructions, are also placed on this window.

Every log message includes the following:

- Timestamp
- Source
- Logging Level
- Message

9.5 KAS Simulator Menus Description

9.5.1 File Menu

Command	Description
Start	Start the application with the Retain Variables.
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

Options	? X
Auto Start:	
Ok	Cancel

Figure 6-40: Options for KAS Simulator

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\

9.5.2 Help Menu

Command	Description
About	Show version numbers and other Find KAS IDE Version about KAS Simulator



10 Using the AKD PDMM or PCMM Controller

Tasks related to the controllers are:

- Configure parameters
- Start and stop your KAS application
- Update the firmware
- Reset to factory settings

Rebooting the controller, recovering the firmware, and resetting the controller may be performed from the device or, more conveniently, using the web server.

NOTE

Please note that any reference to AKD PDMM or PCMM refers to both the 800MHz and the singleand dual-core 1.2GHz variants, unless otherwise noted.

10.1 Booting the PDMM or PCMM

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

10.1.1 Boot Sequence

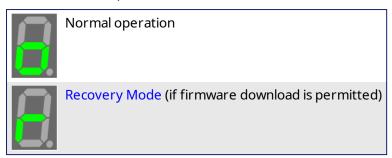
State	Display	Description
Hardware power on		Controller option card has power
Stage 0	8	Reached after the I ² C is initialized
Stage 1		Reached after the DDR3 ram memory is initialized
Stage 2	B	Reached just after the RAM memory relocation At this point the boot is running in DDR3 RAM memory
Stage 3		Reached after the flash memory is initialized
Boot Startup Script	8	After all the previous steps, the startup script starts automatically.
QNX startup	8	Reached after the Boot startup script is finished
Sysinit	8	Reached after specific configuration parameters of the target are loaded, and after the network is started using the rotary switch.
NOTE		

NOTE

The controller may be booted with or without an Ethernet cable attached.

When the controller is booted with a cable attached the configured IP address (depending upon the current position of the rotary switch) will be displayed in the 7-segment display (see Display the IP Address of a PDMM or PCMM). If the controller is started without a network connection then the IP address will not be displayed.

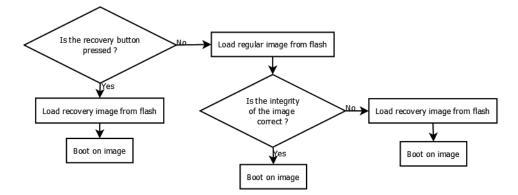
After the boot sequence is successful, the controller will be in one of two modes:



10.1.2 Boot Startup Script

After all the previous steps, the startup script starts automatically. The script first puts the 7-segment display into stage 4.

Before the controller boots up, the following flowchart applies:



10.1.3 Booting from the Recovery Image

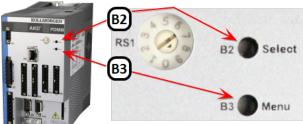
Automatic Mode	The boot from the recovery image is done automatically if the regular image is corrupted.
Manual Mode	If the controller starts booting normally but freezes after the startup script (see image to the right), then you have to boot manually from the recovery image by pressing the recovery button (B2). See Booting from the Recovery Image for more information.

NOTE

The controller will display 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.

10.2 Working with the Hardware

In some cases, using the buttons on the controller may be preferable to using the web server. On the front of the controller there are two buttons, B2 and B3. B2 is above B3. These buttons may be used to enter Recovery Mode (see "Booting from the Recovery Image" on page 472), Display the IP Address of a PDMM or PCMM, stop and start the application, reset the control to factory settings



(see "Reset the Controller to Factory Settings" on page 473), and backup/restore the firmware.

Result	Press and hold		
B2	Recovery Mode		
B3	Menu access		

Table 7-1: B2/B3 button functionality at start-up

Press	Result

B2 Menu option selector

B3 Menu access

Table 7-2: B2/B3 button functionality while running

10.2.1 AKD PDMM or PCMM Memory

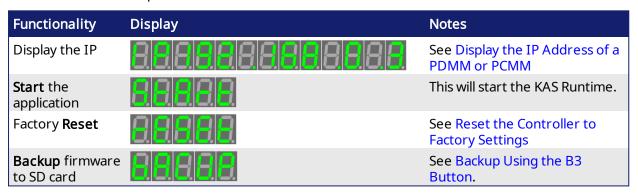
The AKD PDMM is equipped with ample memory to handle the most challenging programs.

Memory Type	Amount	Purpose
Flash	64 MB	Non-volatile memory
SD Card slot	2+ GB	Backup and Restore functionality as well as moving data. See SD Card Support for more information.
DDR RAM	256 MB	everything else

10.2.2 PDMM or PCMM B3 Button Menu

The B3 "Menu" push-button will cycle through a list of menu items displayed on the 7-segment LED. Pressing and holding the B3 button during the boot sequence (before the Boot Startup Script runs) provides access to a menu of functions. Each B3 press will advance to the next menu item. The menu item will be displayed for 10 seconds. Press B2 to select the current menu item. If no button is pressed within the 10 seconds, the 7-segment display will return to Normal operation.





Functionality	Display	Notes
Restore firmware from SD card	8888888	See Restore Using the B3 Button.

Table 7-3: Application is not running



Table 7-4: Application is running

NOTE

Please note that when selected, the **Start**, **Stop**, **Backup**, **Restore**, and **Reset** functions do not initiate immediately; they require confirmation. The 7-segment displays flashes a "y", prompting for confirmation. Pressing B2 confirms the function and the process begins. If the function is not confirmed within 10 seconds the action is canceled.

10.2.3 Display the IP Address of a PDMM or PCMM

The IP Address assigned to the controller can be shown on the 7-segment display. The IP may be displayed at boot and can be accessed from the PDMM or PCMM B3 Button Menu. Note that there is a 5 second delay before this function may be used again.



Figure 7-1: Example of the IP sequence by the 7-segment display.

NOTE

The AKD PDMM or PCMM will not set (or show) an IP address without an attached network cable.

10.2.4 Booting from the Recovery Image

To enter recovery mode you must press and hold B2 during the boot sequence before the Boot Startup Script runs. If the system detects that the button is pressed then it will enter Recovery Mode. The 7-segment display will show a lower-case "r" as seen here.



While in Recovery Mode the AKD PDMM or PCMM will download the firmware from the recovery image. When the firmware is being written to the flash drive the 7-segment display will animate as seen below. Do not power-off the system during this process.



472

When the download is complete the controller will go into normal operation. If the download or write to flash fails the 7-segment display will display a numeric error code.

10.2.5 Reset the Controller to Factory Settings

The AKD PDMM or PCMM can be manually triggered to perform a factory reset. The reset is performed using either of two methods:

- From the File System Tab of the web server while the drive is running. *This method is recommended due to its ease of use.*
- Selecting the function from the PDMM or PCMM B3 Button Menu. This can be done during the boot sequence or while the drive is running.

After two seconds have expired (or longer if pressed during power-up), the 7-segment display on the control will change to an animation pattern indicating that the factory reset has started.

The following changes occur during a factory reset:

- Reset any application previously downloaded
- Reset IP address, Subnet and Gateway settings
- Reset retained variables
- Reset Auto-Start option
- Reset password to the default
- Reset shared directory settings

Some important facts to remember:

- Factory reset cannot be performed while an application is running.
- If the AKD PDMM or PCMM has just been powered up, the B3 button will have to be held down much longer than 2 seconds. In this case, hold down the button until the 7-segment display shows the PDMM or PCMM B3 Button Menu.
- Factory reset will take about 4-5 minutes to complete and the 7-segment display on the controller will animate during this process. The control should not be turned off during this procedure.
- After the factory reset is complete, the control will be powered down and restarted automatically.

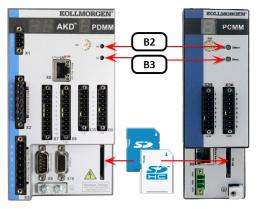
10.2.5.1 Resetting While the Drive is Running

A reset may be done any time after the controller is powered on and an application is not running. Please note that the reset will be ignored if an application is running on the AKD PDMM or PCMM.

$\boxed{\textbf{B}}$	Normal Operation	Press the B2 button to access the menu and using B3 button scroll to the "reset" option in the menu. Press B2 to confirm the "reset" to factory defaults.
B	Program Running	Reset to factory defaults is not permitted. The "reset" menu item is not available.

10.2.6 SD Card Support

The AKD PDMM and PCMM controllers support using an SD card for backup and restore functionality. This lets you manage the configuration, application and operation data. The controller has a SD card slot and push buttons (B2 and B3) which activate file transfers to and from a SD card.



Using the SD card provides an easy way to

- backup and restore a AKD PDMM or PCMM configuration
- store and retrieve an application, including source code
- store and retrieve user data from an application or PC

10.2.6.1 Supported SD Card Formats



[‡] The default file system for the format.

IMPORTANT

We recommend using an SD card which is rated to a 85°C max. operating temperature. Cards which do not meet this criteria should not be left in the device.

10.2.7 Backup and Restore a PDMM or PCMM

The SD card is used for backing up and restoring the data from a AKD PDMM or PCMM.

Backing up may be performed in either of two ways.

- Using the web server, which is recommended as it has functionality the button method does not offer, such as backing up the drive data. For more details, see Backup & Restore from the Web Server.
- Using the PDMM or PCMM B3 Button Menu on the front of the controller. See Backup Using the B3 Button for more details.

🗙 TIP

For a complete backup of both the controller and drive, use the Web Server backup.

Restoring information to the controller may be performed in either of two ways. Both methods restore the controller (and PDMM drive) with whatever was saved to the SD card.

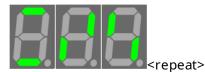
- Using the web server. For more details, see Backup & Restore from the Web Server.
- Using the B3 button menu on the front of the controller. See Restore Using the B3 Button for more details.

Backup & Restore What?	To/From SD card via B2/B3 buttons	To/From SD card via Web Server
Controller: Firmware	✓	✓
Controller: Network Configuration		✓
Controller: PLC Application		✓
Controller: Retain Variables		✓
EtherCAT Devices: Network Topology	×	✓
EtherCAT Devices: AKD Drive Firmware	× ‡	×
EtherCAT Devices: AKD Drive Parameters	×	✓

‡ Restore only

10.2.7.1 About the data transfer

• The 7-segment display will show the chasing lights animation while the backup or restore is occurring.



- The Backup and Restore functions have an "all or nothing" behavior. If there is no SD card inserted, if there is not enough space on the card or if files are missing then nothing will be copied and the 7-segment display will show an error.
- If files already exist on the SD card (in the backup directory), then they will be deleted and replaced with the new AKD PDMM or PCMM backup configuration files. Likewise, the files on the controller will be replaced with the SD files.

NOTE

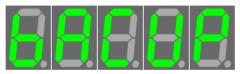
Warning! Do not modify the files on the SD card as this could result in the Restore function failing.

🗙 TIP

If you have multiple AKD PDMM or PCMM backup configurations, you will need to use one SD card per backup configuration.

10.2.7.2 Backup Using the B3 Button

One of the functions accessible from the B3 button is backing up the controller. The Backup function will store a copy of the AKD PDMM or PCMM's data on a SD card. This function is displayed on the 7-segment display as shown here ("bACUP"). Pressing B2 selects the function. This function does not initiate automatically, B2 must be pressed again to confirm the process.



The data that is backed up and copied to the SD card includes:

- PLC Application (including ECAT XML configuration, cam tables, etc.)
- AKD PDMM or PCMMconfigurations (auto-start and IP address)
- PLC application
- Retain Variables

NOTE

Controller Log files and EtherCAT devices are not copied to the SD card. They can be backed up using the Backup Tab in the Web Server.

10.2.7.3 Restore Using the B3 Button

One of the functions accessible from the B3 button is restoring the controller and PDMM's drive. The Restore function will restore and load files onto the controller from an SD card. This function is displayed on the 7-segment display as shown. Pressing B2 selects the function. This function does not initiate automatically. The 7-segment displays flashes a "y", prompting for confirmation. Pressing B2 again confirms the function and the data transfer begins. If the function is not confirmed within 10 seconds the action is canceled.



The data that is restored from the SD card includes:

- PLC Application (including ECAT XML configuration, cam tables, etc.)
- AKD PDMM or PCMMconfigurations (auto-start and IP address)
- PLC application
- Retain Variables

10.2.8 EtherCAT Devices Backup and Restore

The AKD PDMM or PCMM can backup/restore EtherCAT devices on an EtherCAT network. This feature is useful as a maintenance operation to replace any AKD or AKD2G drives in an operational machine. This feature reduces the manual steps for saving/loading each drive's firmware and parameters into a few simple automated steps. The Backup/Restore functionality is located in the controller's web server and is accessible from a web browser. For details about the web server see Using the KAS Web Server.

		KAS Application	Settings	Backup & Restore	Diagnostics	Help
	5		is stored on the	e SD card. The following fe r Online Configuration Mo EtherCAT devices back	de is active.	
Backup		Restore	Import/Expo	rt		
Controller						
	Do you want to add the 1.2 GHz firmware to the backup (800MHz is already included)? O YES NO Backup Controller Backup Controller Backup Controller					ents that are backed ion, retained ile an application is utes to complete. Do
EtherCAT Dev	vices					
Backup Ether	CAT Device	S		the AKD and/or A parameters. For EtherCAT dev and/or AKD2G fir * If a matching fir controller, it will b * If a matching fir controller, you will	T Devices will save th KD2G drives' firmware vices backup operation mware package is req mware package alread e used for future backum ware package does r I be asked to provide ii ckages are available al	e, and the drives' n, a matching AKD uired. ly exists on the ups. rot exist on the t.

See also Backup and Restore a PDMM or PCMM in the KAS IDE help.

10.2.8.1 EtherCAT Devices Backup

The Backup operation discovers all the devices on the EtherCAT network and stores the topology information, AKD/AKD2G firmware files, and AKD/AKD2G parameters to the controller's local storage.

Controller	Local Storage
AKD PDMM or PCMM	SD Memory Card (see <u>SD Card Support</u> for more information)

Before starting a backup of the EtherCAT Devices, you will need:

- The AKD or AKD2G firmware files package. This package is included in the KAS software installation directory (\Program Files (x86)\Kollmorgen\Kollmorgen Automation Suite\Astrolabe\DrivesFW) or is available for download from the Kollmorgen website (www.kollmorgen.com/en-us/website-resources/other/akd-software/). You will be prompted with the expected file.
- An SD memory card must be in the SD slot if you are using an AKD PDMM or PCMM.
- All the EtherCAT network devices must be connected to the AKD PDMM or PCMM, and configured as necessary for machine operation.

🗙 TIP

Make sure 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, e.g. 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.

• A PLC application downloaded to the AKD PDMM / PCMM, containing the EtherCAT device map.

NOTE

The PLC application cannot be running and the IDE must not be in Online Configuration Mode. Please stop your PLC application or disable Online Configuration Mode before a Backup or Restore.

10.2.8.1.1 Backup Steps

- 1. From the web server home page, click on the Backup & Restore tab:
- 2. Under the Backup tab, press 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 will remember your selection and will not ask you again for future backups.

	Backup	Restore		Import/Export	
C	Backu	p Controller		control up are variabi Notes: * This i runnin * Back	function cannot be performed while an application is
E	therCAT Devices				
	Please provide the re-	quired firmware package ⁽¹):		p EtherCAT Devices will save the network topology, D and/or AKD2G drives' firmware, and the drives' eters.
	AKD-Firmware-for-KAS-V	/01-12-00-003.tgz		and/or	herCAT devices backup operation, a matching AKD AKD2G firmware package is required. hatching firmware package already exists on the
	Browse_ No file selec	ted.		control * If a m	ler, it will be used for future backups. hatching firmware package does not exist on the ler, you will be asked to provide it.
	<< Cancel			^(†) All fit	mware packages are available at Kollmorgen.com
	NOTE				

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. Press the Continue Backup EtherCAT Devices button. It will take a couple of minutes or longer, depending on the number of AKD and/or AKD2Gs in the system.

Backup	Restore	Import/Export	
Controller Backu	p Controller	controlle up are t variable Notes: * This fr running * Backu	Controller is used to replicate a PCMM or the er portion of a PDMM. The elements that are backed he firmware, network configuration, retained is, and PLC application. Inction cannot be performed while an application is ip Controller takes several minutes to complete. Do er off the controller once started.
EtherCAT Devices			
Continue Backup Eth	erCAT Devices	the AK parame For Eth and/or A * If a m * If a m	EtherCAT Devices will save the network topology, and/or AKD20 drives' firmware, and the drives' lets: arCAT devices backup operation, a matching AKD AKD26 firmware package is required. atching firmware package already exists on the er, it will be used for future backups: atching firmware package does not exist on the atching firmware package does not exist on the r, you will be asked to provide it.
		⁽¹⁾ All fim	nware packages are available at Kollmorgen.com

When the backup is complete, the web server will indicate 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 Import / Export Backup from Webserver.

10.2.8.2 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 will detect the replaced AKD and/or AKD2G devices and restore them. Advanced users can manually select the specific AKD or AKD2G 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
AKD PDMM or PCMM	SD Memory Card (see <u>SD Card Support</u> for more information)

10.2.8.2.1 Steps

IMPORTANT

The application will start 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 and the EtherCAT device order is correct. *Disable the Autostart option before commanding the Restore operation if you want to check the devices before starting the application.*

NOTE

The EtherCAT Devices Backup and Restore feature may be used in EtherCAT networks which have third party devices, but only the Kollmorgen AKD or AKD2Gs can be backup and restored. Specific configurations applied to third party devices with non-Kollmorgen tools have to be reapplied when the third party device is replaced.

- 1. From the web server home page, click on the *Backup & Restore* tab:
- 2. Under the *Restore* tab, you can choose to restore the replaced EtherCAT AKD and/or AKD2G devices or manually select the devices with the Advanced view.
 - To restore the replaced AKD or AKD2G devices, press the Restore EtherCAT Devices button. When the restore is complete, the web server will indicate whether it was successful.
 - To select the AKD or AKD2G devices and manually restore:
 - 1. Press the Advanced link.
 - 2. Select the devices you want to restore. The controller will identify the replaced drives and pre-select them for you.
 - 3. Press Restore selected devices.

EtherCAT Devices
Restore EtherCAT Devices
<< Advanced
Restore operation allowed: topologies match (Backup date: Tue, 28 Oct 2014 20:03:42 GMT)
Physical topology
→ PDMM Controller
V - KD AKD AKD-M00306-MCEC-0000 (SN: R-1251-00134)
– 🔚 2: AKT Coupler
□ L鞋I3: AKD AKD2G-SPE-6V06D-A1F3-0000 (SN: -1728-00029)
Restore selected devices Refresh

When the restore is complete, the web server will indicate whether it was successful.

10.2.8.3 AKD Backup/Restore Compatibility

The replacement AKD or AKD2G must have the same model number as the device that was originally stored in the backup. The *Restore* operation compares the model numbers between the backup and the replacement AKD. The restore will not be allowed if they are not compatible.

- To check the model number on your AKD or AKD2G, see the sticker on the side of the drive.
- To check compatibility between your replacement drive and the backup, replace the AKD or AKD2G on the EtherCAT network, and press the Scan network button.
 - The web server displays the physical topology and allows you to restore the selected drives if the backup and replacement drives are compatible.

	Backup & Res	Backup & Restore		
		Note that most of the following features are not available when PLC application is running or online config mode is active		
		EtherCAT devices backup: Mon, 29 Oct 2012 08:40:08 GMT		
Backup	Restore	Import/Export		
Controller				
Restore	Controller	Restore Controller is used to replicate a PCMM or the controller portion of a FDMM. The elements that are restored are the firmwein, network configuration, relained variables, and FLC application. Notes: * Restore Controller takes several minutes to complete. Do not power of the controller one started.		
* The controller is rebooted after a Restore.				
Restore EtherCAT Devices				
<< Advanced				
Restore operation allowed: topologies match (Backup date: Tue, 07 Jul 2015 18:12:48 GMT)				
Physica	Physical topology			
🗢 Controller				
📝 🗕 🏹 0: AKD AKD-M00306-M1EC-D000 (SN: R-1432-00169)				
🔲 🗕 🚳 1: AKD AKD-P00306-N	AEC-0000 (SN: R-0934-00008)			
L 🔚 2: Beckhoff Automation GmbH device 0x4602C22				
Restore selected devices	Refresh			

• The web server displays the backup and physical topology and indicates the noncompatible drives if the backup and replacement drive are not compatible

	Backup & Restor	e	
	Note that most of the following features are not available when PLC application is running or online config mode is active		
		Ether	AT devices backup: Tue, 07 Jul 2015 22:11:15 GMT
Backup Restor	re Import/E	xport	
Controller			
Restore Controll	er		Restore Controller is used to replicate a PCMM or the controller portion of a PDMM. The elements that are restored are the firmware, network configuration, retained variables, and PLC application. Notes: Restore Controller takes several minutes to complete. Do not power off the controller once stated.
* The controller is rebooted after a Restore.			
Restore EtherCAT Devices			
<< Advanced Restore operation not allowed: topologies are not matching (Backup date: Tue, 07 Jul 2015 22:11:15 GMT)			
			p date: Tue, 07 Jul 2015 22:11:15 GMT)
Physical topol	ogy		Backup topology
🗢 Controller		, (Controller
- 🔯 0: AKD AKD-P00306-NAEC-000	IO (SN: R-1116-00140) new !	- 20	0: AKD AKD-M00306-M1EC-0000
- 🗃 1: AKD AKD-P00306-NAEC-000	0 (SN: R-1116-00091)	- 10	1: AKD AKD-P00306-NAEC-0000
L 🏭 2: Beckhoff Automation GmbH d	levice 0x4602C22	Refres	

★ TIP

The serial number (SN) is displayed in the Physical topology web server view and on the sticker affixed to the AKD. You can use the serial number to match the actual hardware with its representation on the web server.

10.2.8.4 Import / Export Backup from Webserver

10.2.8.4.1 Controller Backup Import / Export

Controller backup data may be saved to or written from a local drive or USB memory stick. This functionality is available when a controller backup file (see Backup Tab) exists on the device's SD card.

During the import process this function will replace an existing backup file without further prompting, but should the process be interrupted or fail, it will not overwrite the backup file.

10.2.8.4.2 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 allows you to transfer a backup file onto the controller to be used later for restoring a previous configuration.

Backup	Restore	Import/Export	
Controller			
Export backup: Import backup:	Export Controller Backup	rowse	Export Controller Backup: Export EtherCAT Backup: Controller or EtherCAT devices backup file is saved to this computer. Note: The Export Controller/EtherCAT button is only displayed when the respective backup is available.
EtherCAT Device	s		Import Controller Backup: Import EtherCAT Backup: Controller or EtherCAT devices backup file (.TGZ file) is
Export backup: Import backup:	Export EtherCAT Backup	rowse	transferred from this computer to the controller. This backup file is used when you need to restore a previous configuration. Note: The Replace Backup button is disabled until a backup
	Replace EtherCAT Backup		file has been selected. *The backup file name for import should be: 'KAS-controller-backup figz' for Controller backup. 'KAS-network-backup figz' for EtherCAT devices backup.

10.2.8.4.3 Export Procedure

NOTE

The Export button is only displayed if an AKD or AKD2G backup is available.

- 1. Click on the "Export EtherCAT Backup" button. The browser starts transferring a backup file. Depending upon the browser being used, this may involve a prompt confirming that you wish to receive the file.
- 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.

10.2.8.4.4 Import Procedure

- 1. Specify a backup file to import by clicking on the "Browse" button. This backup file will be used to replace the current backup on the controller.
- 2. After a backup file is specified, click on 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 will be replaced. If the import fails, the previous backup will not be replaced.

NOTE

482

- The Replace Backup button is disabled until a backup file has been selected.
- On some browsers, the "Browse" button may be labeled "Choose File".

10.2.8.5 EtherCAT Devices Backup/Restore Limitations

• The EtherCAT Devices Backup and EtherCAT Devices Restore functions are not permitted while a PLC application is running or when the IDE is in Online Configuration Mode. Please stop your

PLC application or disable Online Configuration Mode before a Backup or Restore.

Only a single backup is supported in the controller at one time.

NOTE

If you want to keep multiple backup configurations, you can perform the backup and then export it to a local PC or USB flash memory stick. Later, when you are ready to restore, you can import the specific backup file to the controller, and then perform the restore.

🗙 TIP

Beware that an import will overwrite any backup existing in the controller or AKD PDMM SD card. If you are using an AKD PDMM, the alternative method is to use a separate SD card for each backup/restore configuration.

- All drives on the EtherCAT network must have the same firmware version.
- The AKD firmware version must be 01-06-00-003 or higher.
- AKD and AKD2G firmware packages are available for all production releases.
- Only AKD and AKD2G drives are supported for backup/restore. Kollmorgen S300/S700 drives are not supported by backup/restore.
- AKD PDMM or PCMM system backup or restore is a two-step process:
 - 1. Backup or Restore the AKD PDMM or PCMM controller
 - 2. Backup or Restore the drives (including the AKD drive inside the AKD PDMM).

10.2.8.6 Troubleshooting EtherCAT Devices Backup/Restore

The web server displays an error message if an EtherCAT Device backup or restore fails. The message describes the cause of the failure and a possible remedy. Please be sure to note any error message(s), as they will be helpful with remedying the problem.

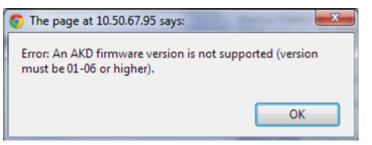
Described below are some common error messages and remedies. The message box format may appear differently depending on the web browser, but the message content is the same.

During AKD/AKD2G backup, EtherCAT fails to reach pre-op mode:

🕤 The page at 10.50.67.95 says:	x
Error: The EtherCAT network failed to reach pre-op mode. The KAS application's expected network topology does no match the discovered topology.	
ОК	

- 1. Using the 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 do one of the following:
- Modify the application's devices to match the physical network.
- Correct the physical network by adding/moving/removing nodes.

During AKD/AKD2G backup, at least one drive has an unsupported firmware error for backup:



- 1. Using the IDE, open the EtherCAT view.
- 2. Upgrade all of the AKDs drive firmware to at least version 01-06.

During AKD/AKD2G restore, the backup file is corrupt:

😨 The page at 10.50.67.95 says:	×
Error during restore: The backup is corrupt.	
	ОК

Before restoring AKD/AKD2G(s), the backup file must be valid. To correct a bad backup file on a controller, import a known good backup file from another source (local hard drive, network drive, USB flash stick, etc.). If you do not have a valid backup file, then you will need to manually configure the replacement drives by downloading firmware and modifying the drive's parameters using the AKD/AKD2G views in the IDE.

Not able to restore AKD/AKD2Gs successfully.

The error message will describe at which step the restore failed. For example, failing to restore parameters:

💿 The page at 10.50.67.95 says:
Error during restore: The AKD drive(s) parameters could not be restored.
ОК

- 1. If an AKD/AKD2G 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 your network cables and try a different replacement drive(s). This test will isolate the problem to the specific drive(s) or the controller problem.
- 3. If you still cannot restore a replacement AKD/AKD2G, then you will need to manually configure the replacement AKD/AKD2Gs by downloading firmware and modifying the drive's parameters using the drive views in the IDE.

10.2.9 Configure Controller Onboard I/O

The procedure to define the local I/Os of the AKD PDMM or PCMM is very similar to the one for I/O slices. Note that the Onboard digital I/O is limited to a 1kHz update rate.

EtherCAT update rate	Onboard digital I/O update rate
500Hz	500Hz
1kHz	1kHz
2kHz	1kHz
4kHz	1kHz

For more details, refer to Map Input and Output to Variables

10.2.10 About Errors and Alarms

484

The AKD PDMM or PCMM controller will continuously display any error or alarm codes after booting, and not in recovery mode.

Only one error or alarm code will be displayed at a time. Errors have a priority over Alarms and the code with the highest priority will be displayed until it is cleared.



10.2.11 Using the KAS Web Server

Kollmorgen Automation Suite[™] comes with a web server that allows you to perform the following operations:

- Read information about the controller (model, Runtime/firmware version, version of your KAS application)
- 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

‡ AKD PDMM and PCMM only

The web server may be accessed two ways:

- 1. Open a web browser and enter the controller's IP address.
- 2. From the Controller node in the Project tree in the KAS IDE.
 - Double-click the Controller node
 - Select "Access Webserver" from the right-mouse menu.

NOTE

If you do not know the IP address assigned to the controller:

- 1. Press B3 once. The 7-segment display will flash the letters I and P.
- 2. Press B2 to select the IP option. The 7-segment display will show the IP address.

The web server consists of the home page, and the KAS Application, Settings, Backup & Restore, Diagnostics and Help tabs. The Help tab is a link which opens the Web Server manual.

KAS Application Settings Backup & Restore Diagnostics Help

Figure 7-2: The Webserver Tabs as seen on an AKD PDMM webserver.

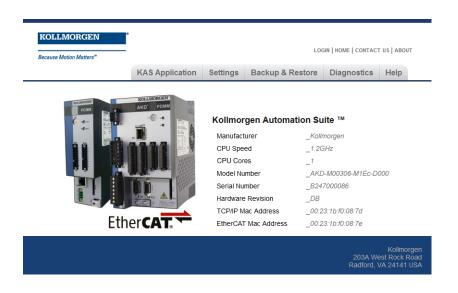
🗙 TIP

Browser Requirements: We recommend using Firefox 11, Google Chrome, or Internet Explorer 9 or later for accessing the web server.

10.2.12 Web Server Home Page

To access the KAS web server home page, enter the controller's IP address.

KAS IDE User Guide | 10 Using the AKD PDMM or PCMM Controller



This page provides an overview of the device including:

- Manufacturer
- CPU Speed ‡
- CPU Cores
- Model Number ‡
- 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.

‡ AKD PDMM and PCMM only

NOTE

Please note that any reference to AKD PDMM or PCMM refers to both the 800MHz and the singleand dual-core 1.2GHz variants, unless otherwise noted.

10.2.12.1 Security

Some parts of the web server are locked in order to protect critical operations from unauthorized users. Simply log into the web server to enable access to the locked functions. See User Authentication for more information.

NOTE

486

Functions will not be locked if you access the web server through the IDE. Doing so automatically grants administrator access.

The functions which are locked are:

- KAS Application Tab
 - Start/Stop/Cold Start an application
 - "Clear User Data"
 - "Clear all Errors"
 - Configure 'Auto-start'

- Settings tab
 - Firmware upgrade -->'Choose File' &'upgrade' ‡
 - Reboot ‡
 - "Reset to Factory Settings"
 - SD card Format ‡
 - Change password
 - Change the network settings (IP address) ‡
- Backup & Restore
 - Backup Controller ‡
 - Backup & Restore AKD PDMM and PCMM only
 - Restore Controller
 - Export Backup
 - Choose File & Replace Backup
- Diagnostics tab ‡
 - Reboot the controller
 - Clear Errors and alarms
 - Clear Crash dump

‡ AKD PDMM and PCMM only

10.2.12.1.1 Timeout After Inactivity

To prevent misuse, if the webserver has been idle (no keyboard activity or mouse clicks) for 20 minutes, the user account will be automatically logged out. A dialog box will open 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.

10.2.12.2 User Authentication

10.2.12.2.1 Logging In

Logging into the web server is required to prevent unauthorized access or changes. This is accessed from the **LOGIN** link at the top of all web server pages. Clicking the link brings up a form to enter user credentials.

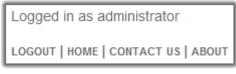
KOLLMORGEN			
Because Motion Matters™			
Username	administrator		
Password			
	Login Cancel		

Enter the password to log in. The factory default password is administrator. This can be changed after logging in.

NOTE

- As there is only one user Administrator, "administrator" is the default user name.
- You must re-enter the password each time you log in; the entered password is not stored in the login form.
- Accessing the webserver from the IDE automatically logs you in as *administrator*.

When you are successfully logged in, the user name will appear in the top-right corner of all web server screens.



10.2.12.2.2 Logging Out

After successfully logging in, the menu in the top right corner of the web server contains a link to **LOGOUT**. Clicking this link will immediately log you out of the web server, and informs you of this.

The page at ///////////////////////////////////	×
Logged out successfully.	
	ОК

10.2.12.2.3 Changing the Password

The user password is managed from the User Account section of the Settings tab. See User Account for more information.

10.2.13 KAS Application

488

This tab allows you to:

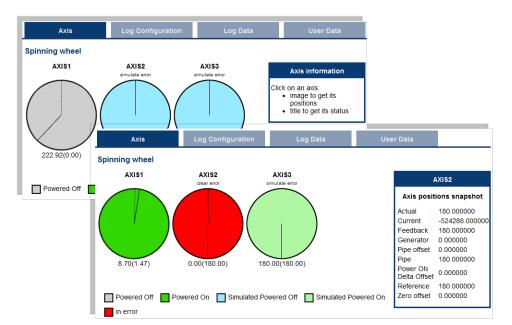
- Display general information about your project that is currently loaded on the controller
- Start and stop the motion
- Display the Axes run by the controller from the Axis tab
- Manage log messages from the Log Configuration and Log Data tabs
- Display User Data present on the controller from the User Data tab
- Connect to a remote computer from the Shared Directory tab

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>.</version></project_name>
	The application's source code may be downloaded to the local computer if it is present on the controller. This is accomplished by clicking on the download icon (). 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, <i>Started</i> or <i>Stopped</i> .
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.
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 will start 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 will use the new setting at the next power-up. This option is recommended when the system is in production.
Clear all errors	Clicking this button will clear the error log for all axes.

10.2.13.1 Axis

You can view a visual representation of the motors from the Axis tab. The axis wheels are visible after your application is started. The following 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, or Error as well as 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



Information that is available by clicking on the axis Title Image (PN axis) Image (PLCopen axis) Image (digitizing axis) AXIS1 AXIS1 PLCOpenAxis2 Axis3 Axis status snapshot Axis positions snapshot Axis positions snapshot Axis positions snapshot Initialised true Actual 174.22050 Actual 3141.856728 Actual 0.000000 0.000000 Power ON false Current Command 3141.856728 Normal Enable Feedback 174.220505 3141.856728 true Found Generator 0.000000 0.000000 true Phase Configured true Pipe offset 0 000000 Super imposed 0.000000 Running false Pipe 0.000000 Power ON Delta Error false 0.000000 See Axis Parameters in Offset Simulated false 0 000000 Reference Connected false the PLCopen Function Zero offset 0.000000 Warning false **Blocks - Overview** Stopping false section of the KAS IDE Stopped false See the Axis Pipe Block documentation for section of the KAS IDE information about the documentation for axis parameters. information about the axis parameters.

Additionally, if an axis is in error, the error can be cleared by clicking the text below the axis title.

10.2.13.2 Log Configuration

You can configure the log to filter the messages that are displayed. Each source can be set with its own level.

Ax	(is	Log	Configuratio	'n	Log Da	ta	User I	Data
Log Config	uration							
	IDE	Controller	EtherCAT	Drive	PLC	Motion	System	All
Error	\bigcirc	\odot	\bigcirc	\bigcirc	\bigcirc	0	\odot	0
Warning	۲	۲	۲	۲	۲	۲	۲	0
Info	0	0	0	\bigcirc	0	\bigcirc	0	\bigcirc
Debug		0	0	\bigcirc	0	\bigcirc	0	\bigcirc
Apply Refr	esh							

Each message has one of the following levels, with importance in descending order: Error > Warning > Info > Debug

🗙 TIP

How to Choose the Appropriate Level?

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

Therefore, DEBUG is the most verbose and ERROR is the least verbose level. Filtering is quicker with less verbose levels, due to the number of messages.

NOTE

Critical messages are always recorded. Therefore, the Critical level is not visible.

Source

Source	Apply to			
IDE	Win32 applications: the KAS IDE and the KAS Runtime Server (also called the KAS Runtime Front-end)			
Controller	For the KAS Runtime items: Drivers, IOEngine, SinopEngine			
EtherCAT	For all kinds of EtherCAT items: Motion bus, I/Os			
Drive	Messages from Kollmorgen drives			
PLC	For application engineers to create custom log within the PLC programs (similar to printf)			
Motion	Messages coming from the Motion engines: PLCopen, Pipe network or VM			
System	For common API and libraries. Also includes messages issued from the operating system.			

Level

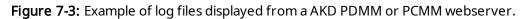
Level	Icon	Description
DEBUG	ø	Any information logged for development purpose. You may safely ignore this log.
INFO	1	Information status of the current process. You may safely ignore this log.
WARNING	▲	System is stable but the KAS IDE warns that an unexpected event can occur. You can ignore this log.

Level	Icon	Description
ERROR	8	The application does not behave as expected but the processes remain stable.
CRITICAL	*	Application crashes or becomes unstable. Data is corrupted. At this point the application behavior can be unpredictable.

10.2.13.3 Log Data

KAS log files may be viewed from the Log Data tab. These messages can help describe the current state of the system and to help identify any operation errors encountered when developing your system. A AKD PDMM or PCMM will display as many as 10 files.

	KAS Application	Settings	Backup & Restore	Diagnostics	Help	
Kalmorgan Automation Sute High PERFORMANCE Martin S PLC ENGINE Decare Motion Maters Decare information and the first of the term information and the term information and the term information and term information and term information and the term information and term information and ter		CEngine CEngine Information Antice	KAS Application Version of KAS App _UnnamedProject::3 Status of KAS App _stopped Start/Stop Start Cold Start S Options Image: Auto-start Apply Clear all axis errors Clear all errors			
Axis Log Files pdmm_log_000000000 pdmm_log_00000000000000000000000000000000000		Lo	ig Data	User Data		
Refresh						



Clicking on a listed log file will open it in your web browser. The log file may be downloaded by clicking on the green download icon next to the log entry. The default name is the same as the file's name. If you try to open a file that no longer exists, the message "/logfiles/<selected file name> not found." Refresh your browser window and try again.

10/20/2011	9:55:33	AM	(389)	Control	ler	WARN	ING HTTP server warning: Error 404: Not
10/20/2011	9:56:15	AM	(112)	System	DEBUG	HTTP	Files handler for GET request, URL is
10/20/2011	9:56:15	AM	(114)	System	WARNING	HTTP	Files handler, file 'E:/Kollmorgen/tru
10/20/2011	9:56:15	AM	(142)	System	DEBUG	HTTP	Kas handler for GET request, URL is ',
10/20/2011	9:56:15	AM	(155)	System	DEBUG	HTTP	Files handler for GET request, URL is
10/20/2011	9:56:15	AM	(531)	System	DEBUG	HTTP	Files handler for GET request, URL is
10/20/2011	9:56:15	AM	(531)	System	WARNING	HTTP	Files handler, file 'E:/Kollmorgen/tr
10/20/2011	9:56:15	AM	(545)	System	DEBUG	HTTP	Kas handler for GET request, URL is ',
10/20/2011	9:56:15	AM	(569)	System	DEBUG	HTTP	Files handler for GET request, URL is
10/20/2011	9:56:17	AM	(829)	System	DEBUG	HTTP	Files handler for GET request, URL is
10/20/2011	9:56:17	AM	(829)	System	WARNING	HTTP	Files handler, file 'E:/Kollmorgen/tr
10/20/2011	9:56:17	AM	(951)	System	DEBUG	HTTP	Kas handler for GET request, URL is '
10/20/2011	9:56:17	AM	(969)	System	DEBUG	HTTP	Files handler for GET request, URL is
10/20/2011	9:56:18	AM	(126)	System	DEBUG	HTTP	Files handler for GET request, URL is
10/20/2011	9:56:18	AM	(126)	System	WARNING	HTTP	Files handler, file 'E:/Kollmorgen/tr
10/20/2011	9:56:18	AM	(136)	System	DEBUG	HTTP	Kas handler for GET request, URL is '
10/20/2011	9:56:18	AM	(156)	System	DEBUG	HTTP	Files handler for GET request, URL is
10/20/2011	9:56:18	AM	(617)	System	DEBUG	HTTP	Files handler for GET request, URL is
10/20/2011	9:56:18	AM	(17)	System	WARNING	HTTP	Files radler, file 'E:// orgen/tr

Figure 7-4: Example of a log file's content, displayed in a browser.

NOTE

492

Log data is collected and updated every 15 seconds on a AKD PDMM or PCMM. A new log file will be 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. The AKD PDMM and PCMM log files both have the same file naming scheme (pdmm_log_xxxxxxxxx).

10.2.13.3.1 Log Message Content

Every log message in the table has the following 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.
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

Table 7-5: Log Messages - List of Field

🗙 TIP

Log messages is an important source of information when you are troubleshooting your project. When reporting an issue to Support, copy/paste the logs in your report.

10.2.13.3.2 PDMM and PCMM Log Files

Logs generated on a AKD PDMM or PCMM are stored in flash memory at /mount/flash/log. The files are stored in a rotating pool consisting of a maximum of 10 files. The files have a maximum size of 200 kilobytes each; the most amount of space the log files will consume is 2 MB. Once an "eleventh" file is created the earliest file is flushed to make room for the new file.

The AKD PDMM and PCMM generated log levels can be controlled form the KAS IDE and Web Server. From the IDE, the log levels can be filtered in the configuration window in the *Logs and Information* tab.

10.2.13.3.2.1 Log File Naming Convention

The logs have the naming format pdmm_logs_n where n is a value ranging from 0000000000 to 4294967295, which is the maximum value a 32-bit location can store.

As an example, when the files are first created they will be named pdmm_logs_0000000000, pdmm_logs_000000001, pdmm_logs_000000002 and so on. The file that will be created after pdmm_logs_4294967295 is pdmm_logs_0000000000. The naming gets reset and continues.

10.2.13.4 User Data

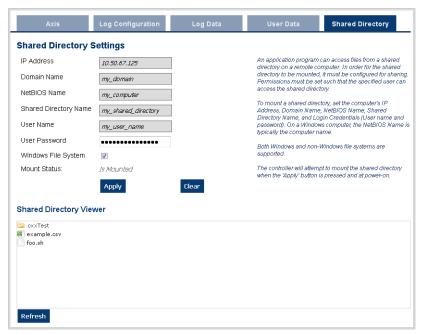
This tab lists any user-generated files or folders found in the AKD PDMM's, or PCMM's flash memory. Clicking a folder will display the folders contents. Clicking on the green download icon will immediately download the file.

Axis	Log Configuration	Log Data	User Data	
User Data				
 testDir test2bir test2.btt MC_MoveAdditivePosition/ MC_MoveAdditivePosition/ MC_MoveAdditivePosition/ TestOutput.csv test.xml \$ 	Array2.csv 📥			
Clear User Data Refresh				

The **Clear User Data** button will erase all of the files in the user data folder.

10.2.13.5 Shared Directory

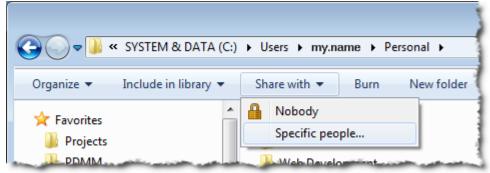
This tab provides the ability to connect to and mount a shared directory on a remote computer (Windows® 7 or better, and non-Windows). This allows a program to read and parse instructions for motion. Once the shared directory is mounted, the contents may be navigated in the **Shared Network Directory Viewer** section of this tab. The following fields must be set to connect to the share.



IP AddressEnter the IP Address of the computer that has a shared directory.Nomain NameThe network domain of the computer that is sharing the directory. This field may or may not be necessary.NetBIOS NameThe unique name that identifies the computer on the network. For Windows computers, this is typically the computer name.Shared Directory NameEnter the name of the shared directory as defined on the remote computer. The contents of this directory will be displayed in the viewer.User NameThe name of a user that is allowed access to the shared directory. If the user name is that of a domain user (instead of a local user), then the computer that hosts the shared folder must be connected to the domain in order to validate the name.User PasswordThe password of a user that is allowed access to the shared directory.Windows SystemIndicate whether the remote computer is Windows or a different operating system. This is used to automatically set file permissions.Mount suplication is running and when the user is not logged in.Pressing this button will mount de a power-on. This button is disabled when the application is running and when the user is not logged in.Clear running and when the user is not logged in.Pressing this button mill dear the above fields, unmount the share, and clear the configuration from flash memory. This button is disabled when the application is running and when the user is not logged in.		
Namemay not be necessary.NetBIOS NameThe unique name that identifies the computer on the network. For Windows computers, this is typically the computer name.Shared Directory NameEnter the name of the shared directory as defined on the remote computer. The contents of this directory will be displayed in the viewer.User NameThe name of a user that is allowed access to the shared directory. If the user name is that of a domain user (instead of a local user), then the computer that hosts the shared folder must be connected to the domain in order to validate the name.User PasswordThe password of a user that is allowed access to the shared directory.Windows File SystemIndicate whether the remote computer is Windows or a different operating system. This is used to automatically set file permissions.Mount statusThis field indicates whether the directory Is Mounted or Is Not Mounted.Apply Pressing this button will mount the share and 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.ClearPressing this button will clear the above fields, unmount the share, and clear the configuration from flash memory. This button is disabled when the application is	IP Address	Enter the IP Address of the computer that has a shared directory.
Namecomputers, this is typically the computer name.Shared DirectoryEnter the name of the shared directory as defined on the remote computer. The contents of this directory will be displayed in the viewer.User NameThe name of a user that is allowed access to the shared directory. If the user name is that of a domain user (instead of a local user), then the computer that hosts the shared folder must be connected to the domain in order to validate the name.User PasswordThe password of a user that is allowed access to the shared directory.Windows File SystemIndicate whether the remote computer is Windows or a different operating system. This is used to automatically set file permissions.Mount statusThis field indicates whether the directory Is Mounted or Is Not Mounted.Apply pulcation is running and when the user is not logged in.Pressing this button will clear the above fields, unmount the share, and clear the configuration from flash memory. This button is disabled when the application is		
Directory Namecontents of this directory will be displayed in the viewer.User NameThe name of a user that is allowed access to the shared directory. If the user name is that of a domain user (instead of a local user), then the computer that hosts the shared folder must be connected to the domain in order to validate the name.User PasswordThe password of a user that is allowed access to the shared directory.Windows File SystemIndicate whether the remote computer is Windows or a different operating system. This is used to automatically set file permissions.Mount StatusThis field indicates whether the directory Is Mounted or Is Not Mounted.ApplyPressing this button will mount the share and 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.ClearPressing this button will clear the above fields, unmount the share, and clear the configuration from flash memory. This button is disabled when the application is		
Namethat of a domain user (instead of a local user), then the computer that hosts the shared folder must be connected to the domain in order to validate the name.User PasswordThe password of a user that is allowed access to the shared directory.Windows File SystemIndicate whether the remote computer is Windows or a different operating system. This is used to automatically set file permissions.Mount StatusThis field indicates whether the directory Is Mounted or Is Not Mounted.ApplyPressing this button will mount the share and 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.ClearPressing this button will clear the above fields, unmount the share, and clear the configuration from flash memory. This button is disabled when the application is	Directory	
PasswordWindows File SystemIndicate whether the remote computer is Windows or a different operating system. This is used to automatically set file permissions.Mount StatusThis field indicates whether the directory Is Mounted or Is Not Mounted.ApplyPressing this button will mount the share and 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.ClearPressing this button will clear the above fields, unmount the share, and clear the configuration from flash memory. This button is disabled when the application is		that of a domain user (instead of a local user), then the computer that hosts the shared
File SystemThis is used to automatically set file permissions.Mount StatusThis field indicates whether the directory Is Mounted or Is Not Mounted.ApplyPressing this button will mount the share and 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.ClearPressing this button will clear the above fields, unmount the share, and clear the configuration from flash memory. This button is disabled when the application is		The password of a user that is allowed access to the shared directory.
StatusApplyPressing this button will mount the share and 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.ClearPressing this button will clear the above fields, unmount the share, and clear the configuration from flash memory. This button is disabled when the application is	File	
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.ClearPressing this button will clear the above fields, unmount the share, and clear the configuration from flash memory. This button is disabled when the application is		This field indicates whether the directory Is Mounted or Is Not Mounted .
configuration from flash memory. This button is disabled when the application is	Apply	so it can be automatically mounted at power-on. This button is disabled when the
	Clear	
Refresh Refresh the contents of the fields and the Shared Directory Viewer.	Refresh	Refresh the contents of the fields and the Shared Directory Viewer.

Example of how to share a directory from Windows:

- 1. In Windows Explorer, navigate to the directory you wish to share.
- 2. Select **Share with** from the Toolbar.



3. Using the resulting window, select the users and/or groups who that should have access to the directory.

★ 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.

NOTE

SMB1 must be enabled to use the Shared Directory feature with a Windows file system. Please review the KDN article "Shared Drive in PDMM or PCMM Application Fails to Mount" for information on configuring SMB1 in Windows.

See also Shared Directory Path Conventions.

10.2.14 Web Server Settings

This section allows you to:

- Display and update the firmware for the KAS Runtime‡
- Display the network settings and modify the IP address ‡
- Reset the control to factory settings
- Access the SD Card Actions ‡
- Access the User Account to change the password.

‡ AKD PDMM and PCMMonly

10.2.14.1 Firmware Tab (AKD PDMM and PCMM Only)

This tab displays the firmware version. Additionally, you may upgrade the firmware from this tab.

	KAS Application	Settings	Backup & Restore	Diagnostics	Help		
Settings Note that most of the following features are not available when PLC application is running or online config mode is active							
Firmware	Network	File system	SD Card	User A	ccount		
Firmware Information							
Firmware version	_2, 8, 1, 54368						
Recommended File Name	_KAS-PDMM-M-MCE	C-{version}.im	g				
Firmware upgrade							
Firmware image file ⁽¹⁾	Browse No file sel	ected.	(.img file) ⁽¹⁾ 2. Click Upgrade 3. After the downly needed to load the	firmware image file fo to start the updating p bad is finished, click e new firmware) es are available at Ko	rocedure Reboot (A reboot is		

10.2.14.1.1 Upgrading the Firmware

This section describes how to upgrade the AKD PDMM or PCMM's firmware using the web server. This operation downloads the KAS Runtime and its version number to the on-board flash memory in the controller.

The firmware files are IMG files with the following name format:

KAS-PCMM-M-{model-code}-{software-revision}.img KAS-PDMM-M-{model-code}-{software-revision}.img

The *model-code* reflects the CPU speed.

Model	CPU Speed	Cores
MCEC	800 MHz	single
M1EC	1.2 GHz	single
M2EC	1.2 GHz	dual

- 1. Download the latest drive firmware and/or runtime firmware from Kollmorgen.com.
- 2. Open the controller's web server in your web browser by entering its IP address.
- 3. Select the **Settings** tabbed-page.
- 4. In the **Firmware** pane, click the **Choose File** button to select the new firmware image file for the KAS Runtime.

The recommended file is displayed in the Current Information section, as seen below.

Firmware Information				
Firmware version	_2.10.0.54368			
Recommended File Name	_KAS-PDMM-M-MCEC-{version}.img			

5. Click **Upgrade** to start the update procedure.

🗙 TIP

If the Upgrade button is disabled, log into the webserver. Click on **Login** at the top of the web page and enter the password. See User Authentication for more information about logging in.

A message and a <u>throbber</u>1 are shown across the web page, indicating that maintenance is in progress. The device's 7-segment display will animate chasing lights.

Successful upgrade	A message similar to the following is shown upon a successful firmware upgrade:
	Upload of firmware KAS-PDMM-M-MCEC-2.10.0.54368.img successful.
	Please reboot the unit in order to boot on the new firmware, and once reboot is performed, press CTRL+F5 in your web browser to force a page refresh.
Incompatible firmware	An error message similar to the following will be displayed 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"

6. After the download is complete, click **Reboot** (for more details on the boot sequence, refer to Booting the PDMM or PCMM).

¹An animated element which indicates something is in progress. Interactions should wait until the process is complete.

A message and a throbber are shown over the web server while the reboot is in progress. The login session will no longer be valid when the reboot is complete. The web server will display a message to indicate the user has been logged out.

NOTE

This step is not necessary of the controller automatically reboots during the upgrade (previous step).

7. Press **CTRL+F5** to force the web browser to refresh the page.

① IMPORTANT

Do not try to refresh the web page until firmware upgrade is done.

10.2.14.1.2 Recovery Mode (PDMM or PCMM Only)

If the AKD PDMM or PCMM detects a problem in the firmware, it displays an "r" on the 7-segment display and will automatically enter Recovery Mode. Recovery Mode provides the ability to select and upgrade a firmware image file containing the KAS Runtime image on the AKD PDMM or PCMM. In the rare case when Recovery Mode cannot be automatically accessed, pressing and holding B2 at boot will force the AKD PDMM or PCMM to boot into Recovery Mode.

IMPORTANT

Warning! The Recovery Mode allows any firmware image file to be loaded into the AKD PDMM's or PCMM's flash memory. The Recovery Mode does NOT check the selected firmware file to verify it's compatibility with the hardware model. Check that 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 will fail to boot into the Runtime image and will fail to automatically boot into Recovery Mode. To recover from this situation requires manually booting into Recovery Mode. For more details, see Booting from the Recovery Image.

10.2.14.2 Network Tab (PDMM or PCMM Only)

The contents of this tab display the current rotary switch position of the AKD PDMM or PCMM and its MAC address. Additionally, you may manually change the AKD PDMM's or PCMM's IP address.

Firmware	Network	File system	
Network Settings			
Rotary Switch Value * MAC Address	_1 _00:23:1b:00:df:df	- 0 for E - 1 for n	otary Switch Position are: hHCP (if no DHCP server, AutoIP Is used) nanual IP address (by default: 192.166.0.101) static IP address (192.168.0.10x)
Manual IP Address **		** This positior	IP address will be used only if the rotary switch is on 1
IP Address	10 50 67	95 The net	v IP address will be effective after reboot
Subnet Mask	255 255 255	0	
Default Gateway			
	Apply Reboot		

Figure 7-5: Example of a AKD PDMM or PCMM with a manually defined IP address

10.2.14.2.1 About the Rotary Switch

The rotary switch on the AKD PDMM or PCMM can be set on a position from 0 to 9.

Switch Position	Description
Position 0	The drive tries to get an IP address from a DHCP server. If the DHCP fails, then the AKD PDMM or PCMM uses AutoIP to get a usable IP address.
Position 1	The default custom static IP address, 192.168.0.101 or a custom IP address.
Position 2-9	The AKD PDMM or PCMM is pre-configured with static IP addresses ranging from 192.168.0.102 (Position 2) to 192.168.0.109 (Position 9).

★ TIP

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

10.2.14.2.2 Change the IP Address

To connect and use your AKD PDMM or PCMM within your computer network, you may configure its IP address by using the web server as follows:

- 1. Open the controller's web server in your internet browser
- 2. Select the **Settings** tabbed-page
- 3. In the **Network** pane, set static IP address according to the position defined via the rotary switch
 - If the rotary switch is set to Position 1 you may use the default custom address or set a value in the Manual IP Address fields.
- 4. Configure the Manual IP Address
- 5. Configure the subnet mask (default is 255.255.255.0)
- 6. (Optional) Configure the gateway address if the AKD PDMM or PCMM is outside your local network
- 7. Click Apply
- 8. Click Reboot

🗙 TIP

498

It is recommended to not use leading zeroes when entering an IP address. The webserver automatically configures numbers with leading zeroes as octal values. For example, use **10.1.1.10** instead of **010.001.001.010**.

10.2.14.3 File System Tab

This section contains a button which allows you to reset the control to the factory settings. The steps to reset the controller vary slightly based on the platform.

	Firmware	Network	File system				
Cu	rent File system				D		
R	eset to Factory Settings		1. Reset a 2. Reset II 3. Reset n	actory Settings will: ny application previously download P address, Subnet and Gateway settings stained variables uto-Start option		1	
	File system	1				1.	AKD PDMMor
	Current File s	ystem			2	2	PCMM Simulator
	Reset to Factory	Settings	1. F 2. F	set to Factory Settings will: Reset any application previously download Reset retained variables Reset Auto-Start option		2.	Simulator
			* R the	les: eset cannot be performed while an application is eset will take about 1 minute to complete and the control will animate during this process. Do not po throl once started.	display on		

Figure 7-6: File System tab on an AKD PDMM or PCMM web server, and when using Simulator.

10.2.14.3.1 Reset to Factory Settings

When this button is pressed, the control will be reset to factory default settings. The user is prompted to confirm this action before the function is performed.

The following changes occur during factory reset:

- Reset any application previously downloaded
- All log files and user data files are erased
- Reset the IP address, Subnet, and Gateway settings ‡
- Reset any retained variables
- Reset the Auto-Start option
- Reset the password to default

‡ AKD PDMM and PCMM only

Notes about the reset:

- The factory reset cannot be performed while an application is running. The "Reset to Factory Settings" button is disabled while an application is running.
- The factory reset will take 4-5 minutes to complete and the 7-segment display on the control will animate during this process. The control should not be turned off during this procedure.
- After the factory reset is complete, the controller will be powered down and restarted automatically.
- This webpage will not update during the reset procedure and can be closed.
- AKD PDMM or PCMM Only: After the controller is restarted, the IP address of the control may change based on the controls rotary switch. If the rotary switch is at position 0, the same IP address as before should be assigned to the control. If the rotary switch is set to 1-9, a preconfigured IP address will be defined and must be taken into account when trying to reconnect to the webpage using a web browser.

10.2.14.4 SD Card Tab (PDMM or PCMM Only)

10.2.14.4.1 SD Card Actions

The *Format* function formats the SD card as FAT32, erasing all data from the card. This function cannot be performed while an application is running.

10.2.14.5 User Account

To change the password you must enter the current password and the new password twice.

Change Passwor	d
Current Password	
New Password	
Confirm New Password	
	Change Password

The new password must meet the following conditions:

- It must be 6-20 characters long
- It may not contain semicolons (;), ampersands (&), spaces, quotes (' and ''), slashes (/ and \), or the number sign(#).

10.2.14.5.1 | forgot my password

Should this happen, you can set a new password from the IDE.

- 1. Open the webserver from the IDE.
- 2. Click on the Settings tab.
- 3. Click on the User Account tab.
- 4. Enter and confirm the new password.

This allows you to create a new password without entering the current one.

10.2.15 Backup & Restore from the Web Server

These functions are used to replicate a AKD PDMM or PCMM (*Backup* and then *Restore*) to or from the SD card. The elements that are backed up or restored are the firmware, the network configuration, the Retain Variables, and the PLC application.

- These functions cannot be performed while an application is running.
- *Restore* and *Backup* take several minutes to complete. Do not power off the control once started.
- The Controller is rebooted after a *Restore*.

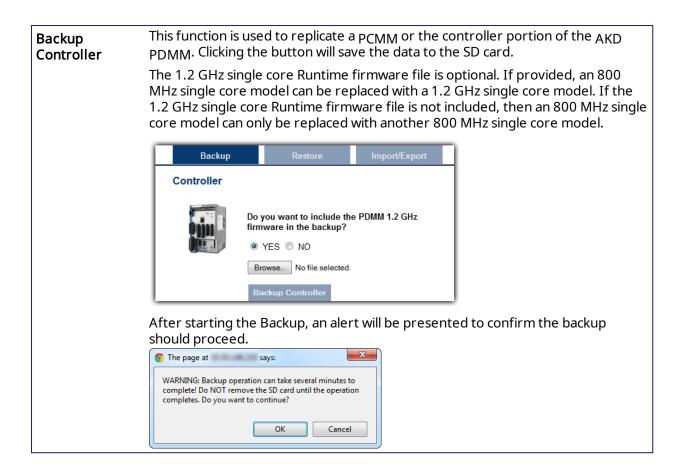
The AKD PDMM and PCMM are available with different CPU variants (800 MHz, 1.2 GHz single core, and 1.2 GHz dual core). The AKD PDMM and PCMM models have the same functionality, but due to the CPU speed differences an application designed for the 1.2 GHz model may not execute as expected on the 800 MHz model. Additionally, an application designed for a single core CPU model may not execute as expected on the dual core model. The Restore will not allow a higher speed model to be replaced with a lower speed model. The Backup contains the model specific Runtime firmware file.

Backup File Source	800 MHx Restore	1.2 GHz Single Core Restore	1.2 GHz Dual Core Restore
800 MHz Backup	Yes	Yes - If 1.2 GHz Runtime file is included in the Backup	No
1.2 GHz Single Core Backup	No	Yes	No
1.2 GHz Dual Core Backup	No	No	Yes

★ TIP

This section provides an overview of the backup and restore processes. For a deeper discussion, see EtherCAT Devices Backup and Restore in the KAS IDE help.

10.2.15.1 Backup Tab



Backup EtherCAT Devices	This function replicates the network topology as well as the AKD, AKD PDMM, AKD-C, AKD-N, and AKD2G drives' unique ID, firmware and data. To accomplish the backup, a copy of the firmware package is required. There are several possible scenarios upon clicking this button.		
	 If an archived copy of the same firmware package is on the controller as is used on the drives then clicking the button will start the backup. If a copy of the firmware package cannot be found, you will be prompted to browse for one or more. A link to the Kollmorgen website is provided; all firmware packages can be found on the site. 		
	EtherCAT Devices		
	Please provide the required firmware package(s) listed below ⁽¹⁾ :		
	AKD-Firmware-for-KAS-V01-18-00-004.tgz		
	Choose File No file chosen		
	AKD2G-Firmware-for-KAS-V02-02-00-000.tgz		
	Choose File No file chosen		
	Continue Backup EtherCAT Devices		
	<< Cancel		
	• If an archive is found on the controller but it does not match the network configuration then you will be prompted to browse for one or download one from the website.		
	For all other EtherCAT devices, such as couplers, IOs, ServoStar drives, and third- party devices, only the unique ID is backed up.		

★ TIP

Files on the website are saved in ZIP format. You must unzip the download to access the TGZ file.

TU.2.15.2 Restore fail	,		
Restore Controller	This function restores a AKD PDMM or PCMM's firmware, network configuration, retained variables and PLC application from the SD card.		
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.		
Advanced	devices. Clicking this button creates a map of the network and it's component a valid firmware file is found and the current topology matches the backup file you may select the device(s) that need to be restored. Click <i>Restore selected devices</i> will restore the drive's firmware. EtherCAT Devices		

10.2.15.2 Restore Tab

10.2.15.3 Import/Export

These functions allow you to save a copy of the EtherCAT configurations backup file to a computer and Import the backup file so it may be used for restore functions.



Export Backup	This button allows you to save a backup of the EtherCAT configuration to a computer.
Choose File	This button allows you to select a previously exported backup file to be imported.
Replace Backup	This button imports the selected backup file, replacing any existing backup.

10.2.16 Diagnostic (PDMM and PCMM Only)

This page displays information about the hardware status (storage space, memory and CPU temperature) and errors and alarms.

	KAS Application	Settings	Backup & Restore	Diagnostics	Help
Diagnostics					
		0	54		
Errors and Alarms	HW Status	Crash Report	s EtherCAT		
Storage space (used/total)		Crash Report	* refreshed autom	atically	
		Crash Report		atically	
Storage space (used/total)	_8.0 / 36.9[MB]	Crash Report		atically	
Storage space (used/total) Available memory	_8.0 / 36.9[MB] _194.8[MB]	Crash Report		atically	
Storage space (used/total) Available memory CPU usage*	_8.0 / 36.9[MB] _194.8[MB] _4%	Crash Report		atically	

10.2.16.1 Errors and Alarms

Any controller errors or alarms generated by the system will be shown here and on the 7-segment display. A common error or alarm is due to the flash memory being full. This is often caused by heavy use of the PLC Advanced File function blocks.

The **Refresh** button updates the list. The **Clear** button will remove the contents of this tab. Please note that some errors or alarms are only cleared by powering off and restarting the AKD PDMM or PCMM.

CODE	DESCRIPTION	REMEDY
E12	Not enough flash memory available.	Clean-up the flash memory by removing log files, application programs, recipes, or other data files.
A12	Flash memory is low on free space.	Clean-up the flash memory by removing log files, application programs, recipes, or other data files. Reset to factory defaults.

See Errors and Alarms for a complete list of codes.

★ TIP

Axis errors can be seen in the KAS Application *Axis* tab.

10.2.16.2 Hardware Status

Storage Space	The diagnostic displays 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 field displays the amount of RAM memory available on the Controller.
CPU usage	This field displays the current load on the CPU. For dual-core systems, this is the average of the two cores. If the load goes over 90%, the field turns red.
CPU Temp	This field displays the temperature of the CPU in Celsius. If the CPU temperature is greater then the CPU warning limit, the temperature background color will be changed to yellow. If the CPU temperature is greater than the CPU critical temperature, the temperature background color will be changed to red. The normal operating range is 0-125°C. For dual-core systems, this value is the average of the two cores.

CPU Fan Present	This field is either True or False , depending upon if there is a CPU fan present in the controller.
Refresh	Clicking this button will refresh the Hardware Status information.
Reboot	Clicking this button will reboot the web server.

() IMPORTANT

Do not try to refresh the web page until the server has rebooted.

10.2.16.3 Crash Reports

The files shown on this tab are reports of the process that failed if there is a crash. These files (GZ archives) may be sent to Kollmorgen for analysis.

Errors and Alarms	HW Status	Crash Reports	EtherCAT
Crashdump Files			
sinope_3_00_0_73351_0	.core.gz 🛓		
Clear Crashdump Ref	iresh		

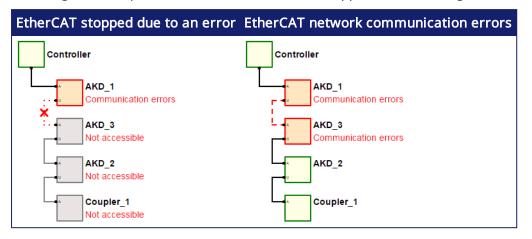
10.2.16.4 EtherCAT Diagnostics

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.
A B	Red Box	This is a node that is experiencing communication errors.
A C	Gray Box	This is a node which cannot be reached, but was known to exist.
	Black Line	This cable is working normally.
	Gray Line	This cable is not accessible.
	Red Dashed Line	This cable is experiencing degraded communication

Element	Description	Meaning
×	Red X in Dotted Line	The connection between the nodes is broken.

Following are examples of error situations when the application is running.



10.3 Using SSH

AKD PDMM and PCMM controllers can be accessed using SSH. SSH is used to perform maintenance operations that are not available through the web interface. Additionally, Kollmorgen Application Engineers may need to use SSH to access the controller to help resolve customer issues.

IMPORTANT

```
To ensure security, it is critical that the login password be changed from the default value. See Change Controller's Password Via SSH for information.
```

10.3.1 Logging In To A Controller Via SSH

An SSH client is required to log in to a controller via SSH.

1. Configure the SSH client using the following settings.

Setting	Value
Host Name or IP Address	(IP Address of the controller)
Connection Type	SSH
Port	22

- 2. Enter the user name and password at the login prompt.
 - Default User Name: root
 - Default Password: administrator

NOTE

Please note that the root password and the webserver's password are separate; one does not affect the other.

A List of Some SSH Clients

Kollmorgen does not recommend any particular program. Following are some common SSH clients, for your convenience.

- PuTTy http://www.putty.org/
- MobaXterm http://mobaxterm.mobatek.net/
- TeraTerm https://ttssh2.osdn.jp/index.html.en
- Terminals https://terminals.codeplex.com/

① IMPORTANT

To ensure security, it is critical that the login password be changed from the default value. See Change Controller's Password Via SSH for information.

10.3.2 Change Controller's Password Via SSH

Following is the method for changing the controller's root password via SSH.

- 1. Log on through SSH
- 2. At the prompt type "passwd" and press Enter.
- 3. Enter the new password at the two prompts.

Example of changing the root password.

```
# passwd
changing password for root
New password:
Retype new password:
#
```

10.3.3 How to Restore the Controller's Default Password and/or SSH Settings

There are two methods to restore the controller's default root password and SSH settings. The first method, restoring the directory via SSH, is recommended if it is possible.

• Restore the /mount/flash/etc directory.

This method should be used if logging in via SSH is possible. This will restore the /mount/flash/etc directory to the default state. Following are the shell commands to accomplish this.

```
# rm -r /mount/flash/etc
# shutdown
```

The controller will load with the default settings upon restarting.

• Reset the Controller to Factory Settings using one of two manual methods.

① IMPORTANT

Performing a factory reset of the controller should be considered very carefully as it will reset everything including the application, the IP address, any retain variables, passwords, and shared directory settings.

IMPORTANT

To ensure security, it is critical that the login password be changed from the default value. See Change Controller's Password Via SSH for information.

11 Tools

11.1 Pipe Network Editor

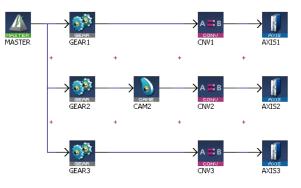


Figure 7-7: Pipe Network Structure

The Pipe Network Editor is a graphical tool dedicated to the description of the motion part of the application (Pipe Network Concept).

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: you can use it to graphically create a Pipe Network or you can decide 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.

11.1.1 Inserting Pipe Blocks or Comments

To insert Pipe Blocks or comments, right-click on a free grid unit and choose the corresponding command in the contextual menu.

11.1.2 Inserting Connections

Connections are simply inserted by clicking on an adequate1 point and dragging the mouse to another adequate point. For more details, refer to Adding Motion.

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.

¹As explained below, an adequate point depends on the type of the connection

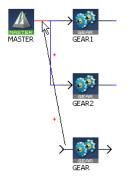


Figure 7-8: Pipe Network - Create a Link

When you try to connect two Pipe Blocks, the editor highlights the target port in red when the connection is allowed.

Relation type for output-input is **1-n**

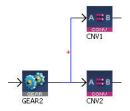


Figure 7-9: Pipe Block - Relation Type for Output-Input

One output can be connected to several inputs, but one input can only be connected to one output.

11.1.3 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.

11.1.4 Pipe Network Editor Controls

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 (**Ctrl+Z**).

Multiple Selections

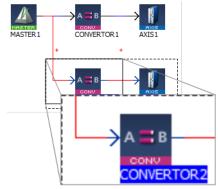
This section details selecting, moving, copying, pasting, and deleting multiple Pipe 🗙 Network blocks.

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 will be highlighted and connectors will be red.



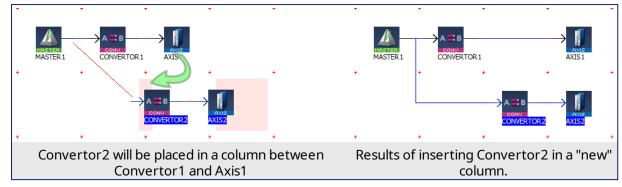
🗙 TIP

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.

5	\times	Delete Selection	Del
		Copy Selection	Ctrl+C
	X	Cut Selection	Ctrl+X
	Ê	Paste Here	Ctrl+V

Once selected, multiple Pipe Network blocks and connectors can be moved as a group. Simply click on a block and drag it to its new location. Light red rectangles show where the blocks will be placed. Narrow rectangles represent that a block will be placed between two current blocks, which will be moved to accommodate the repositioned blocks.



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 two ways.

- Right-clicking in the Pipe Network Editor and selecting Paste Here.
- Clicking in an empty location will place the blocks starting at that spot.
- Clicking on an existing block will place the copied blocks on top of the selection. You will need to move the pasted blocks to see the entire network.
- Pressing Ctrl-V then clicking in the location you want the blocks to be placed.

★ TIP Copied connectors will only be pasted when they are connected to a copied block.

Deleting multiple Pipe Network blocks.

Selected blocks may be deleted using either of two methods.

- Pressing the **Delete** key.
- By right-clicking on a selection and choosing **Delete Selection**.

11.1.5 Plug/Unplug Channels

Right-click on a Pipe Block to plug/unplug a channel of the Softscope. For more details, refer to How to Plug Motion Variables.

11.2 Cam Profile Editor

11.2.1 About the Cam Profile Editor

To open the cam profile editor in a new tab of the workspace, you have to double-click on the profile in the Project Explorer.

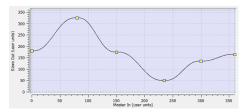


Figure 7-10: A cam profile

The cam profile editor enables you 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 Segment Overview for more information on the segment types.

Master/Input (X-Axis) and Slave/Output (Y-axis) coordinates can be specified to define the position.

In addition to the position, it is also possible to visualize the velocity, acceleration, and jerk diagrams.

11.2.1.1 Windows Overview

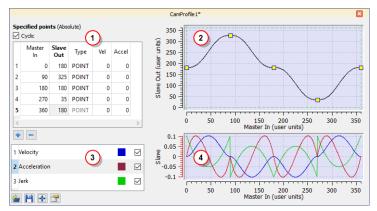


Figure 7-11: Cam Profile Editor Main Window

The cam profile editor contains four distinct parts separated by splitters:

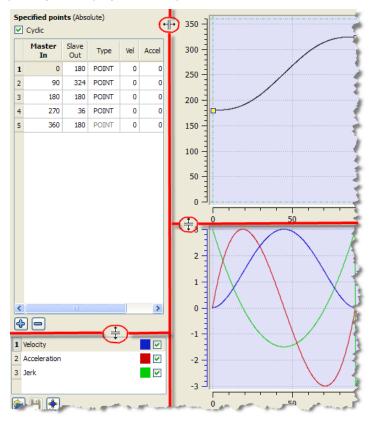
- 1. The cam table (see call out (1)) displays each element and allows editing of the cam.
- 2. The Graphical Area for the cam profile⁽²⁾ The upper graph displays a graphical representation of the cam elements
- 3. The Curve Selection and Color Table ⁽³⁾ allows you to select which plots (velocity, acceleration 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.

Splitters allow you to resize each part.

Improve your display with the 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.

For more information on cam profiles see Adding Cam Profiles and Profiles.

11.2.2 Cam Profile Editor's Cam Table

Specified points (Absolute) Cyclic					
	Master In	Slave Out	Туре	Vel	Accel
1	0	180	POINT	0	0
2	90	324	POINT	0	0
3	180	180	POINT	0	0
4	270	36	POINT	0	0
5	360	180	POINT	0	0
<					>
¢					

Figure 7-12: Cam Table

When a new profile is created, the cam profile contains five points by default.

NOTE

These points could be different from those in the figure above, depending on the offsets and amplitudes specified in the cam profile (see "Create Cam Profiles" on page 199) dialog box.

Column	Description
Master/In	The time is located in the Master/In column. It is the X-axis of the cam profile graph

Column	Description
Slave/Out	The position is located in the Slave/Out column. It is the Y-axis of the cam profile graph
Туре	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)

Table 7-6: Cam Editor - Table Parameters

About Cyclic Cam Element

If the *Cyclic* check box is selected, the cam profile is executed cyclically. This means that, 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.

Specified points (Absolute) Cyclic

Therefore, changing the **Vel** or **Accel** value of the first or last elements automatically changes the other elements' value.

NOTE

When *Cyclic* is first turned on, the Vel/Accel values will automatically be copied from the first element to the last element when they do not match. A warning dialog is displayed to inform you that this change has happened. This alert can be suppressed until KAS is closed.

There are some combinations of points and lines where *Cyclic* will automatically be turned off. If this occurs, the cyclic checkbox label will be changed to *Cyclic (automatically turned off)*. The following changes to the profile will automatically turn off cyclic:

- 1. The first element has been changed from a point to a line. If needed, cyclic can manually be turned back on which will affect the velocity of the last element.
- 2. 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 will be disabled and will only be re-enabled when the first and next to last elements are not lines.
- 3. The first element is a line and the first element is moved. If needed, cyclic can manually be turned back on which will affect the velocity of the last element.
- 4. The first element is a line and the second element is moved. If needed, cyclic can manually be turned back on which will affect the velocity of the last element.
- 5. The first element is a line and the last elements velocity (or slope line) has changed. If needed, cyclic can manually be turned back on which will change the velocity setting just made.

11.2.2.1 Modifying an Element using the Cam Table

You can modify a cam element by clicking in the **Master/Input**, **Slave/Output**, **Vel**, or **Accel** column and typing in a new value. For **Type**, refer to **Modifying the Type of a Cam Element**.

The graphs are updated automatically when an element changes.

Some 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 entered value is invalid (due to the interpolation calculation), it is superseded with the original value without any error message.

About 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.

11.2.2.2 Modifying the Type of a Cam Element

The type of element can either be a point or a line. The element type can be modified by doubleclicking in the **Type** column of an element and then clicking on the down arrow. A list of choices is displayed as shown. Select the type of element from the list.

Specified points (Absolute)								
In Out Type Vel Accel								
1	0	180	POINT	0	0			
2	90	324	POINT 🔽	0	0			
3	180	180	POINT LINE	0	0			
4	270	36	POINT	0	0			
5	360	180	POINT	0	0			
• • •								

Figure 7-13: Modifying an Element Type

11.2.2.3 Cam Table Contextual Menu

Right-clicking on an entry in the cam table displays a contextual menu.

Specified points (Absolute) ✓ Cyclic						
	In	Out	Туре	Vel	Accel	
1	0	180	POINT	0	0	
2	Ó	🔋 Сору		Ctrl	+C p	
3	1	Paste		Ctrl	+V D	
4	2 4	lnsert	New Point	Ctrl	+I D	
5	3 =	Remo	ve Row	Ctrl	+R P	
<						
\$ -						

Figure 7-14: Cam Table Contextual Menu

Command	Shorcut	Description
Сору	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 Adding a Point to the Cam Table
Remove Row	Ctrl+R	Deletes the row that contains the highlighted entry. This command is described in Remove a Point from the Cam Table

11.2.2.4 Adding a Point to the Cam Table

You can add a point to the cam table using one of the following methods:

- Use the menu in the cam table (shown in Cam Table Contextual Menu)
- Click the button located below the cam table
 Use the menu in the cam profile graph

All of these methods displays the Add New Point dialog box:

📕 Add New Po	🗖 Add New Point to CAM Table 🛛 🕐 🔀					
Enter New P	pint Information					
Master In:						
Slave Out:						
Element Type:	POINT	~				
Velocity:						
Acceleration:						
	OK	Cancel				

Figure 7-15: Add New Point

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)
Acceleration	The acceleration of the new point (second derivative)
ОК	Accept the entry and verify if the point can be added.
Cancel	Cancel the dialog box – no point is added.

Table 7-7: Cam Editor - New Point Parameters

When you click OK, a check is performed to see if the point can be added to the cam profile. If not, an error dialog box is displayed.

CAM Editor		
	Error: Could not add point to CAM table.	
	Values may be too large or the new point conflicts with an existing point. Check the entered values with the amplitudes and offsets of the CAM table and any existing points.	
	ОК	

Figure 7-16: 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.

11.2.2.5 Remove a Point from the Cam Table

You can remove a point from the cam table with one of the following methods:

- Use the menu in the cam table (shown in Cam Table Contextual Menu)
- Click the button located below the cam table
 Use the menu in the cam profile graph

The selected point is removed without prompting.

NOTE

The first and last points cannot be removed.

11.2.3 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.

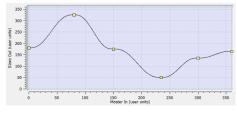


Figure 7-17: Cam Profile Graph

Points and endpoints of lines are displayed as yellow squares () in the graph. The profile offset and amplitude specified in the cam profile editor (see "Create Cam Profiles" on page 199) are displayed with a green dashed rectangle. The yellow squares are always contained within the green dashed rectangle (although calculated points can extend outside it).

11.2.3.1 Modifying an Element

You can modify the profile by moving point with the mouse as follows:

- 1. Move the mouse over a yellow square (the cursor becomes ${igoplus}$ indicating that the point can be selected)
- 2. Click to select the point and hold down the mouse button (left-click). When you move the mouse, the point follows the cursor (note that graphical curves and In/Out values are dynamically updated)

In addition, when a point is selected, a slope line is drawn over the point. This line is dashed purple with two additional grips (\diamond) attached to it. The slope line can be used to change the velocity of the selected point.

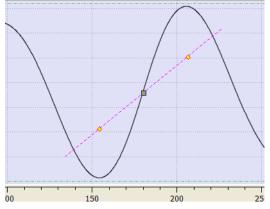


Figure 7-18: Cam Profile Graph - Slope Line

You can change the velocity of the selected point as follows:

- 1. Move the mouse over a slope grip (\diamond). The cursor changes to an open hand $\langle n \rangle$
- 2. Click to select the grip and hold down the mouse. The cursor changes to a closed hand \heartsuit
- 3. When you move the mouse, the slope line follows the cursor, rotating about the selected point and causing the velocity of the selected point to change. (Note that graphical curves and Vel value are dynamically updated)

11.2.3.2 Cam Profile Graph Contextual Menu

A right-click on the cam profile graph displays a contextual menu.



Figure 7-19: Cam Profile Graph - Contextual Menu

Command	Description
Insert Point	Inserts a new point at the X-Y location of the cursor
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

11.2.3.3 Zoom In and Out

In the cam profile graph, you can zoom in or out as follows:

- 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.

11.2.3.4 Panning

In the cam profile graph, you can also pan (or move) in any direction as follows:

- 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

11.2.3.5 Restoring Zoom and Pan

To restore the zoom and pan settings, so the entire curve is displayed in the graphical area, click on

the Auto Fit button 🗺 or select the Auto Fit command in the cam profile graph menu.

11.2.4 Cam Profile Segment Overview

Point Segment Type

Supported by		KAS IDE Profile Editor, MLProfileBuild (default option)
Continuous velocity		\checkmark
Continuous acceleration	n	\checkmark
Interpolation method		5 th 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. For example, sinusoidal profiles can be emulated with 6 to 12 point segments.
Disadvantages		One must 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.
Segment Curve	Velocity	Acceleration Jerk

Line Segment Type

Supported by	KAS IDE Profile Editor, MLProfileBuild
Continuous velocity	
Continuous acceleration	X
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.
Segment Curve Velocity	Acceleration Jerk

Parabolic Segment Type

Supported by		KAS IDE Profile Editor, Noption)	MLProfileBuild (parabolic
Continuous velocity		1	
Continuous acceleratio	n	\checkmark	
Interpolation method		Linear function: f(x) = A and a 2 nd order polyno	x + B mial: f(x) = Cx ² + Dx + E
Advantages		This segment type is use acceleration portions of the peak acceleration n cam point to another. T the motors cannot supp used by other segment	f a profile. This minimizes eeded to move from one his can be useful when port the accelerations
Disadvantages		Acceleration is discontir additional electrical stre motors.	
Segment Curve	Velocity	Acceleration	Jerk
			1 1

In the example shown, the blue line represents the linear (constant velocity) part of the segment, while the black lines represent the parabolic (constant acceleration) parts of the segment.

Spline Segment Type

Supported by	KAS IDE Profile Import. The points are created by separate software and is imported into KAS IDE project.
Continuous velocity	\checkmark
Continuous acceleration	\checkmark
Interpolation method	3 rd 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.
Segment Curve Velocity	Acceleration Jerk

11.2.5 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 shown in figure below, you can select or clear each individual curve to be displayed.

When a curve is selected (see blue highlighted row in figure below), the Y-scale of the Curves graph is adjusted to display the Y-scale of the selected curve. Also, the color of the 'tick' line of the scale is changed to match the color code of the selected curve.

1	Velocity	
2	Acceleration	<
3	Jerk	<

Figure 7-20: Curve Selection Table

11.2.5.1 How to change color

You can change the color of a plot as follows:

1. Double-click on a colored square shown in the Curve Selection Table to open the color selection dialog box

Select color	X
Basic colors Image: Colors	
<u>C</u> ustom colors	Hu <u>e</u> : 240 🔷 <u>R</u> ed: 0 🗘
	<u>S</u> at: 255 ♀ <u>G</u> reen: 0 ♀
	Val: 204 🗢 Blue: 204 🗢
Define Custom Colors >>	Alpha channel: 255 📚
OK Cancel	Add to Custom Colors

Figure 7-21: Standard Color Selection

2. Click on an existing color square to select it, or specify the numerical values for a color. (You can also move the black indicator on the right side until the desired color appears in the large colored rectangle)

11.2.6 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.

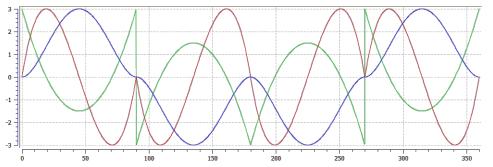


Figure 7-22: Curves Graph

With the check boxes in the Curve selection table shown in Curve Selection Table, 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

In general 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.

11.2.7 Reload, Save, Auto Fit, and Properties Buttons

The following buttons are provided:

Icon	Description
Reload	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	Save a modified profile.
Autofit	Adjusts the zoom and pan settings so that the entire graph is displayed in the graphical area.
Properties	Open the Cam Profile Properties dialog box to modify the Master/Input and Slave/Output Offset and Scale values

Table 7-8: Cam Editor - List of Icons

11.2.8 Import Cam Profile

The KAS IDE can import cam profiles from several sources.

- Cam profiles (.CAM file format) which were exported from another KAS project.
- User-defined cam profiles (in .CSV format) as described below. Upon importing, 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
Ν	1000;1000;SPLINE
N+1	TABLE_END;;

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 Segment Overview** 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;
0.3333333333333; 0.333333333333333;
0.6666666666666667; 0.666666666666667;
1;1;
```

Offset:

10 20

Scale:

526

300 360

Value displayed in profile:

```
10;20;
109.999999999999; 139.99999999999999;
210.000000000009; 260.000000000011;
310;380;
```

11.2.8.1 About the Import

A quick validation is performed when the profile is first imported.

- The profile name is checked and if it is in use you are promoted to provide a new name.
- The data format is validated and we check to see if the profile 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 will be copied to the last elements. A warning message will be posted to the log if this change takes place.

11.2.8.2 When Displaying the Imported Cam Profile

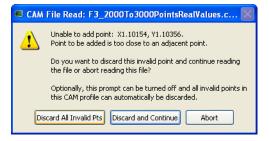
When you display an imported profile a dialog box indicates the progression of the import process.

🛎 CAM Editor progress 👘 🛛 🔀
Populating Cam Profile
43%
Abort

Click the **Abort** button to abort the process, then a default cam profile is created.

11.2.8.3 About 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 is displayed when the process is finished.



11.3 Soft Oscilloscope

The soft oscilloscope (commonly known as "softscope" or "scope") is a tool which allows you to view, in a two-dimensional graph, one or more variables' evolution (vertical axis) across the time (horizontal axis).

The scope (as seen in Figure 7-23: Example of using the Oscilloscope) has a set of channels where each can acquire the evolution of a value. 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. For details on how to assign a variable to a channel, see "Plugging Probes, Mapping Variables" on page 534.



Figure 7-23: Example of using the Oscilloscope

The soft oscilloscope is a child window in the workspace which has two visually distinct parts:

- The Control Panel¹ enables you to assign variables to a channel and to change the settings of the soft oscilloscope (including those of the channels)
- The Graphical Area⁽²⁾ shows the traces (or variable values) acquired by the channels
- The control panel and the graph are separated by a splitter which allows you to change the size
 of the two panes.

★ 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.

11.3.1 Access the Soft Oscilloscope

In order to access the softscope, select the **Oscilloscope** command from the **Tools** menu.

Kollmorgen Automation Suite Integrated Development Environm				
<u>■, F</u> ile <u>E</u> dit <u>V</u> iew	<u>T</u> ools	<u>W</u> indow <u>H</u> elp		
: 🖪 🍙 💾 🚔 🕵	Custom <u>I</u> O Editor			
Project Explorer	<u>O</u> scilloscope		×	
Project View	<u>C</u> ompile		Ctrl+B	
Name	<u>C</u>	ompile And Download	Ctrl+F7	
🗸 📷 System	U	<u>s</u> er Options		
Controller [10.50,68-50] PDMM				

Figure 7-24: Accessing the Scope

11.3.2 About OpenGL

For the Graphical Area, the scope uses **OpenGL** for performance reasons. If you encounter problems in the quality of drawings, we suggest that you consider the following points before contacting our support desk:

Check that your graphical card driver is up-to-date.

Newer drivers often fix the rendering bugs of OpenGL.

Disable some optimizations on the Display hardware acceleration

Open your operating system's **Display** Properties1. In the **Settings** tab, click the **Advanced** button, then select the **Troubleshoot** tab. If **Hardware acceleration** is set to full, try to disable some optimizations. This procedure has proven to be useful in particular with cursor drawing problems that appear when the user performs high-zooming operations (the cursor can indicate a value which is out of the trace).

Change the settings of your graphic card

Open the manufacturer-specific settings of your graphic card. 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.

Display a given amount of samples, according to 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)
- 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.

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.

11.3.3 The Control Panel

As shown in Figure 7-25: Oscilloscope Control Panel, the control panel consists of the following items:

¹The Properties command is accessible in the contextual menu on your desktop (you can also access the Display from the Windows Control Panel)

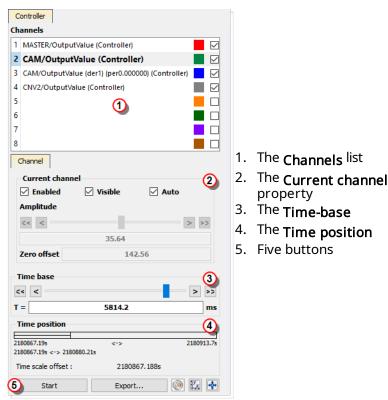


Figure 7-25: Oscilloscope Control Panel

1. The Channels list

This lists the eight available channels, and the variables assigned to each.

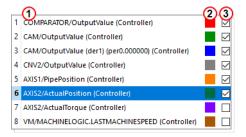


Figure 7-26: Scope Control Panel - Channels

For each channel, it displays:

- 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 lets you quickly show/hide each captured trace.

★ TIP

You can change the color of a curve by double-clicking on its color icon, and its visibility by clicking on its check box.

★ TIP

Right-click on a channel to map, unmap, or edit the color of a variable's trace.

When selecting a channel in the channels list, it is superimposed on the existing traces, and some related information are displayed on the left and lower sides of the graph.

2. The Current Channel

This section details the properties of the currently selected channel.

Current chann	el
🗹 Enabled 🔽	Visible 📃 Auto
Amplitude	
< < ·	-] > •>
2	00
Zero offset	0

Properties	Description
Enabled	A channel has to be enabled to acquire the samples sent by its associated probe
Visible	A channel has to be visible to be drawn on the graph. Even if not visible, it continues to acquire the samples sent by its associated probe
Auto	A channel in auto mode automatically adapts its amplitude (unit/division ¹) and zero offset in order to be able to display all its samples. Setting the auto mode disables the possibility of changing the Amplitude and the Zero offset (see Setting Scale for more details about scaling)
Amplitude	Allows you to control 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

Table 7-9: Scope - Current Channel Properties

NOTE

On some special devices, some more tabs that are specific to extra configurations appear in this widget. For example, S300 device provides trigger functionality, so an additional tab is displayed for the trigger configuration.

3. The time-base

This enables you to set the speed at which all the lines for each channel are drawn, and is calibrated in milliseconds per division.

Its usage is similar to the Amplitude property described in the above section. The time-base can always be changed, even during sampling (see also Time Scale).

Time ba	ise	
< < ·		- > •>
T =	429.982	ms

Figure 7-27: Scope Control Panel - Time-base

¹The term refers to the time-base value for the X-axis and to the amplitude value for the Y-axis. For example, if the user sets a time-base of 10ms and an amplitude of 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.

To setup the time-base properly, the total measurement duration and the required time resolution have to be taken in account.

4. The time position

This enables you to change the time-frame of the acquired samples shown on the graph. It is composed of:

- A single horizontal line representing all the acquired samples with start and stop timings
- A **rectangle** representing only the time slot of the acquired samples, which is displayed in the graphical area (the time-frame) with timings:

Time position		
	200s <-> 204s	
132s	<->	204s
Time scale offset :	0.000s	

Figure 7-28: Scope Control Panel - Time Position

NOTE

The acquisition of samples is limited to 100'000 cycles (ie. 100 s when cycle time is set to 1000 μ s, and 25 s when cycle time is set to 250 μ s). When you reach this limit:

- The first data that are added to the queue are the first data to be removed (FIFO queue)
- The start timing increases

You can change the time slot with the mouse by:

• moving the rectangle

	- Time position	
•	changing the size	
	Time position 102 101995c → 110 457554	

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

How to 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.

You can resize the time-frame in a user-friendly manner by clicking on its left or right ends and dragging.

NOTE

During acquisition the time position item is disabled and displays the progression of acquisition.

5. The five buttons

At the bottom of the controls are five buttons:

- 1. The **Start/Stop** button start allows you to start or stop the acquisition of samples. When starting acquisition, all previous samples are lost.
- 2. The **Export...** button Export... allows you to save the acquisition data in a CSV file. For more details, The Graphical Area.
- 3. The **Plug Trace Times Channels** button 🔘 allows you to quickly load and display the four following channels

• Channel 1: Cycle Jitter (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 μ s). The

CycleJitter is due to EtherCAT transmissions that can vary in a particular cycle (see call out 1). The channel 1 of the scope monitors the time difference between the expected Cycle Time and the actual Cycle Time. (see figure below).

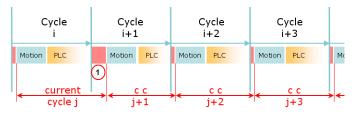


Figure 7-29: Cycle Time Calculation

- Channel 2: Motion execution time (microseconds)
- Channel 3: PLC execution time (microseconds)
- Channel 4: Real Time Margin (microseconds) This channel monitors the available execution time (Cycle Time Period x Controller Core Count EtherCAT network execution time MotionExecTime PLCProgExecTime). The measurement is based on CPU usage, which provides the most accurate measurement. The measurement is updated each cycle period.

Channel 2	Channel 3	Channel 4
Motion	PLC	Idle

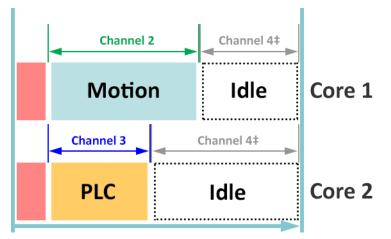


Figure 7-30: Motion, PLC and Real Time Margin Time Calculations

‡ Channel 4 is the total amount of time across all CPU cores.

Figure 7-31: Multiple Core Controller, PLC and Real Time Margin Time Calculations.

NOTE

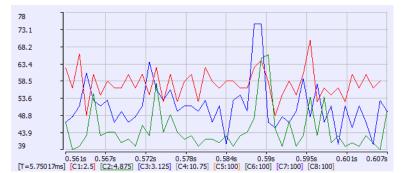
This feature is **not** relevant with the KAS Simulator. The MotionExecTime and PLCProgExecTime traces will be visible with simulated values. The CycleJitter and RealTimeMargin will always remain at zero with the simulator.

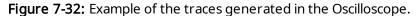
For more explanations, refer to Tasking Model / Scheduling

- 4. The **graduations** button 📖 displays or removes the axis graduations of the graphical area. By graduations are visible by default.
- 5. The **autofit** button the state time-frame of the graph and amplitudes and zero offsets of channels so that they all fit entirely into the graphical area.

11.3.4 The Graphical Area

The graph displays a subset of the collected data: the time-frame.





To better view and analyze the data, the graph has the following features:

- 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 (for more details, see Trace Zoom Feature)
- It is possible to **move** the contents of the graph within the time-base (for more details, Trace Zoom Feature)

NOTE

Moving the contents is possible only when the acquisition is stopped.

See How Do I Export the Oscilloscope Data? for information on capturing the trace data.

11.3.5 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

534

The acquisition of samples is limited to 100'000 cycles (ie. 100 s when cycle time is set to 1000 μ s, and 25 s when cycle time is set to 250 μ s). When you reach this limit:

- The first data that are added to the queue are the first data to be removed (FIFO queue)
- The start timing increases

11.3.6 Plugging Probes, Mapping Variables

A probe is a virtual measurement point that can be connected to a variable.

Three types of variables can be mapped:

- 1. Pipe Block variable
- 2. IEC 61131-3 variable, which is any other variable.
- 3. 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 the following ways:

- from the Softscope, see Mapping a Variable to the Oscilloscope
- from the Dictionary, see Plugging a probe from the Dictionary
- from the Pipe Network, Plugging a probe from the Pipe Network

11.3.6.1 Mapping 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 and select the **Map** option to open the **Scope Variable Selector**.

N	OIE	
If you	r application is not connected and running,	the following message is displayed.
🔳 Scope	Channel Selection	
1	Map/Unmap operations can only be performed when the program is running	
	ОК	

2. Navigate through the available variables and select the one you want to connect to the channel

Scope Variable Selector	\times
Select a variable to plug on the channel	
Project Variables	^
✓ ■ Controller	
> 👩 Motion	
V 🛐 PLC	
🗸 📭 (Global)	
ActiveTestPtIndex	
Axis1Status	
Axis2Status	
> EtherCAT	
MachineSpeed MachineState	_
MasterAbsPos	
MasterDeltaPos	
TempLReal	
> TestProfPoints	
TravelSpeed	
bEStop	
> bLedStatus	
hh da stand ha	~
Derivation Order: 0	•
Derivation Period: 0.0	
Ok Cancel	

Figure 7-33: Using the Variable Selector to choose a variable to be mapped in the Oscilloscope

NOTE

The Scope Variable Selector contains only the PLC variables that are eligible for the softscope (i.e. BOOL, INT, SINT, DINT, LINT, UINT, USINT, UDINT, ULINT, BYTE, WORD, DWORD, LWORD, TIME and LREAL, as long as they are not in a UDFB instance).

Additionally, only a subset of variables are displayed (e.g. ActualVelocity 1 is not visible), when in Simulated mode.

For more details on:

- Axis pipe block positions, About Associated Data on Positions
- PLCopen Axis positions, Axis Positions Data
- 3. (Optional) Set the **Derivation Order**.
- 4. (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.

You can also disconnect a probe as follows:

Unmap a variable(s) from the Scope

In order to unmap a variable, or unplug a probe:

- Right-click on the channel you wish to remove from the channels list and select the **Unmap** command in the menu to disconnect the probes.
- Select the **Unmap all Oscilloscope Channels** command from the right-mouse menu in the Dictionary or the PipeNetwork.

11.3.6.2 Plugging a probe from the Dictionary

A probe (variable) can be directly plugged or mapped to the Oscilloscope from the Dictionary:

- 1. In the **Dictionary**, right-click on the variable
- 2. In the menu, select the Map to Oscilloscope channel ... command

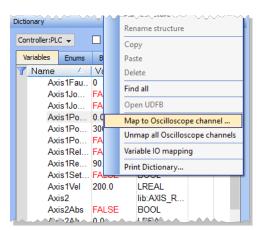


Figure 7-34: Plugging a Probe from the Dictionary

NOTE

536

This command is enabled if the type of variable is eligible for the softscope (i.e. BOOL, INT, SINT, DINT, LINT, UINT, USINT, UDINT, ULINT, BYTE, WORD, DWORD, LWORD, TIME and LREAL, as long as they are not in a UDFB instance).

¹The measured value is the instant velocity of the axis in RPM*1000. Note that you can see some oscillations because it is an instant velocity, not an average velocity.

3. Define the probe parameters.

There is a slight difference depending upon whether you selected a PipeNetwork probe, or a PLC probe. With PipeNetwork variables you will need to select which Data point is being used, while PLC variables may require you to navigate within the group to find the correct item.

		Map to Oscilloscope c	hannel	?	×
		Select a variable/member to	plug on the ch	annel	
		Group : (Global)			
		✓ EtherCAT			
		Coupler_3 AKD_1			
		AKD_2			
		AKD_3 Coupler_1			
		couplei_1			
🔳 Map to Oscillosco	pe channel ? $ imes$				
Name:	AXIS2				_
Type:	AXIS	Channel:	1		•
Channel:	1 👻	Derivation Order:	0		-
Data:	ActualPosition 👻				
Derivation order:	0 👻	Derivation Period: 0.0			
Derivation Period:	0.0				_
Ok	Cancel	Ok	Car	ncel	

Figure 7-35: Associating a PipeNetwork (left) and PLCopen (right) variable to a Channel

Field	Description	
Name	Variable's name	
Туре	Variable's type	
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, 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 (eq. 360.0).	
NOTE		
In order to anable the	Plug on channel dialog box the KASIDE must be connected	

In order to enable the **Plug on channel...** dialog box, the KAS IDE must be connected to the device first!

11.3.6.3 Plugging a probe from the Pipe Network

A probe (variable) can be directly plugged or mapped to the Oscilloscope from the Pipe Network:

1. Right-click on a Pipe Block

2. Select **Plug on channel...** in the menu

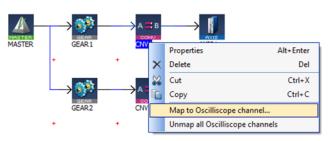


Figure 7-36: Plugging a Probe from the Pipe Network

3. Define the probe parameters.

🔳 Map to Oscillosco	pe channel ? $ imes$
Name:	AXIS2
Type:	AXIS
Channel:	1 🗸
Data:	ActualPosition 👻
Derivation order:	0 🗸
Derivation Period:	0.0
Ok	Cancel

Field	Description		
Name	Variable's name		
Туре	Variable's type		
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, 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 (eg. 360.0).		
NOTE			
In order to anable the Diug on channel dialog boy the KAS IDE must be connected			

In order to enable the **Plug on channel...** dialog box, the KAS IDE must be connected to the device first!

11.3.7 Setting Scale

The soft oscilloscope graph is divided into 8 units for the horizontal time scale (X-axis) and 8 units for the amplitude (Y-axis). These units can be user-defined by using the configuration panels described below.

NOTE

Unit per ulision: the term refers to the time-base value for the X-axis and to the amplitude value for the Y-axis. For example, if the user sets a time-base of 10ms and an amplitude of 1, each ulision in the soft oscilloscope grid corresponds to a time of 10ms for the X-axis and an amplitude of 1 for the Y-axis.

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 while 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

You can also modify the time scale by scrolling the mouse wheel with the cursor located in the graphical area.

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 Current channel control panel.
- The Edit all channels dialog.

★ TIP

You can also 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.

11.3.8 Trace Zoom Feature

The zoom feature is used to magnify or reduce a portion of a trace. Two zoom modes are available:

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.
Amplitude zoom	Used to have a better view of a part of a signal. This zoom operation updates the amplitude & zero offset value

The zoom operations can be done:

- By modifying the corresponding values by hand
- By using the mouse wheel

For more details on setting the amplitude, zero offset and time-base values, refer to Setting Scale.

Mouse Shortcuts

Action	Result
Scrolling up the mouse wheel	Expands the time-base value
Scrolling down the mouse wheel	Collapses the time-base value
Pressing the Ctrl key while scrolling up	Makes the amplitude value greater
Pressing the Ctrl key while scrolling down	Makes the amplitude value smaller

NOTE

When performing an amplitude zoom, the zero offset is automatically set by the cursor position.

11.3.9 How Do I Export the Oscilloscope Data?

To copy the trace data into a CSV file:

- 1. Display the softscope
- 2. Ensure the channels you want to export are Enabled and Visible
- 3. Start the data collection
- 4. Wait for the probe data you want to save to be collected
- 5. **Stop** the data collection
- 6. Click the **Export...** button
- 7. Select where you want to save the CSV file
- 8. Click the Save button

A tip about waiting for all channels to be recording before capturing Scope data.

Channels start recording a few cycles after the previous channel, based on resources. To see all channels at same time you need to wait until data is being sent from all channels. If you start the capture at the first cycle the exported data will not line up. The first table below shows aligned data of the channels from starting the program. Note how it takes several cycles for the data to be captured. The second table shows hohis data would appear in a CSV file. Note how the time columns do not match. Waiting for all channels to be recording will eliminate this.

Time	Channel 1	Time	Channel 2	Time	Channel 3	Time	Channel 4
0	499.9996						
1	499.9996						
2	499.9996						
3	499.9996						
4	499.9996	4	-31.543				
5	499.9996	5	-31.843				
6	499.9996	6	-32.1411				
7	499.9996	7	-32.4428				
8	499.9996	8	-32.7429				
9	499.9996	9	-33.0429	9	-534.543		
10	499.9996	10	-33.343	10	-531.843		
11	499.9996	11	-33.6431	11	-535.143		
12	499.9996	12	-33.9428	12	-535.443		
13	499.9996	13	-34.2428	13	-535.742		
14	499.9996	14	-34.5429	14	-536.043		
15	499.9996	15	-34.843	15	-536.343	15	-459.028
16	499.9996	16	-35.143	16	-536.643	16	-458.728
17	499.9996	17	-35.4431	17	-536.943	17	-458.428
18	499.9996	18	-35.7428	18	-537.242	18	-458.128
19	499.9996	19	-36.0429	19	-537.542	19	-457.829

Table 1: Scope output is aligned to show actual results

540

Time	Channel 1	Time	Channel 2	Time	Channel 3	Time	Channel 4
0	499.9996	4	-31.543	9	-534.543	15	-459.028
1	499.9996	5	-31.843	10	-531.843	16	-458.728
2	499.9996	6	-32.1411	11	-535.143	17	-458.428
3	499.9996	7	-32.4428	12	-535.443	18	-458.128
4	499.9996	8	-32.7429	13	-535.742	19	-457.829
5	499.9996	9	-33.0429	14	-536.043	20	-459.028
6	499.9996	10	-33.343	15	-536.343	21	-458.728
7	499.9996	11	-33.6431	16	-536.643	22	-458.428
8	499.9996	12	-33.9428	17	-536.943	23	-458.128
9	499.9996	13	-34.2428	18	-537.242	24	-457.829
10	499.9996	14	-34.5429	19	-537.542	25	-459.028
11	499.9996	15	-34.843	20	-535.742	26	-458.728
12	499.9996	16	-35.143	21	-536.043	27	-458.428
13	499.9996	17	-35.4431	22	-536.343	28	-458.128
14	499.9996	18	-35.7428	23	-536.643	29	-457.829
15	499.9996	19	-36.0429	24	-536.943	30	-459.028
16	499.9996	20	-35.143	25	-537.242	31	-458.728
17	499.9996	21	-35.4431	26	-537.542	32	-458.428
18	499.9996	22	-35.7428	27	-536.943	33	-458.128
19	499.9996	23	-36.0429	28	-537.242	34	-457.829

Table 2: Actual output to CSV file for Scope data captured from the start

You can now import the data into Microsoft Excel.

NOTE

The Export operation is possible even when acquisition of samples is in progress. But in that case, the latest exported data are the data collected when you have defined the CSV file.

NOTE

The acquisition of samples is limited to 100 s when the cycle time is set to 1000 μ s (respectively 50 s with 500 μ s, and 25 s with 250 μ s)

About the CSV file format

Each channel takes 2 columns: one for the **time** and the other for the **value**. This allows exporting channels with different time-base.

The **List separator** and the **Decimal symbol** are hard-coded (they are not bind to the regional settings)

- List separator is comma (,)
- Decimal symbol is dot (.)

★ TIP

If your regional settings are different, then you have to specify explicitly those two characters in Microsoft Excel to correctly import the CSV file

11.3.10 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. If the **CPU usage** is less than 90% then the CPU load (both Real Time and Non-Real Time) is okay. If the **CPU usage** is 90% or higher then the CPU is too heavily loaded and should be reduced by simplifying the application or reducing the CycleTime update rate.

NOTE

This page assumes that a single-core controller is being used. The technique to measure the CPU load with dual-cores is the same, but the load limits change. Please refer to The Control Panel for more information about the trace times.

HW Status	Errors and Alarms
Storage Space (used/total)	_7.0/36.9[MB]
Available Memory	_204.7[MB]
CPU usage*	97%
CPU Temp*	_51.9[C]
CPU Fan Present	_False
	Refresh Reboot

The IDE Oscilloscope trace times can be used to analyze the application performance on a controller or programmable drive. This section describes some techniques you can use to interpret the trace times to examine the real-time performance.

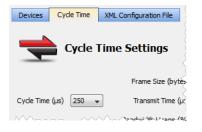
There are two major parts to consider when evaluating total performance:

Real TimeEtherCAT + 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, but the best indicator is the overall **CPU usage** and the Controller Log messages.

First, you will want to know the Cycle Time for your system. From the **Project View**, select the **EtherCAT** view and the **EtherCAT Master Settings Tab** tab. The update period for the system in this example is set to 250 microseconds.



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, as seen here.

11.3.10.1 Collect some data by pressing the "Start" button

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. Below are some recommended values based on several Cycle Time values.

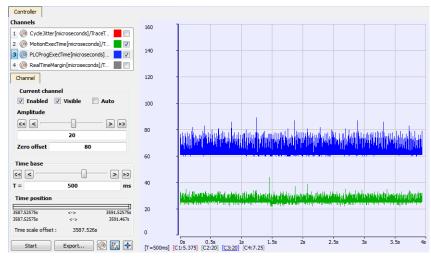
Controller
Channels
1 CycleJitter[microsecon 📕 🗹 ^
2 MotionExecTime[microsecon
3 PLCProgExecTime[microsecon
4 RealTimeMargin[microsecon 🔳 🗌 🗸
Channel
Current channel
🗹 Enabled 🗹 Visible 🗹 Auto
Amplitude
<< <
44.9988
Zero offset 180.005
Time base
< < >>>
T = 6863.64 ms
Time position
2776181.61s <-> 2776236.52s
2776181.61s <-> 2776229.34s
Time scale offset : 2776181.608s
Start Export 🝥 🔛 🔄

Cycle Time	Amplitude	Zero Offset
250ms	20	80
500ms	40	160
1000ms	80	320

🗙 TIP

Unchecking the "CycleJitter" and "RealTimeMargin" traces is useful so they don't clutter the view.

The following example has a Cycle Time of 250 microseconds. The "MotionExecTime" average is about 27 microseconds and the "PLCProgExecTime" average is about 68 microseconds.



The average time for the MotionExecTime + PLCProgExecTime is 95 (27 + 68 = 95), which is about 38% of the cycle (95 / 250). This is a good value.

NOTE

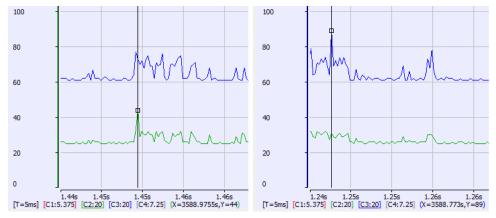
For multi-core controllers, we want to investigate the percentage of cycle time for each MotionExecTime and PLCProgExecTime. In the above example, MotionExecTime is about 11% of the cycle, and PLCProgExecTime is about 27% of the cycle. These are both good values.

11.3.10.2 Check the peak times

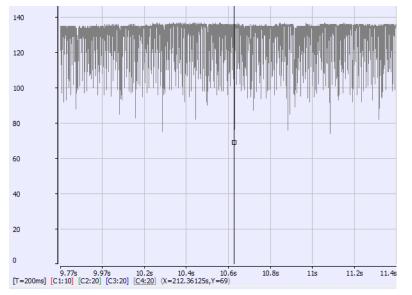
The next step is to examine the spikes. We will examine the "MotionExecTime", "PLCProgExecTime", "RealTimeMargin" and "CycleJitter" traces.

- 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 and the "PLCProgExecTime" peak is 89. This is reasonable.



For the "RealTimeMargin" peaks configure the **Amplitude** and **Zero offset** so you can see the trace near zero. In this example the minimum peak (closest to zero) is 69 microseconds. This provides a 28% (69 / 250) Real Time margin which is good.

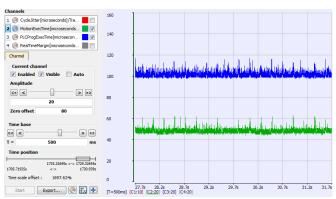


For the "CycleJitter" trace configure the **Amplitude** and **Zero offset** so you can see the trace *centered* at zero. This trace is not too interesting unless a system is misbehaving. A jitter of +/-15 microseconds is acceptable.

20							
10	nd and an attending and			a dada			
0		A TA A A A A A A A A A A A A A A A A A	a (na sa	F (5 1993) FILE FILE Radi _{da} bi di Landarda	alayii (joojii) (jooji) waxaa ka ka ka ka ka ka	, part a parte da ante da ante Ante da ante da	
-10			· "	•	·		1 '
-20							

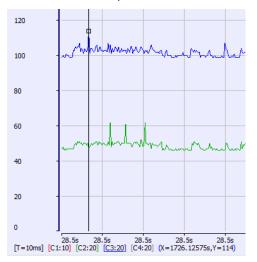
11.3.10.3 Heavily Loaded CPU Example

Here is an example of an application that is heavily loading a AKD PDMM or PCMM with the EtherCAT Cycle Time = 250 microseconds. Using the techniques described in Practical Application: Using Trace Time To Measure CPU Load, examine the "MotionExec" and "PLCProgExec" times first:



The average MotionExec + PLCProgExec = 50 + 105 = 155 microseconds. This is about 62% (155 / 250) of the cycle time.

Take a look at the peaks:



This shows the MotionExec at 62 microsec and the PLCProgExec at 114; there is not much time left over.

Check the "RealTimeMargin":

80	
60	lan an ann an an ann an an ann an an ann an a
40	
20	
0	
[T=500ms]	28.5s 29s 29.5s 30s 30.5s 31s 31.5s 32s 32.5s] [C1:10] [C2:20] [C3:20] [C4:20]

Notice the minimum time is 20 microseconds or 8% Real-Time margin (20 / 250). This is not a comfortable margin for deterministic Real-Time performance.

Checking the Controller log we see that the Virtual Machine (PLCProgExec) is missing a cycle occasionally:

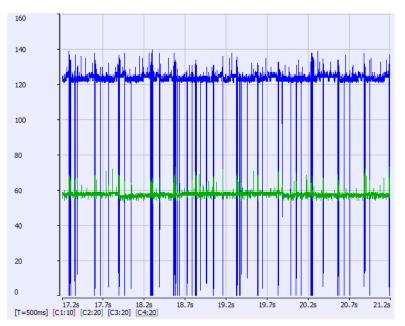
753		4/10/2012 10:37:26 AM (083)	Motion	WARNING	The Virtual Machine missed 1 cycle(s) of PLC execution.
752	▲	4/10/2012 10:37:25 AM (583)	Motion	WARNING	The Virtual Machine missed 1 cycle(s) of PLC execution.
751	Δ	4/10/2012 10:37:25 AM (083)	Motion	WARNING	The Virtual Machine missed 1 cycle(s) of PLC execution.
750	Δ	4/10/2012 10:37:24 AM (583)	Motion	WARNING	The Virtual Machine missed 1 cycle(s) of PLC execution.
749	▲	4/10/2012 10:37:24 AM (154)	Motion	WARNING	The Virtual Machine missed 1 cycle(s) of PLC execution.
748	▲	4/10/2012 10:37:23 AM (654)	Motion	WARNING	The Virtual Machine missed 1 cycle(s) of PLC execution.
747	▲	4/10/2012 10:37:23 AM (154)	Motion	WARNING	The Virtual Machine missed 1 cycle(s) of PLC execution.
746	▲	4/10/2012 10:37:22 AM (654)	Motion	WARNING	The Virtual Machine missed 1 cycle(s) of PLC execution.
745	▲	4/10/2012 10:37:22 AM (154)	Motion	WARNING	The Virtual Machine missed 1 cycle(s) of PLC execution.
/***	4	T/ 10/ 2012 10:07:21 MPI (00T)	mouon	WARLING	The virtual Plachine misseu i cycle(s) of FLC execution.

Lastly, take a look at the overall CPU load. At 88% usage there's not much CPU bandwidth available.

HW Status	Errors and Alarms
Storage Space (used/total) _7.0/36.9[MB]
Available Memory	_203.5[MB]
CPU usage*	_88%
CPU Temp*	_50.0[C]
CPU Fan Present	_False
	Refresh Reboot

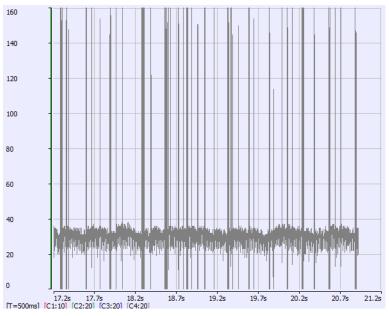
11.3.10.4 Over Loaded CPU Example

Now, let's take a look at an example of an application that is overloading a AKD PDMM or PCMM with the EtherCAT Cycle Time = 250 microseconds. Using the techniques described above, examine the "MotionExec" and "PLCProgExec" times first:



The average MotionExec and PLCProgExec times are 57 + 125 = 182 or 73% (182 / 250) of the Cycle Time. Notice the big spikes on the PLCProgExec?

Next, look at the "RealTimeMargin":



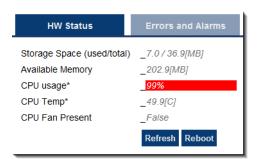
There are many cycles with zero real-time margin. Notice the big spikes? This is a degraded case.

The Controller log confirms the missing VM cycles and an A23 alarm:

992	4/10/2012 10:52:26 AM (876)	Motion	WARNING	The Virtual Machine missed 68 cycle(s) of PLC execution.
993	4/10/2012 10:52:27 AM (376)	Motion	WARNING	The Virtual Machine missed 40 cycle(s) of PLC execution.
994	4/10/2012 10:52:27 AM (876)	Motion	WARNING	The Virtual Machine missed 104 cycle(s) of PLC execution.
995	4/10/2012 10:52:28 AM (376)	Motion	WARNING	The Virtual Machine missed 64 cycle(s) of PLC execution.
996	4/10/2012 10:52:28 AM (876)	Motion	WARNING	The Virtual Machine missed 70 cycle(s) of PLC execution.
997	4/10/2012 10:52:29 AM (376)	Motion	WARNING	The Virtual Machine missed 30 cycle(s) of PLC execution.
998	4/10/2012 10:52:29 AM (620)	Controller	WARNING	UserInfo : Alarm A23 : CPU is heavily loaded
999	4/10/2012 10:52:29 AM (876)	Motion	WARNING	The Virtual Machine missed 54 cycle(s) of PLC execution.
1000	4/10/2012 10:52:30 AM (376)	Motion	WARNING	The Virtual Machine missed 47 cycle(s) of PLC execution.

Lastly, the overall CPU load is 99%. Clearly this application is overloading the CPU:

KAS IDE User Guide | 11 Tools



11.4 Human-Machine Interface Editor

This chapter covers the 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

11.4.1 Using Kollmorgen Visualization Builder

KOLLMORGEN	Kollmorgen Automation Suite
KAS	KOLLMORGEN VISUALIZATION
Stife Ear Total Venetic Fill 2292222	BUILDER "Because Motion Matters"
	Kolimorgen Visualization Builder Software is part of the high-performance Motion Control and Automation Systems Development Suite of tools from Kolimorgen Using this software suite, you can quickly and easily create Operator Interface panels that are easy to develop and intuitive to use.

To work with Kollmorgen Visualization Builder, do as follows:

- Tag the PLC variables you want to export and map with the HMI (for more details, refer to Map Variables to HMI)
- Compile your project to generate the Modbus mapping file
- Create a KVB project 1 within the KAS IDE, and open it
- Design Your HMI Panel with KVB
- 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.

• 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. So you can directly go to Design Your HMI Panel with KVB.

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, refer to the following manual:

KVB Guide	Description
Kollmorgen	Quick Start that covers the most important points to install and use
Visualization Builder™	Kollmorgen Visualization Builder, in order to configure HMI Panels and PC
Quick Start Guide	operated control applications.

1There is no built-in feature to import/export KVB projects

KVB Guide

Kollmorgen
Visualization Builder
User Manual

Contains all the content to help you with Kollmorgen Visualization Builder.

11.4.1.1 Create A New KVB Controller

Description

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

Choose Operator Panel Choose your operator panel in the menu below	Choose Controller Choose your preferred controller or OPC server in the menu below Select the location of your project in the menu below
	AKI-CDT-MOD-15T Graphic 15.1" touch operator panel with support for 64k colors. 1024x768 pixels, TFT LCD display. Ethernet, USB, R5232 and R5485/R5422 ports. <u> Ernish</u> <u> Cancel</u>
PC AKI-CDT-MOD-15T	AKI-CDT-MOD-10T AKI-CDT-MOD-06T AKI-CDT-MOD-04T

 On the next dialog, select the Kollmorgen controller with the Modbus protocol, then click the Next button

Choose Operator Panel Choose your operator panel in the menu below	Choose Controller Choose your preferred controller or OPC server in the menu below	Select Location Select the location of your project in the menu below
Controllers Select brand DEMO Kollmorgen Kollmorgen	Select protocol	Next > < <u>Previous</u> <u>Finish</u> <u>Cancel</u>
O Remote Server	Browse Refresh	

• Enter the name of the project and where you want to create the project. Then click the **Finish** button

Project s\William.Gaffga\Documents\Kollmorgen Projects Browse Previous Finish Cancel
Finish
Cancel

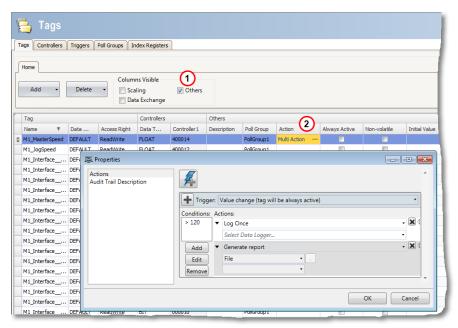
11.4.1.2 Import Variables Into The Project

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

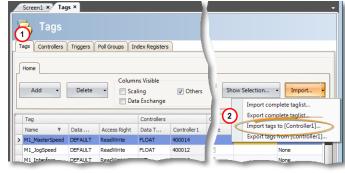
To reimport variables, (if they were changed or more were added), simply ensure they are tagged as "KVB" in the Variables list, recompile your KAS project, and open the KVB from the KAS IDE.

- 1. Select Others to display the Action column
- 2. You can edit the tag actions



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

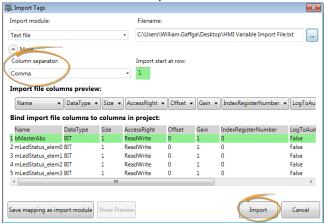


• In the import dialog, specify the filename by clicking the ... button

🕮 Import Tags	
Import module:	Filename:
Text file •	(
More Column separator: Import start at row: Comma	
Save mapping as import module Show Preview	Import Cancel

- 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

📧 Select Tags	
 All Items MasterAbs MulcdStatus_elem0 MulcdStatus_elem1 MulcdStatus_elem2 MulcdStatus_elem3 	
ОК	Cancel

 You are notified of how many items are successfully imported. Click the OK button to return to the project.

🖭 Import Result	×
Import was completed. Some items were changed, View log for more information. Imported items: 5 / 5	overwritten or skipped.
ОК	Log

11.4.1.3 Design Your HMI Panel with KVB

IMPORTANT

Do not modify **Project Name** and **Title** to keep consistency between Kollmorgen Visualization Builder and the KAS IDE.

1	Home	Project	System	Insert	View	Dynamics					
Build	Rebuild R	un Sim	ulate Debug	Download	Upload Databas		Back Up Project	Settings	Referenced Assemblies	Project Name Project Title	MyBigProject MyBigProject
	F	lun			Transfer			_	Pi	oject	

11.4.1.3.1 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.

11.4.1.3.2 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

11.4.1.3.3 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)

① 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.

11.4.1.4 Download the KVB Panel

To download your panel you have to use the Project ribbon in Kollmorgen Visualization Builder that contains the **Transfer** command . In order to download you must be connected directly to the HMI panel through an Ethernet cable. As the IP address is already defined (for more details, Configure the Controller), nothing special has to be done before transferring your panel to the graphic operator terminal.



Button	Description
Download	The Download control sends the project to an HMI-panel or a controller with an Kollmorgen Visualizer RT installation, in the network environment.
Upload Database	The Upload Database command will collect the database from an HMI-panel in the network environment to a file on the development PC. The panel will pause during the upload, and will be started automatically afterwards.
Export	The Export command builds and exports a copy of the project and all needed files to transfer the project to another location.

NOTE

554

- If you transfer your project on a USB stick, place it in the USB port of the AKI panel while it is booting up.
- For more details, refer to the online help in Kollmorgen Visualization Builder.

11.4.2 Design the Control Panel with the Internal Control Panel Editor

This section details the Controls and Properties used to define the Control Panel when you need to debug your application, as well as the procedure for Mapping Variables to the Control Panel controls.

11.4.2.1 Create Control Panel

Control Panel are managed in the Project Explorer and can be created as follows:

- 1. In the Project Explorer, right-click the Controller item to open the menu
- 2. Select the New Control Panel command
- 3. Right-click on the newly created item and select the Rename command to change its name
- 4. Double-click the new Control Panel to open it in the graphical editor

11.4.2.2 Use the Control Panel control library

• Select a control in the Libraries toolbox (Controls tab) and drag-and-drop it in your Control Panel.

Libraries			₽×
Function Blocks	Controls	Properties	
Name			
Analog buttons Bar Graph Bar Graph Binary Bitmap Charts Charts LeD Displays Links Meters Miscellaneous Scale Shapes Switches Text	3		

Figure 7-37: Control Panel Control Library

For an exhaustive list of controls, refer to Graphic Objects.

11.4.2.3 Edit the Control panel

When a control is selected, you can change its properties (displayed in the Libraries toolbox) by double-clicking the **Value**.

Libraries			đΧ
Function Blocks	Controls	Properties	
Property	Value		^
Identifier Aspect Variable symbol Spying delay Border size Border color Data format Data color Background color	Round swit CUSTOM bEStop 0 0	ch	=
Background mode Text	TRANS		
Text color Text mode	NORMAL		~

Figure 7-38: Control Panel Control Properties

For an exhaustive list of properties, refer to Graphic Objects Properties

NOTE

- You can perform multi-selection with the mouse (all the controls that are even partly inside the selection area are selected)
- You can add controls to your selection either with the Ctrl or Shift keys
- You can use Arrow keys to move the Control Panel page Up, Down and sideways.
- You can use 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 (do not forget to release the mouse button first, before the **Ctrl** key).



Figure 7-39: Control Panel - Selection of Controls

11.4.2.3.1 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.

•	 Undo		Grid Settings	×
	Redo		Show Grid	
1	Сору		X =	16 💌
	Paste		Y =	16 💌
•	Cut	· ·	5	Snap to Grid 🔽
	Grid		_	
			🗌 X max	0 🕂
			TY max	0 茾
			ОК	Cancel

Parameter	Description		
Show Grid	Display or hide the grid in the Control Panel		
X= / Y=	pace between each grid dot in X/Y directions		
Snap to Grid	Controls will be 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. Please note that objects may still be placed within the grayed out area as it is merely a visual aid.		

11.4.2.4 Mapping Variables to the Control Panel

How do I define a variable for PLC programs?

To link your Control panel with the PLC programs, some controls contain a property called **Variable symbol**

Libraries				
Function Blocks	Controls	Properties		
Property	Value			
Identifier	Round switch			
Aspect	CUSTOM			
Variable symbol	bEStop	N		
Spying delay	0	43		
Border size	0			

Figure 7-40: Map variables to a Control Panel control

To map the variable:

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

Libraries		ē×		C	ontrol Panel				ι	IntitledST
Function Blocks	Controls	Properties		-						
Property	Value		1				Co	ontro	ol Pan	el
Identifier	Push butt									
Aspect	DEMULT									
Variable symbol Spying delay	bMasterA	bs	•	#	Machine	STATE		E-8	STOP	4
Border size	2							1		
Border color			? ?		Auto Mo	ode		1		4
Data format	_		岁							- T
Data color			<u>**</u>		Manual	Mode				(
Background color	OPAQUE								-	
Background m Text	Abs Move				OFF			STOP	RUN	
Text color			10							
Text mode	BOLD		48							
Font name	ARIAL		P	M	anual Mo	ode Fur	ction	s	AL	ito Mod€
Font size	30									
Action Pathname for "	PUSHBUT	ION	- ₽	Trav	el Speed				Machine	e Speed
Pathname for "			1	F	· · · · · · · · · · · · · · · · · · ·			_		
Bitmap display	ORIGINAL				s Move					
					<u>ل</u> هع			_		
				Re	Move					
Project Explorer	Libraries		(30,37	707						
	cioraneo		(50,57							
Dictionary										₽×
Controller:PLC		•		Frack Selection						
Variables Enu	ums Bit I	Fields								
14 Name		Tyr	Dim.	Attrib.	Init val	User	КVВ	Tag	Description	
Machine		aVT			0					*
bMaste		BOOL			FALSE					
bMaste		BOOL			FALSE					
bEStop		BOOL			FALSE					
bLedSt		BOOL	[03]							
Profiles		ProfilesCo		Dood Only			_			

Figure 7-41: Map Variables to a Control Panel Control in the Graphical Editor

The Variable symbol is automatically updated in the Properties tab.

NOTE

A warning will be generated when the program is compiled if the control is mapped to a variable which is not in the Dictionary. Double-clicking the warning will highlight the control object and open the editor so the variable can be defined.

11.4.2.5 Graphic Objects

Below are available basic objects you can insert in your graphics:

11.4.2.5.1 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).

- Aspect (shapes)
- Background color
- Background mode
- Border color
- Border size
- Color when not connected
- Data format
- Direction (basic shapes)
- FALSE color
- Font name
- Font size
- Identifier

- Spying delay
- Text
- Text color
- Text mode
- TRUE color
- Variable symbol

11.4.2.5.2 Bitmaps

Bitmap file (BMP, GIF, JPG) can be inserted in the graphic area.

Properties:

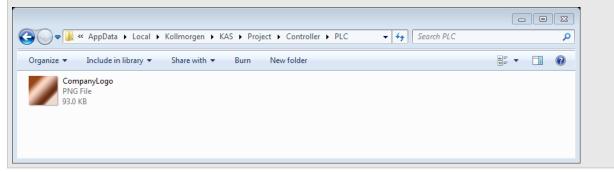
- Background color
- Background mode
- Bitmap display mode
- Bitmap pathname
- Border color
- Border size
- Border style
- Font name
- Font size
- Identifier
- Text
- Text color
- Text mode

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 IDEwill attempt 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.



11.4.2.5.3 Scales

0 10 20 30 40 50 60 70 80 90 100

Scales are static drawings representing an X or Y axis, generally used to document other objects such as trend charts or bargraphs.

Properties:

558

- Background color
- Background mode

- Border color
- Border size
- Border style
- Color when not connected
- Direction (scale)
- Font name
- Font size
- Identifier
- Maximum value
- Minimum value
- Nb divisions (main)
- Nb divisions (small)
- Placement (scale)
- Scale color
- Text
- Text color
- Text mode

11.4.2.5.4 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)
- Background color
- Background mode
- Border color
- Border size
- Border style
- Data format
- Font name
- Font size
- Identifier
- Spying delay
- Text
- Text color
- Text mode
- Variable symbol

11.4.2.5.5 Switches and 2-state displays

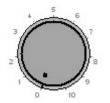


Buttons, switches and 2-state displays are used for control or display of a Boolean variable.

- Action (switch)
- Aspect (switches)
- Background color
- Background mode
- Bitmap display mode
- Bitmap for "FALSE" state

- Bitmap for "TRUE" state
- Border color
- Border size
- Data color
- Data format
- Font name
- Font size
- Identifier
- Spying delay
- Text
- Text color
- Text mode
- Variable symbol

11.4.2.5.6 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
- Background mode
- Border color
- Border size
- Border style
- Data color
- Data format
- Font name
- Font size
- Identifier
- Maximum value
- Minimum value
- Scale color
- Spying delay
- Text
- Text color
- Text mode
- Variable symbol

11.4.2.5.7 Bar Graphs

Bargraphs are rectangles filled according to the value of an analog variable. Bargraphs can be horizontal or vertical.

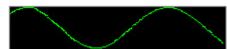
Properties:

560

- Background color
- Background mode
- Border color
- Border size

- Border style
- Data color
- Data format
- Direction (bar graph)
- Font name
- Font size
- Identifier
- Maximum value
- Minimum value
- Spying delay
- Text
- Text color
- Text mode
- Variable symbol

11.4.2.5.8 Charts

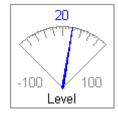


Charts enable the tracing of a variable as with an oscilloscope.

Properties:

- Aspect (trend charts)
- Background color
- Background mode
- Border color
- Border size
- Border style
- Data color
- Data format
- Font name
- Font size
- Identifier
- Maximum value
- Minimum value
- Nb of points (trends)
- Spying delay
- Text
- Text color
- Text mode
- Variable symbol

11.4.2.5.9 Analog meters



Analog meters provide a graphical display of an analog value.

- Background color
- Background mode

- Border color
- Border size
- Border style
- Data color
- Data format
- Font name
- Font size
- Identifier
- Link
- Maximum value
- Minimum value
- Nb divisions (main)
- Nb divisions (small)
- Scale color
- Spying delay
- Text
- Text color
- Text mode
- Variable symbol

Sliders

Sliders are used for entering an analog value with a horizontal or vertical mouse driven cursor.

Properties:

- Background color
- Background mode
- Border color
- Border size
- Border style
- Data color
- Data format
- Direction (slider)
- Font name
- Font size
- Identifier
- Maximum value
- Minimum value
- Scale color
- Spying delay
- Text
- Text color
- Text mode
- Variable symbol

11.4.2.5.10 Digital meters



Digital meters (digits) display the value of a variable with the same aspect as a digital clock.

- Aspect (digits)
- Background color
- Background mode
- Border color
- Border size
- Border style
- Data color
- Data format
- Font name
- Font size
- Identifier
- Maximum value
- Minimum value
- Spying delay
- Text
- Text color
- Text mode
- Variable symbol

11.4.2.5.11 Links

Back to main page

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
- Background mode
- Border color
- Border size
- Border style
- Font name
- Font size
- Identifier
- Link
- Text
- Text color
- Text mode

11.4.2.5.12 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.

- Background color
- Background mode
- Border color
- Border size
- Border style
- Data color
- Data format
- Font name
- Font size
- Identifier
- Spying delay
- Text

- Text color
- Text mode

11.4.2.5.13 Gauges



Analog view meter.

Properties:

- Background color
- Background mode
- Border color
- Border size
- Border style
- Data color
- Data format
- Font name
- Font size
- Identifier
- Maximum value
- Minimum value
- Nb divisions (main)
- Nb divisions (small)
- Needle color
- Spying delay
- Title
- Title color
- Title placement
- Variable symbol

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

564

11.4.2.6 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-clicking an object will open the Graphic Item Properties dialog box, which lists the properties particular to that object.

Kollmorgen	Auton	Graphic Item Properties	x	
2		Name	Value	
Co	ontrol Pa	Identifier	Round switch	
		Object type	Button	
		Aspect	CUSTOM	
# Machine STATE	E-STOP	Variable symbol	bEStop	
	çş	Spying delay	0	
Auto Mode		Border size	0	
	á di	Border color		
Manual Mode		Data format		
		Data color		
OFF	STOP RUN	Background color		
		Background mode	TRANS	
		Text		
Manual Mode Function	s	Text color		
	- 1	Text mode	NORMAL	
Travel Speed	Mach	Font name	ARIAL	
		Font size	16	
Abs Move		Action	SWITCH	
	ويربيه والمحمد والمحمد والمحمد والمحمول	Pathname for "TRUE" bitmap	SwitchP SWITCH 0x50.bmp	
		Pathname for "FALSE" bitm		
		Bitmap display mode	STRETCH	
			OK Cancel	J

Name	Description of Value
Action (switch)	Indicates the possible mouse action for switches. The following values are possible: STATIC = no mouse action PUSHBUTTON = the variable is forced to TRUE when pressed and to 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. The following values are possible: STATIC = no mouse action EDIT = double-click opens an edit box for entering the variable value
Aspect (digits)	This property indicates the type of drawing for a digital meter. Possible aspects are: DEFAULT = plain drawing BEZEL = all segments have a 3D effect
Aspect (shapes)	This property indicates the type of basic shape to be drawn. Possible aspects are: CYLINDER = a 3D like cylinder ELLIPSE = an ellipse HALFELLIPSE = one half of an ellipse GATE = a simple vector drawing for a valve RECTANGLE = a rectangle ROUNDRECT = a rectangle with rounded corners TRIANGLE = a triangle
Aspect (switches)	This property 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

Name	Description of Value
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	This property 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.
mode	This property indicates whether the background of the object must be filled or not. If this property is OPAQUE, then the background is filled with the specified background color. If this property is TRANS (transparent) then 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 will affect the real-time performances of graphic updates.
Bitmap display mode	For bitmap-based objects, this property 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 having the "CUSTOM" aspect, this property specifies the pathname of the bitmap to be displayed 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 having the "CUSTOM" aspect, this property specifies the pathname of the bitmap to be displayed 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
Bitmap pathname	For bitmaps, this property 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
Border color	This property indicates the color of the border drawn around the object.
	This property indicates the width of the border drawn around the object, expressed as a number of pixels. If this property is 0, then no border is drawn.

566

Name	Description of Value			
Border style	This property indicates the possible 3D effect used for drawing the border around the object. Possible values are: FLAT = no 3D effect 3DUP = depressed 3D effect 3DDOWN = pressed 3D effect 3D = default 3D effect			
Color when not connected	For shapes, this property indicates the color used for filling shapes when no variable is attached to the graphic object.			
Data color	This property indicates the c within the object (for examp		sent the value of a connected variable	
Data format	If defined, this property indi	cates that the valu ect. You must spec	e of the connected variable must be ify for this property a format string	
	① IMPORTANT			
	The " text " property is ignor	ed when a data for	mat is specified.	
			ous "printf" function of "C" language. It of the following possible pragmas that	
	%s = default formatting according to IEC syntax %d = integer (decimal) %X = hexadecimal %g = floating point %. <i>n</i> f = decimal real (<i>n</i> is the number of displayed decimal digits)			
	Below are some examples:			
	Format	Value	Displayed string	
	%d	12.3	12	
	Var = %g meters	1.2	Var = 1.2 meters	
	%.2f	1.12345	1.12	
	NOTE Only one % pragma can be u	used in a string.		
Direction (bar graph)	For bar graphs, this property to the top or to the bottom.	y indicates the grow	wing direction: to the left, to the right,	
Direction (basic shapes)			ses or cylinders, this property indicates right, to the top or to the bottom.	
Direction (scale)	For scales, this property 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 property india (TOP).	cates whether the	slider is horizontal (RIGHT) or vertical	
FALSE color	For shapes, this property inc variable has the FALSE state,		ed for filling shapes when the attached Js.	
Font name	This property indicates the n graphic object.	ame of the charact	ter font used for drawing texts in the	

Name	Description of Value
Font size	This property 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	You can 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	This property indicates the name of the target .GRA animated document for shortcuts. If no directory is specified in the link, then the file is searched in the project folder.
Maximum value	For analog animated objects (meters, bar graphs or trends) this property 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 (meters, bar graphs or trends) this property 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 property indicates the number of main division marks to be drawn in the scale.
Nb divisions (small)	For objects including a graphic scale, this property 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 property 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 property indicates the location of the scale within the object rectangle: on the left, on the right, on the top or at the bottom.
Scale color	For objects including a graphic scale, this property 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, refresh is done as fast as possible.
Text	If defined, this property indicates the text to be displayed on the graphic object.
	IMPORTANT This property is imported when a data format is specified
_	This property is ignored when a data format is specified.
Text color	This property indicates the color used for inserting texts in the graphic object.
Text mode	This property indicates the font effect used for drawing texts in the graphic object. Possible values are: HIDE = text is not displayed NORMAL = normal font BOLD = bold text ITALIC = italic text 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.

Name	Description of Value
Title placement	For gauges, this is the position of the title within the gauge.
TRUE color	For shapes, this property indicates the color used for filling shapes when the attached variable has the TRUE state, or non zero for analogs.
Variable symbol	It 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".

11.4.2.7 Operate the Control Panel

The Example program has a default control panel built-in to make it easy to start an application. Perform the following steps to operate the control panel:

1. Double-click on **Control Panel** in the Project Explorer to open the form



Figure 7-42: Control Panel

- 2. Start by moving 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

11.4.2.7.1 About KAS Simulator Display

The KAS Simulator displays the status and position of the axes. It also displays the log messages.

Kollmorgen Automation Su	ite Simulator -	[Axes]		
🙈 File Help				
Axes	Custom 3	IO Editor		
DriveAXIS1			DriveAXIS2	
77.47 (77.47)			77.49 (77.49)	
Log	₽×	System Terminal		₽×
Filter *	*	Loading buses	and IOs drivers	
Time Source L	.evel	Initialisation do	ne !!!	
<	>			
			IPO: 1	0.155.94.43

Figure 7-43: Display of KAS Simulator

You can 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

11.4.2.8 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 the following documentation:

- Getting Started
- User Manual
- Technical Reference PLC Library
- Technical Reference Motion Library
- Online Help

12 Advanced Topics

12.1 Coordinated Motion

Coordinated motion in KAS IDE is discussed in several locations and manners.

- Overview this section helps you to understand the concepts behind Coordinated Motion and the terminology associated with Coordinated Motion.
- Design Coordinated Motion this section helps you to get started quickly with coordinated motion by walking you through the steps of setting up a project that uses coordinated motion.
- Functions and Function Blocks this is the reference section for function block parameters.

NOTE

The coordinated motion library supports coordinated motion for up to 128 axes. Hardware limitations may impose a lower limit for most applications.

12.1.1 Overview

This section provides an overview of Coordinated Motion, including general concepts you will need to understand to use Coordinated Motion.

- Coordinated Motion Terminology
- Group State Diagrams
- Coordinate Systems

Create a Linear or Circular Coordinated Motion Application.

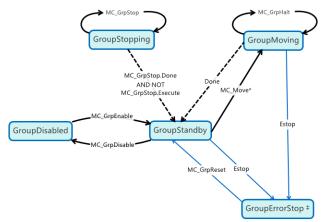
12.1.1.1 Coordinated Motion Terminology

Term	Definition
ACS	Axes Coordinate System. The system of coordinates related to the physical motors and the single movements caused by the single drives.
Blending	A way that consecutive function blocks cooperate in the transition from the first to the next.
Contour Curve	Inserted curve that modifies the original path. It is the resulting curve after blending
Coordinate system	The reference system in which a coordinate or path is described
Corner deviation	The shortest distance between the programmed corner point and the contour curve
Corner distance	Distance of the start point of the contour curve to the programmed target point.
Direction	The orientation components of a vector in space. (Note: this is different from the MC_Direction input as used in part 1).
Drive	A unit controlling a motor via the current and timing in its coils
Group-FB	The set of function blocks that can work on a group of axes

Term	Definition
MCS	Machine Coordinate System. The system of coordinates that is related to the machine. A Cartesian coordinate system with the origin in a fixed position relative to the machine (the origin is defined during the machine setup).
	Sometimes called "World Coordinate System" or "Base Coordinate System". (Note: with Cartesian build machines, MCS is a Cartesian Coordinate system and may be identical to ACS, or mapped via a trivial transformation). The coordinate system from the physical multiple axes ACS is linked to the MCS via a kinematic transformation (forward and backward conversion). The MCS represents an imaginable space with up to 6 dimensions.
Motor	An actuator focused to a movement, converting electrical energy in a force or torque.
Orientation	The rotational components of a vector in space.
Path	Set of continuous positions and orientation information in multi- dimensional space Geometrical description of a space curve that the TCP of an axesgroup moves along.
PathData	Description of a path which can include additional information like velocity and acceleration.
PCS	Product Coordinate System or Program Coordinate System. The PCS is based on the MCS typically by shifting and maybe rotating the MCS. The Zero point of the PCS is related to the product and can be changed during runtime by the program. The real work piece can have a rotation or shift to the MCS coordinate system or even might be moving relative to the MCS coordinate system. By specifying a trajectory in PCS one is able to describe the trajectory independent from the machine situation. To map these two worlds (MCS to PCS and vice versa), a Cartesian or cylindrical transformation is normally done.
Position	Position means a point in space which is described by different coordinates. Depending on the used system and transformation it can consist of up to 6 dimensions (coordinates) meaning 3 Cartesian coordinates in space and 3 coordinates for the orientation.
	In ACS there can be even more than 6 coordinates.
	If the same position is described in different coordinate systems the values of the coordinates are different.
Scara	A special kinematic for robot or handling applications.
Speed	Speed is the absolute value of the velocity without direction.
Synchronization	Combines an axis or axes group (as slave) with an axis as master in order that the slave executes its path with synchronization to the progress of the master, meaning linked to a one dimension source for synchronization.
ТСР	Tool Center point, the point in the machine that is commanded to move, typically the center or the head of the tool. It can be described in different coordinate systems.
Tracking	Is characterized by an axis group that follows with its movement the movement of another axis group.

Term	Definition
Trajectory	Time dependent description of the path the TCP of an axes group moves along. Additionally to the geometrical description of the space curve, time dependent state variables like velocity, acceleration, jerk, forces etc. are specified.
Velocity	 For a group of axes this means: in ACS the velocities of the different axes in MCS and PCS it provides the velocity of the TCP

12.1.1.2 Group State Diagrams



‡ - A limitation exists where a single axis ErrorStop condition will not change the AxesGroup state to GroupErrorStop. (PLCopen Part 4, Section 3.1)

12.1.1.3 Coordinate Systems

There are three different coordinate system (CS) types:

- Machine (MCS)
- Axes (ACS)
- Product/Program (PCS)

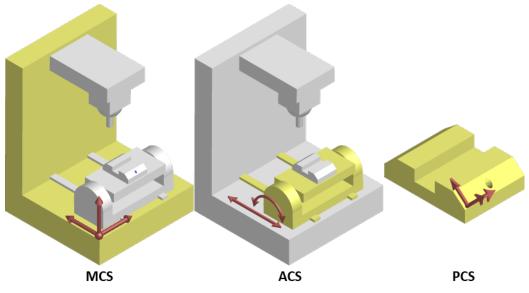
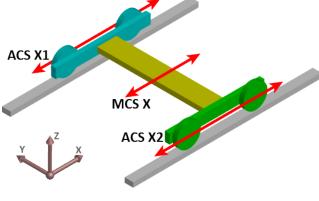


Figure 8-1: Examples of CS types on a machine and part.

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.

For 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 also be commanded independently for minor alignment adjustments.



★ TIP

The Product Coordinate System is often rotated and/or offset from the Machine Coordinate System.

12.1.2 Design Coordinated Motion

This section discusses how to create a coordinated motion application, including adding coordinated motion to existing applications.

For more information on Coordinate Motion an the associated functions and function blocks see:

- Coordinated Motion in the Advanced Topics section
- Coordinated Motion Function Blocks

12.1.2.1 Create a Linear or Circular Coordinated Motion Application

A Coordinated Motion application can be created in one of two 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. More information about Axes Groups can be found in the section What are Axes Groups?.

NOTE

Coordinated motion can only be performed with PLCopen axes. Pipe Network axes do not support this feature, although 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.

Related axes are "grouped" in an axes group. Coordinated motion is then performed on an axes group. For more information see What are Axes Groups?.



There are two vital concepts to remember 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, and non-group coordinate system positions are only affected by non-group motion.

Group motion commands	Non-Group motion commands
 Coordinated Motion Group control Info Motion MC_AxisSetDefaults MC_GrpHalt MC_GrpSetOverride MC_MoveCircAbs MC_MoveCircRel MC_MoveDirAbs MC_MoveLinAbs MC_MoveLinAbs MC_MoveLinRel 	 Motion/PLCopen Control I/O Info PLCopenMotion MC_AddSuperAxis MC_Halt MC_MoveAbsolute MC_MoveAbsolute MC_MoveAbsolute MC_MoveAbsolute MC_MoveSuperimp MC_MoveVelocity MC_RemSuperAxis MC_SetOverride

NOTE

Typically 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.

Typically, the following set of 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 needs to be called between MLMotionInit() and MLMotionStart().

Inst_MC_CreateAxesGrp(TRUE, 'GROUP1', 6, 2, Group1_ref);

In the example above, 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' will be 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 the example above, the kinematic limits for axes group 'Group1_ref' will be set. The velocity limit will be set to 100.0 user units/second, acceleration and deceleration limits will be set to 300.0 user units/second2 and jerk will be set to 1000.0 user units per second3 (Jerk will be supported in a future release).

4. Call MC_CreatePLCAxis (AxisName, BusInterface, BusAddress, AxisNumber, AxisType, UserUnits, FeedbackUnits, Rollover, UpdateRate) to create a Coordinated Motion Axis. This function needs to 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 the example above:

- 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, which 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 be added to the group.

```
Inst_MC_AddAxisToGrp(TRUE, Group1_ref, CoordAxis1_ref, 0);
Inst_MC_AddAxisToGrp(TRUE, Group1_ref, CoordAxis2_ref, 1);
```

In the example above, 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. Note that 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 the example above, drives CoordAxis1_ref and CoordAxis2_ref will be enabled and the position loop will be closed. Note that parameters 'TRUE, TRUE, 0' are place holders for future use and are not currently used.

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 the example above, 'Group1_ref' state will be changed from GroupDisabled to GroupStandby. The group must be in GroupStandby in order to perform motion.

 For the examples that follow, we want to set the current location of the axes in the group to 0, 0. This can be 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 the example above, the axis positions of 'Group1_ref' will be 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. The buffer mode, '0', is a placeholder for future use and is not currently used.

NOTE

No motion will be performed when this function block is executed.

- 10. Optional: To Add more axes to the group, modify the above code in the following 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 that will added to the group.
 - In Step 6: Add the additional axes to the group.
 - In Step 7: Power on the additional axes.
 - In Step 9: You will need to increase the size of the PosAbs array so it matches the number you used in step 2, and set the position of the additional axes to zero.

After the above function calls have been made, we can start coordinated motion moves.

Performing a Linear Move

Performing a Circular Move

12.1.2.1.1 What are Axes Groups?

Related axes are grouped in an Axes Group to support interpolation. Axes Groups are accessed via the type AXES_GROUP_REF. The following image shows the relationships between the different CSs and groups.

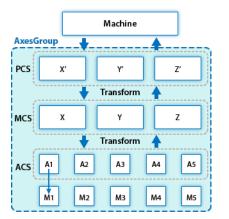


Figure 8-2: Overview of AxesGroup

The Axes Group, shown in blue above, 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.

12.1.2.1.2 Performing a Linear Move

Linear moves can be programmed using absolute or relative positions using the following function blocks:

- MC_MoveLinAbs which commands interpolated linear movement on an axes group to the specified absolute positions.
- MC_MoveLinRel which 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" on page 574). Once these steps have been performed, a linear move can be performed.

In the following examples, two linear moves will be performed. The first move is an absolute linear move that goes from (0, 0) to (100, 200). The second move is a relative linear move that goes a distance of (-75, 50) from the end of the first move. The BufferMode input is set to 'Buffered', meaning this move will wait for the first move to complete before it begins executing.

• To Perform an Absolute Linear Move

Call MC_MoveLinAbs (Execute, AxesGroup, PositionArray, Velocity, Acceleration, Deceleration, Jerk, CoordSystem, BufferMode, TransitionMode, TransitionParameter). PositionArray is an array of absolute end positions containing one position for each axis in the group. The inputs velocity, acceleration, deceleration, and jerk establish the maximum values for the move.

In this example, PosArrayAbs[0] represent the x-axis and PosArrayAbs[1] represent the y-axis.

```
PosArrayAbs[0] := 100;
PosArrayAbs[1] := 200;
TransParam[0] := 0;
TransParam[1] := 0;
Inst_MC_MoveLinRel(TRUE, Group1_ref, PosArrayAbs, MaxVel, MaxAcc,
MaxDec, 0, MC_COORDINATE_SYSTEM_ACS, 1, 0, TransParam);
```

In the example a linear move will be 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 will be moved to 100.0. The axis stored in postiion 1 of the group will be 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/sec2'.
- The maximum jerk is currently not supported and can be set to a value of 0.
- The coordinate system is ACS
- The BufferMode is set to 1, indicating the move is buffered. For more information about buffer modes, see the Buffer Modes overview.
- The TransitionMode is set to 0, indicating no transition mode will be used. For more information about transition modes, see the Transition Between Moves section.
- The TransParam array is required and the contents can be set to 0 since the transition mode is not being used. There has to be one array entry for each axis in the group.

• To Perform a Relative Linear Move

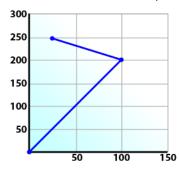
578

Call MC_MoveLinRel (Execute, AxesGroup, Distance, Velocity, Acceleration, Deceleration, Jerk, CoordSystem, BufferMode, TransitionMode, TransitionParameter). The Distance input is an array of distances, one distance for each axis in the group. The inputs velocity, acceleration, deceleration, and jerk establish the maximum values for the move.

In this example, DistArrayRel[0] represent the x-axis and DistArrayRel[1] represent the y-axis.

```
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;
TransParam[1] := 0;
Inst_MC_MoveLinRel(TRUE, Group1_ref, DistArrayRel, MaxVel, MaxAcc,
MaxDec, 0, MC_COORDINATE_SYSTEM_ACS, 1, 0, TransParam);
```

In the example above, 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 will be moved a distance of -75.0. The axis stored in postiion 1 of the group will be moved a distance of 50.0.



To Perform a 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 will do two dimensional moves if it only has two valid axes mapped to it.

In order to perform higher dimensional moves, additional axes must be added to the group. The steps to do this are detailed in Create a Linear or Circular Coordinated Motion Application.

After the additional axes are added perform the following steps.

- 1. From within the Dictionary, update the array size of the variable being passed (PosArrayAbs and DistArrayRel in the examples above) to the Position input so that 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.

12.1.2.1.3 Performing 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" on page 574). Once these steps have been performed, a circular move can be performed.

In the following examples, two circular moves will be performed. The first move is an absolute circular move that goes from (0, 0) to (90, 90). CircMode specifies that the aux point (0, 180) will be crossed during the paths start to end. The second move is a relative circular move whose end point is (90, 90) from the end of the first move. In this move, CircMode specifies that the aux point (0, 90) is the relative center of the circle. The BufferMode input is set to 'Buffered', meaning this move will wait for the first move to complete before it begins executing.

• To perform an Absolute Circular Move:

Call MC_MoveCircAbs (Execute, AxesGroup, CircMode, AuxPoint[], EndPoint[], PathChoice, Velocity, Acceleration, Deceleration, Jerk, CoordSystem, BufferMode, TransitionMode, TransitionParameter).

In this example, PosCircAuxAbs[0] and PosCircEndAbs[0] represent the x-axis. PosCircAuxAbs [1] and PosCircEndAbs[1] represent the y-axis.

```
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;
Inst_MC_MoveCircAbs(TRUE, Group1_ref, MC_CIRC_MODE_BORDER,
PosCircAuxAbs, PosCircEndAbs, MC_CIRC_PATHCHOICE_CLOCKWISE, MaxVel,
MaxAcc, MaxDec, 0, MC_COORDINATE_SYSTEM_ACS, MC_BUFFER_MODE_
BUFFERED, MC_TRANSITION_MODE_NONE, TransParam);
```

In the example a circular move will be performed on axis group 'Group1_ref'.

- CircMode is defined as MC_CIRC_MODE_BORDER. This mode indicates that the AuxPoint array input will indicate a point on the circle which is crossed on the path from the starting point to the end point. See Circular Moves Diagrams for more information on 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, MC_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 will be 90.0. The absolute end point of the axis stored in position 1 of the group will be 90.0.
- PathChoice is only relevant when CircMode is set to MC_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/sec2'.
- The maximum jerk is currently not supported and can be set to a value of 0.
- The coordinate system is ACS
- The BufferMode is set to MC_BUFFER_MODE_BUFFERED, indicating the move is buffered. For more information about buffer modes, see the Buffer Modes overview.
- The TransitionMode is set to MC_TRANSITION_MODE_NONE, indicating no transition mode will be used. For more information about transition modes, see the Transition Between Moves section.

580

• The TransParam array is required. The TransParam array is a 2-element array containing the corner distance and velocity for the transition. Transitions are not used in this example and therefore the contents can be set to 0.

• To perform a Relative Circular Move:

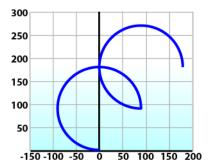
Call MC_MoveCircRel (Execute, AxesGroup, CircMode, AuxPoint[], EndPoint[], PathChoice, Velocity, Acceleration, Deceleration, Jerk, CoordSystem, BufferMode, TransitionMode, TransitionParameter).

In this example, PosCircAuxRel[0] and PosCircEndRel[0] represent the x-axis. PosCircAuxRel[1] and PosCircEndRel[1] represent the y-axis.

```
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, MC_CIRC_MODE_CENTER,
PosCircAuxRel, PosCircEndRel, MC_CIRC_PATHCHOICE_CLOCKWISE, MaxVel,
MaxAcc, MaxDec, 0, MC_COORDINATE_SYSTEM_ACS, MC_BUFFER_MODE_
BUFFERED, MC_TRANSITION_MODE_NONE, TransParam);
```

In the example all the variables have the same meaning as the circular absolute example except:

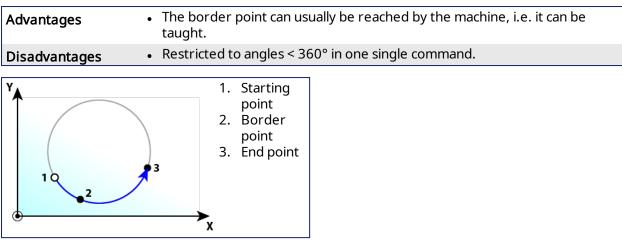
- CircMode is defined as MC_CIRC_MODE_CENTER. This mode indicates that the AuxPoint
 array input will indicate the center point of the circle. See Circular Moves Diagrams for
 more information on CircMode movement options.
- The AuxPoint array, 'PosCircAuxRel', defines the relative center point of the circle. The contents of this array are determined by the CircMode variable, MC_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 will be 90.0. The relative end point of the axis stored in postiion 1 of the group will be 90.0.
- PathChoice is relevant when CircMode is set to MC_CIRC_MODE_CENTER. In this case, PathChoice is MC_CIRC_PATHCHOICE_CLOCKWISE which specifies the direction of the path.



12.1.2.1.3.1 Circular Moves Diagrams

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.



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.

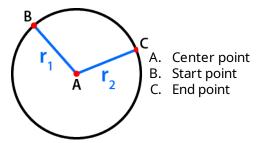
Advantages	Full 360° moves are possible.		
Disadvantages			
A 360° move will b	 1. Starting point 2. Center point 3. End point 4. Clockwise move 5. Counter-clockwise move 5. Counter-clockwise move 6. Source 7. Starting point 8. Starting point 9. Starting		
 Y I. Starting & Ending point I. Starting & Ending point I. Center point I. Clockwise move I. Counter-clockwise move 			

12.1.2.1.3.2 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 will be the same but calculation errors or input data precision can cause the two distances to be slightly different. If validation fails then PLCopen error 50 ("Cannot construct a circle with specified parameters", see "PLCopen Function Block ErrorID Output" on page 419) will be returned from the function block. Input parameters are validated using the methods described below.

582

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.



There are two methods for validating the circle parameters, the default method and a method using a custom tolerance.

NOTE

If MC_GRP_PARAM_CIRCLE_TOLERANCE is set to zero (0), the default validation method will be used.

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 then this requirement will be 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 :

r₂= r₁(1+**ɛ**)

If **ɛ** exceeds a value of 10-5, then the validation will fail and PLCopen error 50 will be returned from the function block. will be returned from the function block.

Examples of Default Validation

- If the desired circle has a radius of 50 user units, then the center must be specified with a precision of 0.0005 user units.
- If the desired circle has a radius of 2,000 user units, then the center must be specified with a precision of 0.02 user units.
- If the desired circle has a radius of 500,000 user units, then the center must be specified with a precision of 5 user units.

Validation with Custom Tolerance

A custom tolerance can be specified via the group parameter MC_GRP_PARAM_CIRCLE_ TOLERANCE (see "Axes Group Parameters" on page 413). 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 r2 is greater than r1, we can write r2 in terms of r1 and a deviation value named Δ :

r2= r1 + ∆

The validation will fail if \triangle exceeds the value of the group parameter MC_GRP_PARAM_CIRCLE_TOLERANCE and PLCopen error 50 will be returned from the function block.

Examples of Validation using a Custom Tolerance

- If r1 has a value of 50 user units and the circle tolerance parameter is .1 user units, then r2 must be in the range of 49.9 to 50.1 user units.
- If r1 has a value of 2000 user units and the circle tolerance parameter is 1 user units, then r2 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. For example, the LREAL version of cos is cosl. This will help avoid errors.

12.1.2.1.3.3 How to perform a complete circular move

A full circle may be performed using the following procedure.

- 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°.

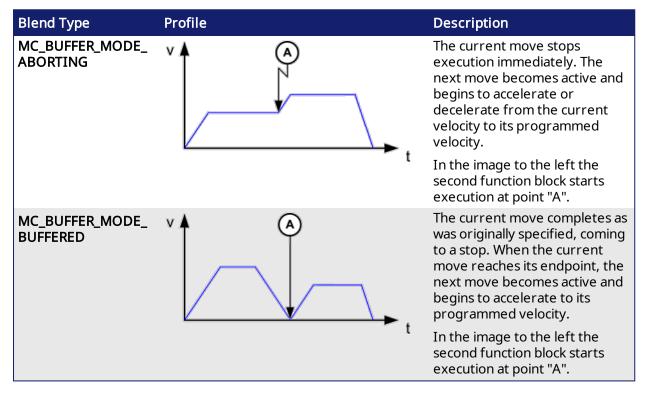
12.1.2.2 Blending Between Moves

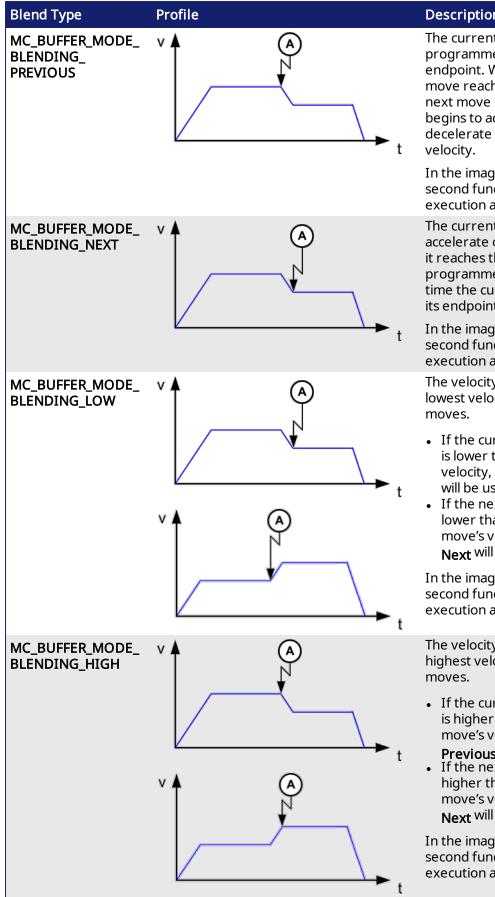
Some Coordinated Motion Function Blocks have a BufferMode input parameter. Possible buffer modes include:

- MC_BUFFER_MODE_ABORTING = 0
- MC_BUFFER_MODE_BUFFERED = 1
- MC_BUFFER_MODE_BLENDING_PREVIOUS = 2
- MC_BUFFER_MODE_BLENDING_NEXT = 3
- MC_BUFFER_MODE_BLENDING_LOW = 4
- MC_BUFFER_MODE_BLENDING_HIGH = 5

When the current and next motion function blocks are blended (2 through 5 above), the axes group will not stop between motions. The velocity will be blended according to the specified blending mode.

In addition, the Function Block TransitionMode parameter can be set to provide a smooth circular arc between moves. Transition modes are described in Transition Between Moves.





Description

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

In the image to the left the second function block starts execution at point "A".

The current move will begin to accelerate or decelerate so that it reaches the next move's programmed velocity at the time the current move reaches its endpoint.

In the image to the left the second function block starts execution at point "A".

The velocity is blended to the lowest velocity of the two

- If the current move's velocity is lower than the next move's velocity, Blending Previous will be used.
- If the next move's velocity is lower than the current move's velocity, Blending Next will be used.

In the images to the left the second function block starts execution at point "A".

The velocity is blended to the highest velocity of the two

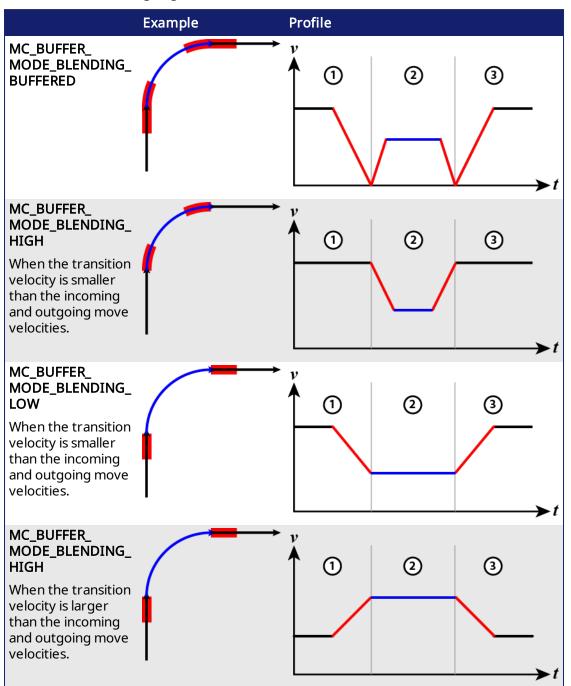
- If the current move's velocity is higher than the next move's velocity, Blending Previous will be used.
- If the next move's velocity is higher than the current move's velocity, Blending Next will be used.

In the images to the left the second function block starts execution at point "A".

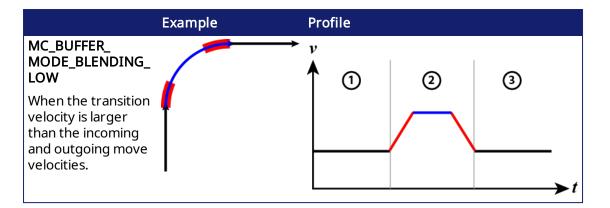
12.1.2.3 Blending with Transitions

When blending is specified when a transition is specified, then the blending 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.

The most common blending mode choices for applications are listed below showing the velocity profile and where acceleration occurs on the path. In the examples, blue denotes the transition arc, red denotes where the acceleration occurs, 1 indicates the incoming move, 2 indicates the transition, and 3 indicates the outgoing move.



586



12.1.2.4 Transition Between Moves

A transition mode must be specified when a new move is appended to a move that is already in progress. Different transition parameters may be required, depending on the transition mode. This characterizes the contour of the transition segment.

The supported transition modes are:

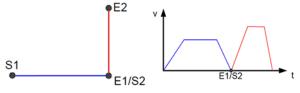
- No Transition ("TMNone")
- Corner Distance ("TMCornerDistance")

Transition Mode	Number of Transition Parameter Array Elements			Transition Parameter Description	Units
TMNone	0				
TMCornerDistance 2	2	0	Corner Distance	Distance to the corner of the deviation and the return point from the original contour	User units
		1	Velocity	The velocity value fo the transition segment	User units per second

Table 8-1: Transition Mode Parameters

12.1.2.4.1 No Transition ("TMNone")

"Insert no transition contour segment."



Motion Path

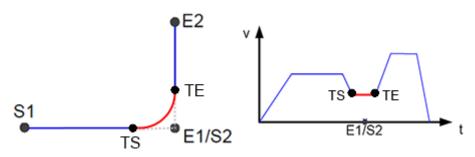
Motion Velocity Profile

The motion blocks are not modified and no transition curve is inserted using this mode. This is the only possible transition mode for the "Buffered" buffer mode.

No transition parameters are used for this transition mode.

12.1.2.4.2 Corner Distance ("TMCornerDistance")

"Transition with given corner distance."



Motion Path

Motion Velocity Profile

The corner distance transition mode is specified using <code>TMCornerDistance</code> as the transition mode for a motion block.

TransitionParameter 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).

- Line-to-Line transitions will shorten the next move by the corner distance. See Line to Line Transitions for more information.
- Line-to-Arc and Arc-to-Line transitions shorten the linear move. See Line-to-Arc and Arc-to-Line Transitions for more information.
- Arc-to-Arc transitions will shorten the arc with the larger radius by the corner distance. See Arc-to-Arc Transitions for more information.

12.1.2.4.3 Related Functions

MC_MoveCircAbs (Function Block)

- MC_MoveCircRel (Function Block)
- MC_MoveLinAbs (Function Block)
- MC_MoveLinRel (Function Block)

12.1.2.4.4 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 (except as noted below), allowing for a smooth transition (see n-Degree Transition below). 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.

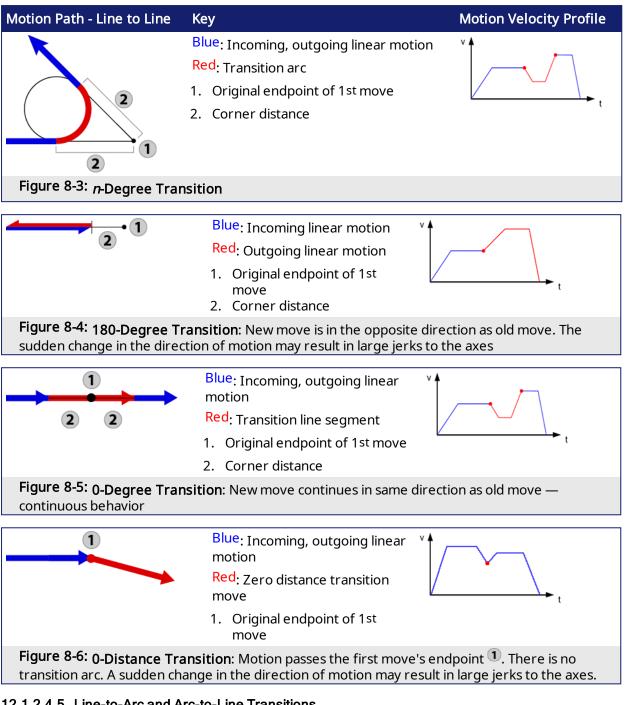
NOTE

588

There are three special cases for line to line transitions:

- Zero-degree transitions: The path will remain unchanged but a linear transition move with the specified transition velocity will be inserted.
- 180-degree transitions: The current move will be shortened by the corner distance when the next move travels in the opposite direction. If the buffer mode is "Buffered" then the path velocity will go to zero. All other buffer modes may cause a large acceleration and jerk.
- Zero-distance transitions: A zero distance transition move will be inserted, which will only affect blending.

If the buffer mode is "Buffered" then the path velocity will go to zero. All other buffer modes may cause a large acceleration and jerk.



12.1.2.4.5 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 (except as noted below), allowing for a smooth transition (see n-Degree Transition below). 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.

NOTE

There are three special cases for Line-to-Arc and Arc-to-Line transitions:

• Tangent transitions:

The linear move will be shortened by the corner distance and a linear transition move will be inserted to cover the distance that was removed. • Intersection transitions: This will only when the arc intersects the line corner distance away from the point where the line and the arc meet. The transition move will be zero distance. If the buffer mode is "Buffered" then the path velocity will go to zero. All other buffer modes may cause a large acceleration and jerk. Zero-distance transitions: A zero distance transition move will be inserted, which will only affect blending. If the buffer mode is "Buffered" then the path velocity will go to zero. All other buffer modes may cause a large acceleration and jerk. Motion Path - Line-to-Arc / **Motion Velocity Profile** Key Arc-to-Line Blue: Incoming, outgoing linear and circular motion Red: Transition arc Original start/endpoint of linear move 2. Corner distance Figure 8-7: *n*-Degree Transition Blue: Incoming, outgoing linear and (1 circular motion Red: Transition line segment 2 1. Original start/endpoint of linear move 2. Corner distance Figure 8-8: Tangent Transition: Line is tangent to the arc Dashed: Original shape Blue: Incoming, outgoing linear and circular motion Red: Zero distance transition 1. Original endpoint of 1st move 2. Corner distance Figure 8-9: Intersection Transition: The line intersects the arc a "corner distance" away from the beginning of the new move. Blue: Incoming, outgoing linear and circular motion Red: Zero distance transition 1. Original endpoint of 1st move Figure 8-10: 0-Distance Transition: Special behavior for 0-distance transitions.

12.1.2.4.6 Arc-to-Arc Transitions

In arc-to-arc transition mode the transitions are handled as follows.

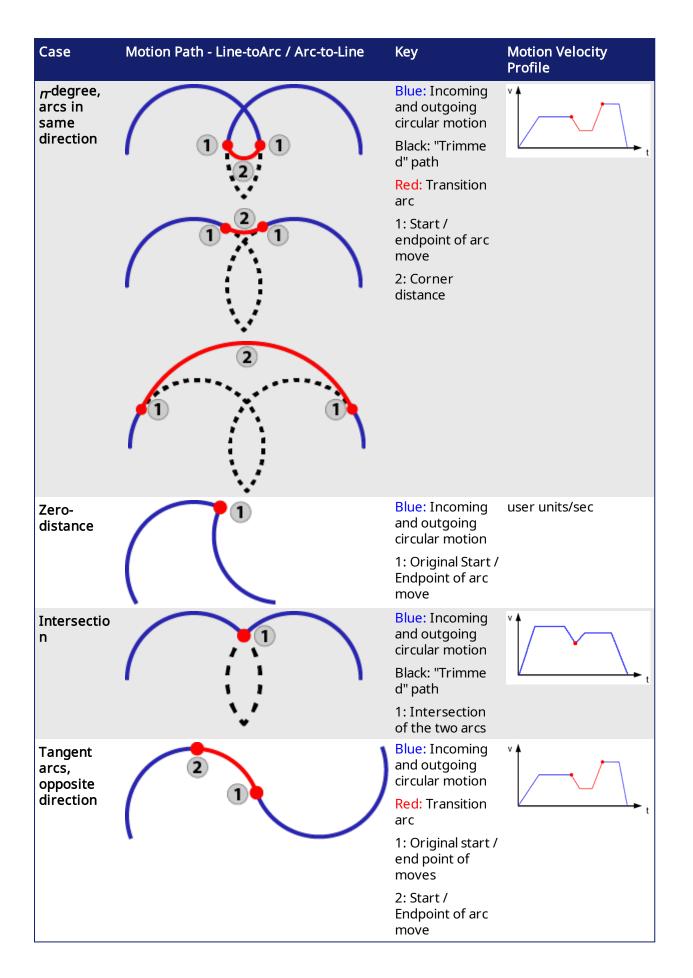
- 1. The arc with the larger radius is shortened by an arc length equal to the corner distance.
- 2. Using this point, a transition move is calculated such that it is tangent to both arcs.
- 3. The arc with the smaller radius is shortened to the point where it is tangent with the transition move.

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.

There are six special cases for Arc-to-Arc transitions.

Transition Type	Description
Zero-distance Transitions	A zero-distance transition move is inserted when the corner distance is zero, which will only affect blending. The path velocity will go to zero if the buffer mode is "Buffered". All other buffer modes may cause a large acceleration and jerk.
Intersection Transitions	This will occur 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. The transition move will be zero distance. The path velocity will go to zero if the buffer mode is "Buffered". All other buffer modes may cause a large acceleration and jerk.
Tangent Transitions with Opposite Direction	A portion of the arc with the larger radius will be replaced by an arc whose length is equal to the corner distance. This will not affect the path, but will affect blending. If the arcs have the same radius, the incoming arc will be treated as having a larger radius.
Line Segment Transitions	For certain values of corner distance, the transition arc has an infinite radius and a line segment is used instead.
Same Circle, Same Direction Transitions	A transition arc with a length of twice the corner distance will be added if both arcs lie on the same circle and are in the same direction. This will not affect the path, but will affect blending.
Same Circle, Opposite Direction Transitions	The arcs will be shorted by an arc length equal to the corner distance and a zero distance transition will be inserted if both arcs lie on the same circle and are in the opposite direction. The path velocity will go to zero if the buffer mode is "Buffered". All other buffer modes may cause a large acceleration and jerk.

Case	Motion Path - Line-toArc / Arc-to-Line	Кеу	Motion Velocity Profile
<i>n</i> -degree, arcs in opposite		Blue: Incoming and outgoing circular motion	
direction	Black: "Trimme d" path	t t	
		<mark>Red:</mark> Transition arc	
		1: Start / endpoint of arc move	



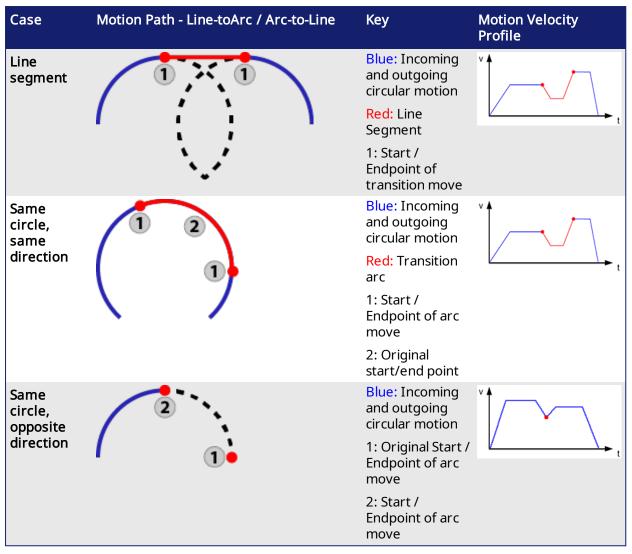


Figure 8-11: Examples of Arc-to-Arc Transitions

12.1.2.5 What Does MC_GrpHalt Do?

MC_GrpHalt (Execute, AxesGroup, Deceleration, Jerk) performs a controlled motion stop of all axes in a coordinated motion group. Differences between MC_GrpHalt and MC_GrpStop.

12.1.2.5.1 MC_GrpHalt Application Example

The following example demonstrates a linear coordinated move with a starting point of (0,0) and an ending point of (200, 0). The first and second MC_GrpHalt commands are called while the linear coordinated move is still moving to the endpoint. As noted 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 end point 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 the section Create a Linear or Circular Coordinated Motion Application.

```
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,
    MC COORDINATE SYSTEM ACS, MC BUFFER MODE ABORTING, MC TRANSITION
MODE 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;
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,
         MC COORDINATE SYSTEM ACS, MC BUFFER MODE ABORTING, MC
TRANSITION MODE NONE, TransParam);
    Inst MC GrpHalt1(FALSE, Group1 ref, default deceleration, default
jerk);
END IF;
```

When MC_GrpHalt is called in the example above, the current move will be aborted and a controlled motion stop will be applied to axes group 'Group1_ref'. The deceleration value, 'halt _deceleration' is set to 50.0 user units/sec2 on the first call and 200.0 user units/second2 on the second call. Jerk is currently not supported.

When MC_GrpHalt is called, the deceleration value from the function block is applied to the path velocity until it reaches 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 zero. Any coordinated moves in the buffer are flushed, and new coordinated moves can be queued up upon completion of the MC_GrpHalt command.

NOTE

This behavior is different than the MC_GrpStop command. For differences between MC_GrpStop and MC_GrpHalt, see "Differences between MC_GrpHalt and MC_GrpStop" on page 598.

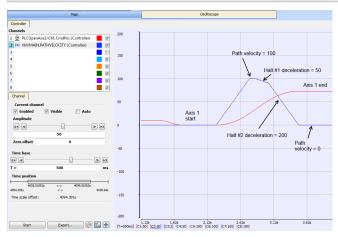


Figure 8-12: Oscilloscope Representation of linear coordinated move with a MC_GrpHalt command called twice

Exceptions:

- The deceleration rate from the MC_GrpHalt function block is only applied to the absolute and relative coordinated motion moves. Direct coordinated motion moves use the default deceleration value as defined by the AxisRef.
- A MC_GrpHalt cannot occur if a group is not enabled.
- A 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 zero before the end of the *Active* move, the path velocity will continue to reach zero during the *Next* move.
- The deceleration rate will be increased if there is only one *Active* coordinated move and the path velocity cannot reach zero before the endpoint. This will prevent overshooting the endpoint.
- A MC_GrpHalt command does not prevent single axis motion from being performed.

12.1.2.6 What Does MC_GrpStop Do?

MC_GrpStop (Execute, AxesGroup, Deceleration, Jerk) performs a controlled motion stop of all axes in a coordinated motion group. Differences between MC_GrpHalt and MC_GrpStop.

12.1.2.6.1 MC_GrpStop Application Example

The following example demonstrates a linear coordinated move starting point of [0,0] and ending at point of [200, 0]. MC_GrpStop is called while the linear coordinated move is still moving to the endpoint. As noted in the oscilloscope, the path velocity reaches zero (approximately [100,0]) before the linear coordinated move reaches its end point value.

Linear Move Parameters	MC_GroupStop Parameters
Velocity = 100	stop_velocity = 75
Acceleration = 200	
Deceleration = 200	
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 the section Create a Linear or Circular Coordinated Motion Application.

```
Inst MC GrpReadCmdPos( TRUE, Group1 ref, MC COORDINATE SYSTEM ACS,
CmdPositionArray );
Inst MC GrpReadCmdVel2( TRUE, Group1 ref, MC COORDINATE SYSTEM ACS,
VelocityArray);
CASE MC MoveCounter OF
0: // Enable the group
Inst MC GrpEnable(TRUE, Group1 ref);
Inst TON3 ( true, t#1500ms ); // Allow for turning on the scope
IF ((Inst TON3.Q) and (Inst MC GrpEnable.Done OR Inst MC
GrpEnable.Error)) THEN
     Inst TON3(false, t#1s);
     IF (Inst MC GrpEnable.Error) THEN
          PrintMessage( LEVEL INFO (*DINT*), 'MC GrpEnable failed -
ErrorID: ' + any to string(Inst_MC_GrpEnable.ErrorID));
     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,
     MC_COORDINATE_SYSTEM_ACS, MC BUFFER MODE ABORTING, MC TRANSITION
MODE NONE, TransParam);
Inst TON3 ( true, t#600ms ); //Allow for the move to reach path velocity
before calling MC GrpStop
IF 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))
```

596

```
THEN
    IF Inst_MC_GrpStop.Error THEN
        PrintMessage( 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,
        MC_COORDINATE_SYSTEM_ACS, MC_BUFFER_MODE_ABORTING, MC_TRANSITION_
MODE_NONE, TransParam);
    Inst_MC_GrpStop(FALSE, Group1_ref, stop_deceleration, default_jerk
);
END_IF;
```

When MC_GrpStop is called in the example above, the current move will be aborted and a controlled motion stop will be applied to axes group 'Group1_ref'. The deceleration value, 'stop_deceleration', is set to 75.0 user units/sec2 and is applied to the path velocity until it reaches zero. Jerk is currently not supported.

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 zero 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, and new coordinated moves can be queued up upon completion of the Stop command

NOTE

This behavior is different than the MC_GrpHalt command. For differences between MC_GrpHalt and MC_GrpStop, see "Differences between MC_GrpHalt and MC_GrpStop" on page 598.

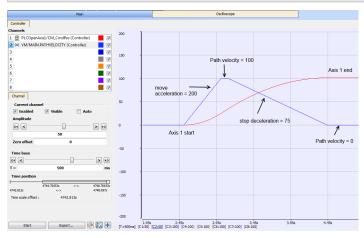


Figure 8-13: Oscilloscope Representation of linear coordinated move with a MC_GrpStop

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.

- A MC_GrpStop command *cannot* be aborted by any other commands (while MC_GrpHalt can be).
- When there are two coordinated motion moves (*Active* and *Next*) in the queue, and the path velocity does not reach zero before the end of the *Active* move, the path velocity will continue to reach zero during the *Next* move.
- The deceleration rate will be increased if there is only one *Active* coordinated move and the path velocity cannot reach zero before the endpoint. This will prevent overshooting the endpoint.
- A MC_GrpStop command does not prevent single axis motion from being performed.

12.1.2.7 Differences between MC_GrpHalt and MC_GrpStop

While MC_GrpHalt and MC_GrpStop both perform a controlled motion stop of all axes in an axes group, some differences exist between the operation of the function blocks.

- MC_GrpStop can not be aborted by any other command. MC_GrpHalt can be aborted by another MC_GrpHalt command, a MC_GrpStop command or a MC_Move command such as MC_ MoveLinAbs, MC_MoveCircRel, etc.
- While MC_GrpStop is active, the group state (see Group State Diagrams) is 'GroupStopping'. While MC_GrpHalt is active, the group state is 'GroupMoving'.
- When MC_GrpHalt is complete, the group state goes to 'GroupStandBy'. When MC_GrpStop is complete, the DONE output will be true. The EXECUTE input must be set to false before the group state can go to 'GroupStandBy'.

12.1.2.8 Handling Axis Errors

Coordinated Motion Error handling is configurable on a per axis group basis. When a PLCopen axis error occurs the Default Behavior is for all axes in the group stop. This means motion interpolation stops, active and next queues are cleared, and the group state "GroupErrorStop" is enabled. Additionally, the position loop on the drives in the group are opened and the drives are disabled.

The Optional Behavior 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 will be disabled. The application is then 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 just having one axis stop.

To configure the optional behavior, use the function block MC_GrpWriteBoolPar, setting the parameter 'IGNORE_AXIS_ESTOP' (1000) to TRUE. To check if the optional behavior has been set, call the function block MC_GrpReadBoolPar, reading 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

12.1.2.8.1 Default Behavior

In the default motion error handling case, a user can detect errors using any of the following group function blocks:

- MC_GrpReadStatus Output GroupErrorStop is true if a group error has occurred.
- MC_GrpReadError Error output will be true and Error ID output will be 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 will be set to 12. In addition, MC_ReadStatus for each axis in the group can be called where the output ErrorStop will be 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 will be TRUE if all axis errors have been reset. This function block may take up to 3 seconds to reset some error conditions.

12.1.2.8.2 Optional Behavior

🗙 TIP

This is configurable on a per group basis.

In the optional motion error handling case, the group status GroupErrorStop will not be 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 deteced using the following group function block:

• MC_GrpReadBoolPar — Read the result of parameter ID 'AXIS_ESTOP_ACTIVE' (1001). If the result is TRUE, an axis error exists. Note that the group function blocks used for the default case of detecting errors (MC_GrpReadStatus and MC_GrpReadError) will 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 will be set to 12. In addition, MC_ReadStatus for each axis in the group can be called where the output ErrorStop will be 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 will not yet be reset when this function returns. MC_ReadStatus will still have to be called to verify that the drive error has been resolved.

12.1.2.8.3 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 due to 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 will try 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. For 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 will 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 will abort the current operation and take control of the axes that can still be controlled. To determine which axis has faulted, the MC_ReadStatus() FB can be 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.

When the error recovery is completed and the faulting axis restored to normal operation, in order to restart Coordinated Motion, an MC_GrpSetPos() command must be issued to tell the Coordinated Motion Engine where the axes in the faulted Axes Group are then at, 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 will enter 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.

12.2 Motion Techniques

This chapter explains advanced concepts and procedures related to **motion techniques** that are possible with the KAS IDE.

12.2.1 PLC Online Change

() IMPORTANT	
You have to save 💾 and compile 🛃 your project before doing an online change.	

This section provides a detailed description of the PLC Online Change functionality. See Using PLC Online Change for an overview of using this functionality.

12.2.1.1 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 as shown in the figure below; and then ask the controller to switch the execution to the new application.

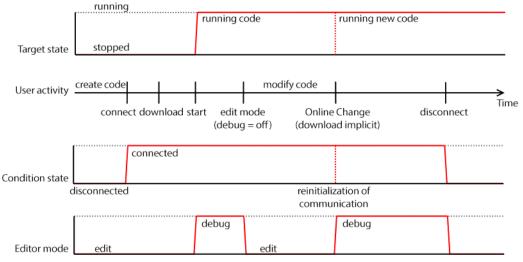


Figure 8-14: Online Change - Process Diagram

🗙 TIP

This capability applies only to PLC code. This is not supported in the PipeNetwork 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 an INFO level message as follows: Online Change done in X μ s. For more details, click here 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 the following situations:

- **Development phase**: you can modify the application and apply these modifications incrementally without stopping the controller
- **Update in production**: you can update the running motion application (for instance with a bug fix release) without stopping the whole production chain

When Online Change is enabled, you can perform the following 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
- The following are not allowed:
 - 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 (but 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 (for more details, as well as workaround, Pulse Limitations with Online Change).

The WAIT and WAIT_TIME instructions must not be used
 ① 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 need to explicitly insert a variable in the loop.

① IMPORTANT

When Online Change is active and custom libraries are being 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. Turn the Online Change back on if desired

NOTE

602

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 is displayed.

For limitation about breakpoint with Online Change, see "About Online Change" on page 221.

About the states and transitions

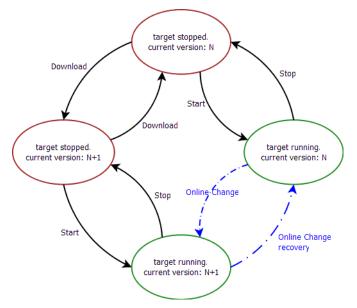


Figure 8-15: Online Change - States and Transitions

12.2.1.2 How to Activate Online Change

To allow Online Change, you need to open the PLC options and set the relevant parameters.

PLC options
Compilation
Release Debug
Code generation: Normal
Online Change
Online Change Enabled
PLC Variable Locking
Allow Variables to be Locked
OK Cancel

Figure 8-16: PLC Options - Online Change Enable

Set the parameters

This box allows you to enable or disable the **Online Change** feature (see call out (1)).

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 the two following 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
- Note: Check the Controller Log window for any errors that occur.

You also need to ensure that you have selected **Normal** code ⁽²⁾ as Online Change is not possible with **Optimized** code. Note that when **Optimized** code is selected, then Online Change is always deactivated.

Then you can compile your application, which now allows future changes on the fly.

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 Toolbar.

Perform the Online Change

604

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).

Online Change Progress	
Downloading new code.	
	30%

Figure 8-17: 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.

Dictionary behavior

When the Online Change is enabled, the dictionary shows:

- new variables in blue
- deleted variables in red

ictionary				
Controller:PLC Track Selection				
🝸 Name	Туре	Dim.	Attrib.	Init value
ResetBut	BOOL			
OpenBut	BOOL			
CloseBut	BOOL			
StartMove	BOOL			
Cnt	UINT			UINT#0
_del_Incr	UINT		Deleted	
Profiles	ProfilesCo			
PLCopen	PLCopenC			
EtherCAT	EtherCAT		Read Only	
MyNewV	BOOL		Added	

Figure 8-18: Online Change - Dictionary

🗙 TIP

The deleted variables can be for new variables.

12.2.1.3 What is the Revert button

The **Revert** button is for security purposes. It allows you, after an Online Change, to revert your change quickly and go back to the previous application. That means switching the execution of the controller to the P-code that was running before the last Online Change (note that the source code in the KAS IDE is not replaced). The WAIT and 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 user must manually reconnect to the controller.

NOTE

You can go back to the previous version only when the Online Change feature is activated and while the controller is not stopped.

① 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

12.2.1.4 Difference between Local and Controller versions

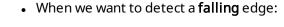
When you restore a project with the Revert feature after an Online Change, KAS provides a tool to show the differences between two versions of the project. This tool can help you in checking all modifications before the next Online Change. It is also a useful tool when you want to compare your code with the last version after a Revert.

For more details, refer to Compare PLC Programs.

12.2.1.5 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:



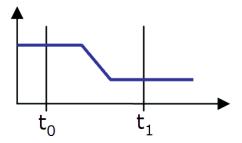


Figure 8-19: Pulse Limitations with Falling Edge

• When we want to detect a **rising** edge:

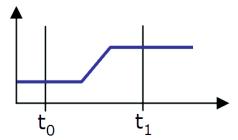


Figure 8-20: Pulse Limitations with Rising Edge

🗙 TIP

If you want to avoid this limitation, you must use declared instances of R_TRIG and F_TRIG function blocks.

NOTE

This limitation is temporary and is going to be fixed in a future release.

12.2.2 Using PLC Online Change

This section provides an overview of how to use Online Change. See PLC Online Change for descriptions of the functionality.

12.2.2.1 Set up 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. Compile the project 🗾,

12.2.2.2 Enable Online Change

1. Select the PLC Options button from the tool bar.

PLC options PLC aptions
Compilation
Release O Debug
Code generation: Normal 👻
Online Change
Online Change Enabled
OK Cancel

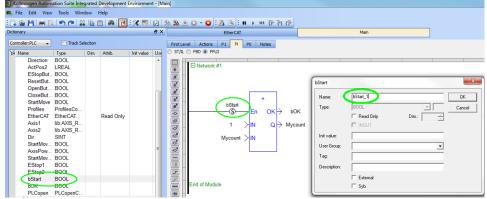
- 2. Enable Online Change.
- 3. Compile the project 🛃, connect 🌇 and download 👛 the application to the device.
- 4. Start executing the application.

12.2.2.3 Using Online Change

Enable the Toggle Edit/Debug mode button in the tool bar. Note that application variables will not be updated, even though the application is running.

Local: UnnamedProject:3 Ortroller: UnnamedProject:3 Drives inactive Running

2. Modify a local or global variable name in the Dictionary.



3. Compile and click the PLC Online Change download button from the tool bar to download the changes to the controller. Note that the status bar has changed from "Running" to "Paused" during the download, and back to "Running" once the download has finished. The applications variables should be updating.

NOTE

If Online Change is disabled while the application is running, even temporarily, downloading will fail.

12.2.2.4 Revert Online Change

- 1. Click the PLC Online Change Revert button 츠 from the tool bar.
- 2. A message should be displayed stating that revert was successful.
- 3. Note the reverted state of the application in the IDE.
- 4. Connect to the device. 嵡
- S. Note the state of the application and the device.
 Local: UnnamedProject:4
 Controller: UnnamedProject:3
- 6. From this state, Online Change can only be used again after ...:
 - 1. Stopping the application
 - 2. Making changes
 - 3. Recompile
 - 4. Download the application to the controller

12.2.3 What Are Fast Inputs?

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. Thanks to the precise timing of external events, an application can improve its control algorithm, resulting in higher operating performance. Fast (or high-speed) inputs are digital inputs of a drive that are configured to latch the time at which they are triggered.

Drives inactive

Paused

Not Connected

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 has two capture engines which can be freely linked to any input. These high speed inputs can be used in application which, when triggered, caused 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.

★ TIP

When using S300 or S700, Fast input has to be enabled by setting the drive keywords IN1MODE to 26 and IN2MODE to 26. This can be achieve using DriveGUI configuration tool.

See the following sections for fast input information specific to the motion engines.

12.2.3.1 Fast Inputs with Pipe Network Motion

★ TIP

For the Axis and Trigger Pipe Network blocks, refer to MLAxisCfgFastIn which 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)
MLTrigSetEdge	MLAxisRstFastIn
MLTrigIsTrigged	MLAxisIsTrigged
MLTrigReadPos	MLAxisCfgFastIn

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 (for more details on this concept, see "Distributed Clock (Synchronization)" on page 435). 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" on page 456). However, when working with KAS, time latching is more useful, because the positions of all the drives in the application can then be interpolated by means of the Trigger block with the MLTrigReadPos function block.

12.2.3.2 Fast Inputs with PLCopen Motion

For PLCopen, refer to MC_TouchProbe, which arms a Fast Input and returns the latched position when the Fast Input event occurs.

Other PLCopen function blocks which arm a Fast Input and return the latched position to the function block when the event occurs are shown below.

Registration operation	Homing operation
MC_MachRegist	MC_Reference
MC_MarkRegist	MCFB_StepAbsSwitchFastInput
	MCFB_StepLimitSwitchFastInput

Table 8-2: PLCopen function blocks which arm Fast Inputs and return the latched position.

12.2.4 Torque Feed-forward

The torque feed-forward tells the controller what forces is required to move the axis in an arbitrary trajectory.

Here 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 feedforwards 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 feedforward. However, the amount of misadjustment in the torque feed-forward necessary to cause such instability is very rare.

12.2.5 PLCopen Homing

12.2.5.1 PLCopen Homing Description

The homing features provided in PLCopen create tools for homing of PLCopen axes. Homing may be performed utilizing the MC_Reference function block, utilizing Custom Homing Library UDFB's or by writing your own homing cycles.

• Utilizing MC_Reference

The application specifies a position for an axis to be assigned to a reference position, 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 desired reference position and the captured actual position, the coordinate system is shifted to correlate the desired reference position to this location.

- Writing your own homing cycles UDFBs can be written to provide specific "canned" homing cycles based on feedback type, and desired homing sequences such as 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.
- Utilizing Custom Libraries
 A library already contains a set of homing UDFB. Contact the Support for more information.
 To add the library to your project, refer to Create and Use Custom Libraries

12.2.5.2 PLCopen Homing Methods

The following common homing methods (among others) can be performed in PLCopen. This section details the setting of the ADK parameters and the PLCopen function blocks to accomplish these methods.

PLCopen does not limit you to these methods, as the capture engine is very configurable.

12.2.5.2.1 Home using 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.

12.2.5.2.2 Find Input

Homing using a drive input is accomplished by configuring the AKD capture engine, and then using the MC_Reference function block. The following capture engine parameters need to be configured, along with the following 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".

12.2.5.2.3 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.

- 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.

12.2.5.2.4 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".

12.2.5.2.5 Find Index

Homing using a drive index pulse is accomplished by configuring the AKD capture engine, and then using the MC_Reference function block. The following capture engine parameters need to be configured, along with the following 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.

12.2.5.3 AKD Capture Engine Configuration

The AKD capture engine provides a broad range of capabilities for configuration of the capture event (s). Furthermore, 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. 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

The following section details the configuration parameters for the ADK capture engines.

- 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 edge capture edge is programmed in the MC_Reference function block.
 - 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

For more details, refer to CAP0.PRESELECT, CAP1.PRESELECT section.

- 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
- 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 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

12.2.6 Pipe Network Homing

UDFBs can be written to provide specific "canned" homing cycles based on feedback type. Contact the Support for more information.

12.2.6.1 Fast Homing Example with the Pipe Network Motion Engine Axis Pipe Block

This use case 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.

A Setup sequence example.

- 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. Therefore the drive must be configured in order 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
- Note that 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 and its value is explained in the section How To Interpret a Timestamp.

- 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

The following 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;

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 the following Kollmorgen KUDFBs in the IDE Function Block Library: MLFB_HomeFindHomeFastInput, MLFB_HomeFindHomeFastInputModulo, MLFB_HomeFindLimitFastInput, and MLFB_HomeFindLimitFastInputModulo.

The Axis Position can also be derived from the Time Stamp by with the following 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*TimeS
tamp+PreviousPosition;
```

12.2.7 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. There are two basic forms of registration: single-axis registration and master/slave registration.

12.2.7.1 Deciding Which Function Blocks to Use for Registration

There are two good methods by which 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 Details	MC_TouchProbe, Trigger_Ref structure, and MC_ MoveSuperimp This method incorporates:	MC_MarkRegist or MC_MachRegist, Trigger_Ref structure, MC_MoveSuperimp, and MC_ ReadParam 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. 	 MC_MarkRegist or MC_MachRegist, which in addition to capturing a position from a drive automatically calculates the position correction and can either make the offset move (linear acceleration / deceleration) or 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.	Incorporates additional functionality including:
	 Fast to develop due to having fewer input values to determine . Simple function block functionality leads to less debug time. 	 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	https://www.kollmorgen.com/en- us/developer- network/registration-plcopen- example-project/	https://www.kollmorgen.com/en-us/developer- network/registration-plcopen-example-project/

12.2.7.2 Single-Axis Registration

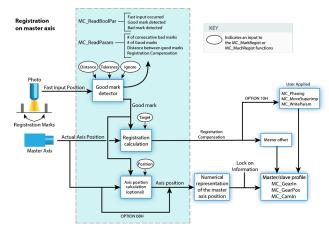
Single-axis registration is performed on an axis running a discrete move such as MC_MoveAbsolute or MC_MoveRelative. When the Fast Input latches the position of the product, the axis position is reset, typically to zero. This resets the axis's coordinates for each product to accommodate for variations in the distance between products and keep the process synchronized to the product over many repetitions.

12.2.7.3 Master/Slave Registration

Master/slave registration is performed on an axis running a master/slave move such as MC_GearIn or MC_CamIn. 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 and keep the process synchronized to the product over many repetitions.

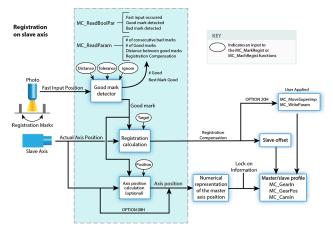
12.2.7.3.1 Master Registration

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.



12.2.7.3.2 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.



The figure Registration below shows an example of a printing application using 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.

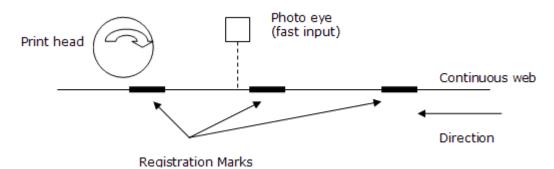


Figure 8-21: Registration

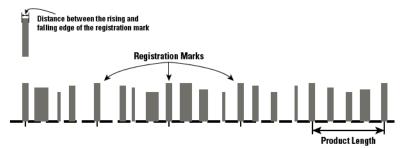
12.2.7.4 Registration Application Guide

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.

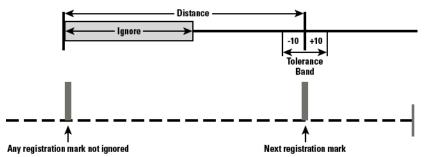
Many factors can contribute to inconsistency. Some examples are:

- 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.



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 and Mark to Machine Registration.

12.2.7.4.1 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.

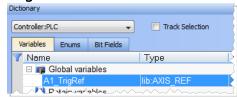
12.2.7.4.2 Mark to Machine Registration

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.

12.2.7.5 Fast Homing Example 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. For this example, the name A1_ TrigRef is used.



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 the following Kollmorgen KUDFBs in the IDE Function Block Library MCFB_StepAbsSwitchFastInput and MCFB_StepLimitSwitchFastInput.

12.2.8 Pipe Network Registration and Fast Homing

12.2.8.1 Fast Homing Example with the Pipe Network Motion Engine Axis Pipe Block

This use case 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.

A Setup sequence example.

- 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. Therefore the drive must be configured in order 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
- Note that 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 and its value is explained in the section How To Interpret a Timestamp.
 - 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

The following 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;
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 the following Kollmorgen KUDFBs in the IDE Function Block Library: MLFB_HomeFindHomeFastInput, MLFB_HomeFindHomeFastInputModulo, MLFB_HomeFindLimitFastInput, and MLFB_HomeFindLimitFastInputModulo.

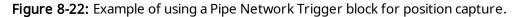
The Axis Position can also be derived from the Time Stamp by with the following code:



12.2.8.2 Registration Position Capture Example 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.





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 the following parameters.

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 the following constants: MLFI_FIRST or MLFI_SECOND for the trigger block to be triggered on the arrival of the first or the second input respectively.
TRIGGER_MODE	Indicates if the trigger block responds to the rising edge or the falling edge of the Fast Input Specify one of the following constants: MLFI_RISING_EDGE or MLFI_ FALLING_EDGE

Figure 8-23: Configuration of the Trigger block

After configuring the Trigger block, the order of calls to its motion library functions is as follows:

- 1. MLTrigWriteDelay(DINT,TriggerID,LREAL,delay)
 - This function sets the delay compensation, typically starting 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

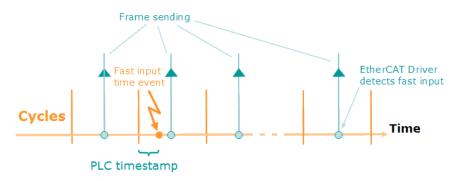
You have to correct the position by taking into account the delay due to the number of cycles needed to read the timestamp of the Fast Input. You can use MLTrigWriteDelay to address this issue.

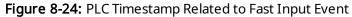
- 6. MLTrigReadTime(PipeNetwork.TRIGGER1)
 - This function returns the time associated with the Fast Input as explained in How To Interpret a Timestamp.
 - Note that this function is of lesser importance compared to the previous one.
- 7. MLTrigClearFlag(PipeNetwork.TRIGGER1)
 - This function rearms 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.

12.2.9 How To Interpret a Timestamp

The timestamp is based on the EtherCAT system time. For this value to make sense, distributed clock must be activated in the drive (see Distributed Clock tab) and in the EtherCAT master.





The timestamp returned is relative to the beginning of the cycle in which the Fast Input is triggered. It is called PLC timestamp since it is the value that you can use in PLC programs.

12.2.10 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.

For 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 can be 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 being changed, such as during an upgrade from AKD to AKD2G drives, then the parameters will need to be retuned to maintain optimal performance.

Feature	Parameters	Read Me For More Information
PLCopen	Axis parameters MC_AXIS_PARAM_VELCOMP_	Axis Parameters
Camming	FACTOR, MC_AXIS_PARAM_VELCOMP_FILTER	Velocity Compensation
PLCopen Gearing		
PLCopen TouchProbe	Axis parameter MC_AXIS_PARAM_SENSOR_DELAY	MC_TouchProbe and Time- Based Capture (KDN Article)
PLCopen Registration		PLCopen Registration
PipeNetwork External Feedback	Use a Phaser block to compensate for the delay	Motion Library - Phaser
PipeNetwork 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). For more details see ____.

12.2.11 Error Management

When a non-fatal error occurs and motion must be stopped quickly, the following procedure can be taken:

For each axis:

Step	Example Application Code
Send Stop Command for each axis	MLAxisStop(PipeNetwork.AXI_A1_Axis, TRUE, DEF_ A1_StopDec);
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);

For the machine:

Step	Example Application Code
Stop Command at the master block level	MLMstRunMLMstRun(PipeNetwork.MASTER, 0.0);
Wait for Master command to be stopped	IF A1_AckState = DEF_StateErrorStop AND A2_Ackstate = DEF_StateErrorStop AND MLBIkIsReady(PipeNetwork.MASTER) THEN PrintF('*** ErrorStop M1=%i ***', M1_ StatusWord,0,0,0); M1_AckState := DEF_StateErrorStop;

NOTE

This code is only applicable to PipeNetwork applications. It will not work in a PLCopen application.

This procedure for error management is based on the **Project Structure Guidelines** as described in Application Software Structure - Implementation (-> p. 683)

For information on **restarting the motion**, refer to Restarting Motion with PipeNetwork (> p. 625)

12.2.12 Restarting Motion with PipeNetwork

An advantage of the Pipe Network is the ability 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	<pre>AxisStatus := MLAxisStatus(PipeNetwork.AXI_A1_Axis); IF AxisStatus.6 THEN StepCounter := 1; END_IF;</pre>
Turn axis back on (re- enable)	<pre>IF MLAxisPower(PipeNetwork.AXI_A1_Axis, PowerUp) THEN StepCounter := 2; END_IF;</pre>
Calculate position difference between the Reference and Actual Positions	<pre>DeltaPos := (MLAxisCmdPos(PipeNetwork.AXI_A1_Axis) - MLAxisReadActPos(PipeNetwork.AXI_A1_Axis));</pre>
Determine how far to move	<pre>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;</pre>
Wait for move to be completed	<pre>IF MLAxisReAlgnRdy(PipeNetwork.AXI_A1_Axis) THEN StepCounter := 4; END_IF;</pre>

For the machine:

Step	Example Application Code	
Execute multi-axis	MLMstRun	
move	MLMstRun(PipeNetwork.MASTER, 500);	

12.2.13 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. Up to four different Superimposed Axes can be superimposed upon a Receiving Axis.

12.2.14 Overtravel Conditions

The PLCopen and PipeNetwork motion engines can respond to overtravel conditions from the AKD and AKD2G drive family. As a safety feature, AKD and AKD2G drives ignore motion commands once an overtravel is detected in the direction of a hardware limit switch. The PLCopen or PipeNetwork motion engine applies a stop action to axes when these overtravel conditions occur.

NOTE

Generation 1 AKD drives require setting a fieldbus parameter to enable this behavior. See Configuring Generation 1 AKD Drives below.

12.2.14.1 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).

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.

12.2.14.2 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 drives 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.

12.2.14.3 PipeNetwork Behavior

A PipeNetwork 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 will no longer be accepting pipe positions. The pipe that the axis is connected to will continue 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 the page Restarting Motion with PipeNetwork to understand how to accomplish this.

NOTE

If a hardware limit flag is being triggered at the program startup by a PipeNetwork 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.

12.2.14.4 Configuring Generation 1 AKD Drives

When using generation 1 AKD drives, bit 14 of the parameter FBUS.PARAM05 (see FBUS.PARAM1 TO FBUS.PARAM10 must be set high to allow KAS IDE to respond to the hardware limit switches,

If FBUS.PARAM05 originally had a value of 0, typing FBUS.PARAM05 16384 would turn on bit 14. Remember to save your drive parameters to flash memory once bit 14 is set, otherwise the behavior will turn off once the drive is power cycled.

12.2.15 How To Create a Digitizing Axis

Digitizing axes are supported by PLCopen-based projects. Following is the general process. The specifics will vary depending upon your EtherCAT network.

1. From the Project View, right-click on the *Controller* > *Motion* > *PLCopen* item and select **New Axis**.

Project View		
Name		Туре
🗸 📷 System	1	Sinope
🗸 💼 Co	ntroller	[10.50.67.76] PDMM or PCMM
> 💽	PLC	
× 🐻	Motion	
	📷 Profiles	
~	PLCopen	P
	✓ Axes	New Axis
	🙀 PLCOpenAxis1	2
	PLCOpenAxis2	
	Axes Groups	
0.	PDMM Onboard I/O	
	Ether AT	

- 2. Give the axis a name, note that spaces are not supported. See Common Axis Parameters for details.
- 3. Digitizing axes have *Bus* data which needs to be defined. See Digitizing Axis Parameters for details.
- 4. Lastly, define the Axis Data Parameters.

The following list provides more detail about each method or circumstance under which a digitizing axis may be created.

- How To Create a Digitizing Axis When EtherCAT Devices Are Defined
- How To Create a Digitizing Axis When EtherCAT Devices Are Not Connected
- How To Create a Digitizing Axis When Using an Imported ENI File
- How To Create a Digitizing Axis From Code

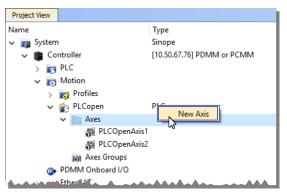
Also see Working With A Digitizing Axis in PLCopen for more in-depth information about digitizing axes.

12.2.15.1 How To 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.

KD Drive
KD Drive
KD2G Dual Axis SIL2

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 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 for details.

Axis Name: DigitizingAxis	s Type: Digitizing V	
Number: 3		
EtherCAT Bus		
Device: AKD2G_1	 Address: 1002 	
Feedback Settings		
CoE Object: Positio	n actual value - Axis 1 🗸 🗸	
CoE Object Index: (hex) 0x 6064 CoE Object Subindex: 0		
CoE Object Index: (h	nex) 0x 6064 CoE Object Subindex: 0 🌩	
	nex) 0x 6064 CoE Object Subindex: 0 🗘	
	er Type: Singleturn Absolute	
Encode	er Type: Singleturn Absolute	
Encode Single-turn position bit	er Type: Singleturn Absolute	
Encode Single-turn position bit Digitizing Axis	r Type: Singleturn Absolute	
Encode Single-turn position bit Digitizing Axis Axis Data	r Type: Singleturn Absolute v count: 2 Degree (motor shaft) v	
Encode Single-turn position bit Digitizing Axis Axis Data Position Unit:	r Type: Singleturn Absolute count: 2 • : Degree (motor shaft) 360.0 deg / revolution	

4. Lastly, define the Axis Data Parameters.

12.2.15.2 How To Create a Digitizing Axis When EtherCAT Devices Are Not Connected

This process assumes that you are setting up a project and are not connected to the EtherCAT network. In this example we have added several devices to the EtherCAT branch of the Project View, including an AKD2G, an AKD-C, and two AKD-Ns.

➡ EtherCAT			
🖉 AKD2G_1	AKD2G Dual Axis SIL2		
🖉 AKD_C_1	AKD-C String 1		
🖉 AKD_N_1	AKD-N		
🖉 AKD_N_2	AKD-N		

1. From the Project Tree, right-click on the *Controller* > *Motion* > *PLCopen* item and select **New Axis**.

Project View		
Name		Туре
🗸 📷 System	ı	Sinope
🗸 🍘 Co	ntroller	[10.50.67.76] PDMM or PCMM
> 💽	PLC	
× 📷	Motion	
~	PLCopen	P
	✓ Axes	New Axis
	🚮 PLCOpenAxis1	~
	🙀 PLCOpenAxis2	
	M Axes Groups	
0 *	PDMM Onboard I/O	
	Ether AT	

- 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 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 for details.

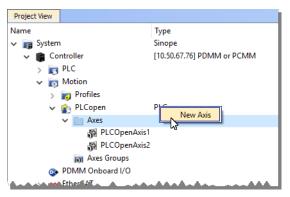
Axis Name: Digitiz	tingAxis	т	ype: Digitiz	ina	~
Number: 3			yper bigitz		
EtherCAT Bus				_	
Device: AKD	2G_1		\sim	Address: ()
Feedback Setting	js				
CoE Object:	AIN2.VALUE				~
		[2724]		ubindex:	1
CoF Object In	dex:(hex)()x	3/81			
CoE Object In			-	don dext	
-		Singleturn Absolu	-	~	
-	Encoder Type:		-	~	
-	Encoder Type:	Singleturn Absolu	-	~	-
Single-turn posit	Encoder Type:	Singleturn Absolu	-	~	
Single-turn posit Digitizing Axis Axis Data	Encoder Type: ion bit count:	Singleturn Absolu	-	~	-
Single-turn posit Digitizing Axis Axis Data Positic	Encoder Type: ion bit count:	Singleturn Absolu 16 🜩	-	~	volution
Single-turn positi Digitizing Axis Axis Data Positic User	Encoder Type: tion bit count: on Unit: Degr r Units: 360.	Singleturn Absolu 16 🜩	ute	~	

4. Lastly, define the Axis Data Parameters.

12.2.15.3 How To Create a Digitizing Axis When Using an Imported ENI File

This process assumes that you are setting up a project when using an imported ENI file. See ENI File Tab for information about using ENI files.

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 for details.
- 3. Digitizing axes have *Bus* and *Feedback Settings* data which needs to be defined. The ENI file will provide some information about the digitizing axis but you will need to provide the **Address** of the device to use as well as the **CoE Object Index**, **Subindex**, and the **Encoder Resolution**. See Digitizing Axis Parameters for details.

EtherCAT	🔍 PLCopen Axis Data 🛛 🕹
Devices Master ENI File ESI Files EtherCAT Network Information	Axis Name: DigitizingAxis Type: Digitizing ~ Number: 3
Import / Export	EtherCAT Bus Device: Address: 3
Export ENI File	Feedback Settings CoE Object: CoE Object Index: (hex) 0x 2050 CoE Object Subindex: 0
~	Encoder Type: Singleturn Absolute
	Digitizing Axis Axis Data
	Position Unit: Degree (motor shaft) User Units: 22.5 deg / revolution
	Update Rate: 250 usec Rollover Position: 0 deg
	OK Cancel

4. Lastly, define the Axis Data Parameters.

12.2.15.4 How To Create a Digitizing Axis From Code

Following is a method for creating 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 );
```

632

12.2.16 Working With A Digitizing Axis in PLCopen

This topic provides the basics of setting up a digitizing axis and describes how to accomplish some common tasks. See Common Axis Parameters for reference material.

• A Digitizing Axis requires a feedback position. The feedback position can come from a secondary feedback of a drive or an EtherCAT device. Typically the AKD or S300/S700 drives provide the feedback position via the CoE object **0x2050**.

	puts			
Index	Subindex	Object Name	Size [bit]	PLC Variable
0x2050	0	Position actual val	32	(Global)/ActPosA1_Feedback2
0x20	0	Latch status word	16	
0x20	0	Latch position	32	
0x3470	4	AIN.VALUE	16	☞ Managed in I/O tab
0x6041	0	Status word	16	
0x6063	0	Position actual val	32	
0x606C	0	Velocity actual val	32	
0x6077	0	Torque actual value	16	
0x60F4	0	Following error	32	
0x60	0	Digital inputs	32	Managed in I/O tab

• The Digitizing Axis is created by setting the Type to "Digitiizing", selecting the EtherCAT device which will provide the feedback, and setting the correct CoE object.

🔳 PLCopen Axis Data		×
Axis Name: DigitizingAxis Number: 3	Туре:	Digitizing -
Bus Device: AKD_1	▼ Address: 1001	
Feedback Settings		
CoE Object:	Position actual value 2	-
CoE Object Index: (hex) 0x	2050	CoE Object Subindex: 0
NOTE		

NOTE

First generation AKD drives will use certain pre-set values.

The actual position in user units can be read with the MC_ReadActPos function block.

	M	C ReadActPos	
bMotionEngineStarted	Enable	– Valid	ReadActPosValid2
	Enable	valiu	ReadActPosBusy2
Axis2	Axis	Busy	
			ReadActPosError2
		Error	
			0
		ErrorID	ReadActPosErrorID2
			127.617980
		Position	→ ActPos2

• The position can be offset using the MC_SetPos function block.

h Daardi kara Daa		MasterPos— _SetPos]	
b_ResetHomePos	Execute	Done		-0
Axis2 300.0	Axis	Busy		—O——I
A2_ResetPos	Position	Error		— <u> </u>
A2_RestPosMode	≻ <mark>Mode</mark>	ErrorID	\rightarrow	ResetPosError

For AKD drives, the polarity of the feedback signal can be changed using drive parameter FB2.DIR.

Parameters				
This page lists the current	values of all parameters	on the device.		
Full Name	Value Units	Parameter	Read/Write 🗶	
FB2 counting direction (impact	0	FB2 DIR	read-write	
Position Feedback Resolution	1,000	FB2.ENCRES	read write	
Feedback 2 Mode	0 - Input - A/B Sig	FB2.MODE	read-write	
FB2.MOTORPHASE	0 deg	FB2.MOTORPHA	read-write	
FB2.MOTORPOLES	6	FB2.MOTORPOL	read-write	
Feedback 2 scaled position		FB2.P	read-only	
NOTE	·····•••••••••••••••••••••••••••••••••		rr Virita	
If an Endat 2.2 device is order to tie the PDO ind				

12.2.16.1 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.

Parameters						
This page lists the current values of all parameters on the device.						
Full Name	Value	Units	Parameter 🔀	Read/Write		
- Fault Actions						
Gets/Sets the Resulting Action for Fa.	1 - Ignore (fault will not be reported)		FAULT130.ACTION	read-write		
Gets/Sets the Resulting Action for Fa.	0 - Disable Amplifier		FAULT131.ACTION	read-write		
Gets/Sets the Resulting Action for Fa	0 - Disable Amplifier		FAULT132.ACTION	read-write		
Gets/Sets the Resulting Action for Fa	0 - Disable Amplifier		FAULT134.ACTION	read-write		
Gets/Sets the Resulting Action for Fa	0 - Disable Amplifier		FAULT139.ACTION	read-write		
Gets/Sets the Resulting Action for Fa	0 - Disable Amplifier		FAULT451.ACTION	read-write		
Gets/Sets the Resulting Action for Fa	4 - Generate Warning (motion will con.		FAULT570.ACTION	read-write		
Petr Sets the Resulting Action for Fa.	A-Pisable Amplify		FAULT7PRACTION	rez 4-write		

12.2.17 Practical Applications - Camming

The following topics provide procedures you may find useful when programming camming.

12.2.17.1 Positioning 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 also: Cam Profile Switching

12.2.17.2 Resuming 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).

12.2.17.3 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.

The following steps will guide you through creating and building a new Cam Profile on the fly.

- 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 PipeNetwork 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

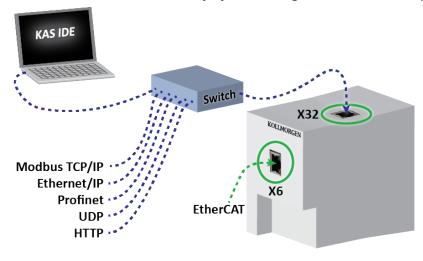
If a cam profile is Active, by having called MC_CamTblSelect and then MC_CamIn in PLCopen or MLCamSwitch in PipeNetwork, it cannot be released. In this situation use a second non-active cam to release, build (or rebuild), then activate.

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.

12.3 Motion Bus and Fieldbuses

Depending on the fieldbus used in your project (EtherCAT, Profinet, etc.), you have to make use of the configuration tools discussed below. The AKD PDMM or PCMM can be connected to multiple Ethernet devices simultaneously by connecting a switch to the X32 port on the top of the device.



★ TIP

See Performance Guidelines for information on how the CPU load is affected when transmitting and receiving data between network devices and PLC variables

12.3.1 EtherCAT

- For configuration, Configure EtherCAT Motion Bus
- For I/O mapping, Map Input and Output to Variables
- For error management, EtherCAT Error Messages

See the Beckhoff Web site for EtherCAT XML Device Description (http://www.beckhoff.se/english.asp?download/elconfg.htm).

12.3.2 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. The mapping of PLC variables to Profinet is as simple as a drag and drop.

- Setting up Ethernet/IP IO Scanner (Client)
- Setting up Ethernet/IP Adapter (Server)
- Setting up Ethernet/IP Tag Client

12.3.3 Modbus & TCP/IP

• Setting up Modbus Slave

12.3.4 Profinet

This fieldbus is Profibus over ethernet. We define one controller and have 1 or more devices.

- For configuring the controller, see Profinet IO RT Controller Configuration .
- For configuring devices, see Profinet IO RT Device Configuration.

12.3.5 Add Unsupported EtherCAT Device

NOTE

This procedure is for advanced users only

When your project contains EtherCAT devices that are not supported by KAS, you have to create the configuration with an external tool, and perform the following steps:

- 1. Get the AKD device description ESI file from the official AKD distribution
- 2. Ensure all the device description ESI files are available for the external tool
- 3. Use the external tool and do all the configuration, including the following points:
 - Set the Cycle Time
 - Turn on the distributed clocks option for all slave drives in order to share a global system time through EtherCAT
 - Assign PDO to each drive (inputs and outputs)
 - Set the mode of operation of the drives into position mode
 - Insert variable names and do the mapping (see details below)
- 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 your project

① IMPORTANT

Importing an external ENI file overrides all EtherCAT project device information and configuration settings in the IDE. The following 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.

12.3.5.1 How to modify the EtherCAT image in cyclic mode

In your application program, when integrating non-standard EtherCAT devices, use the following function blocks to update EtherCAT frame:

- ECATWriteData(Function)
- ECATReadData(Function)

12.3.5.2 How to configure EtherCAT device

You need to use the following Functions Blocks:

- ECATWriteSdo (Function Block)
- ECATReadSdo (Function Block)

12.3.5.3 How to 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 the following convention:

<pre>@Scope.Varia</pre>	ableName <mark>+</mark> StartBit <mark>-</mark> 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) ProgramName Note that even for the case of nested child SFC programs, the variables still belong to a unique well defined subprogram Parentheses must surround the scope when it is Global or Retain.
+StartBit	<i>(Optional)</i> Integer that defines the bit from which the data must be written or read from the PLC variable
-Size	<i>(Optional)</i> 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.</bitsize>

Examples:

```
(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**.

12.3.6 Set Up FSoE Master and an AKD2G with SafeMotion Monitor

This topic describes how to manually set up FSoE PDOs and copy links for a FSoE Master and an AKD2G with SafeMotion Monitor. This example uses a simple EtherCAT topology consisting of a PCMM (the **Controller**), a BBH SCU safety master (**Device_1**), and an AKD2G with SMM Basic safety drive (**AKD2G_1**).

★ TIP

If using a BBH SCU-1-EC and SafePLC2 software, then the FSoE PDOs can be automatically setup by the KAS IDE and SafePLC2. For more details see Adding or Importing a SafePLC2 Project. For all other third party FSoE Masters, use the procedure described below.

12.3.6.1 Create Safety Application

First, use the 3rd party Safety PLC software to create a safety application for the FSoE master. For example, the application will contain a FSoE master and an AKD2Gwith Axis1 and Axis2 as slave devices.

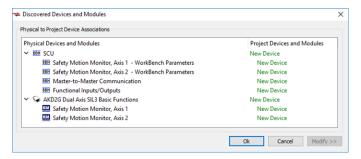
- 1. Set the AKD2G Connection IDs to match the the AKD2G SafeIDs.
- 2. Compile and download the safety application to the FSoE Master.
- 3. Run the safety application in FSoE Master.

The state will be "Run Intern" and the FSoE state will be "8".

12.3.6.2 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 PCMM controller is set in the project.
- 2. Scan the EtherCAT network to discover the devices.

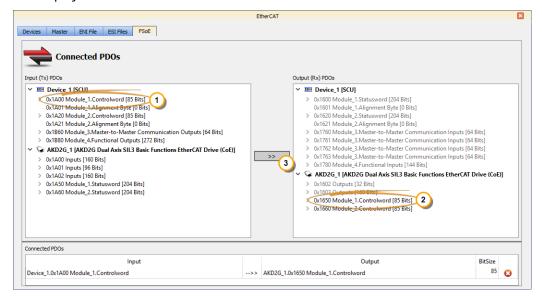


- 3. We are prompted to recompile the project due to changes in the EtherCAT topology. We respond "No" because we have not set up the PDO copy links.
- 4. To connect the PDOs we open the FSoE / PDO Connections Tab which is populated with Input and Output PDOs.



evices Master ENI File ESI Files FSOE Connected PDOs	
pput (Try) PDOS V EM Device_1 [SCU] > 0x1A00 Module_1.Controlword [85 Bits] 0x1A20 Module_1.Alignment Byte [0 Bits] 0x1A20 Module_2.Controlword [85 Bits] 0x1A20 Module_3.Master-to-Master Communication Outputs [64 Bits] 0x1B80 Module_3.Master-to-Master Communication Outputs [64 Bits] 0x1A80 Inputs [160 Bits] 0x1A00 Inputs [160 Bits] 0x1A01 Inputs [160 Bits] 0x1A01 Inputs [160 Bits] 0x1A03 Module_1.Statusword [204 Bits] 0x1A60 Module_2.Statusword [204 Bits] 0x1A60 Module_2.Statusword [204 Bits]	Output (Rx) PDOs Will Device_1 [SCU]
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.



- 6. We make the following 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.
- 8. The BBH master has changed the state to "Run" and the FSoE state will be "4".

The Safety Master and the Safety Slave are now exchanging data over EtherCAT.

See Also:

- AKD2G Safety Parametrization Using FSoE with SCU-1-EC and PCMM/AKD PDMM
- FSoEParamsInit
- Troubleshooting FSoE Safety Parameters

12.3.7 AKD2G Safety Parametrization Using FSoE with SCU-1-EC and PCMM/AKD PDMM

The safety features in the AKD2G safety drive (Basic and Advanced) can be activated through either FSoE (Fail Safe over EtherCAT) or digital input or both. The source of activation is configured inside the safety parameters and it needs to match the system requirements.

This document describes how to configure and enable safety functions through BBH's SafePLC and KAS IDE applications.

12.3.7.1 Requirements

- Hardware SCU-1-EC (BBH FSoE Master)
 - PCMM or AKD PDMM
 - AKD2G safety drive
- Software SafePLC2 (BBH Safety Software)
 - KAS IDE v3.05 (or newer)

12.3.7.2 How-To

Following are the general steps 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 configure via WorkBench" module, based on your application need.
- 4. When using the "SRA Parameters" profile, add the FSoEParamsInit function block to your 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 the steps listed in Connecting to a SafePLC2 Project Direct Method if SafePLC2 is installed on the same computer or Connecting to a SafePLC2 Project Import Method if it is installed on a different computer than the KAS IDE.
- 6. Compile and download the SafePLC application to the BBH safety master (SCU-1-EC) and run the application.
- 7. Compile the KAS IDE project.
- 8. Run the compiled KAS application.
- 9. Execute the FSoEParamsInit function block when the application is ready and EtherCAT is in OP state.

See Also:

- Working with SafePLC2 Projects
- Set Up FSoE Master and an AKD2G with SafeMotion Monitor
- FSoEParamsInit
- Troubleshooting FSoE Safety Parameters

12.3.8 Fieldbus Editor

12.3.8.1 PCMM/AKD PDMM Ethernet/IP Data Communications

The following applies to both the Scanner and Adapter

- Up to 4 different connections:
 - One Adapter Connection with one O -> T Assembly and one T -> O Assembly
 - Up to 3 Scanner Connections each with one O -> T Assembly and one T -> O Assembly
 - Total: Up to 8 Assemblies (4 O -> T and 4 T -> O) total
- Connections can be:
 - All to one product (with unique IP Addresses)
 - Or up to 4 products (with 4 unique IP Addresses)
- Each Assembly can transfer up to 500 bytes
- PxMM max communications: 8 x 500 = 4000 bytes
- Assemblies and Connections can be mixed in any suitable manner. Examples:
 - Set up the PCMM/AKD PDMM to talk to 4 different Adapters with each one having a O -> T Assembly and T-A Assembly
 - Set up the PCMM/AKD PDMM:
 - as a Scanner (Client) to connect to 3 different Ethernet/IP Servers (Adapters) each with one O -> T Assembly and one T -> O Assembly
 - and also set up as a Server (Adapter) to connect to a Scanner (Client) with one O -> T Assembly and one T -> O Assembly

12.3.8.2 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.

12.3.8.2.1 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. Then select the Ethernet/IP IO Scanner in the configuration selector



The configuration is represented as a tree:

- Ethernet/IP IO Scanner
 - Server (an Ethernet/IP adapter device) (*)
 - IO Assembly (Originator to Target)
 - Exchanged Variable (*)
 - IO Assembly (Target to Originator)
 - Exchanged Variable (*)

(*) The items with this mark can appear several times in the configuration.

In addition to the IO configuration, explicit messaging can be performed using the blocks eipReadAttr and eipWriteAttr.

12.3.8.2.2 Configuration of the Server

Click the Insert Master icon 🚠 to declare a server (slave adapter). Each server is identified by its IP address and an optional Description text.

Server	X
Address:	127 . 0 . 0 . 1
Description:	Home
Configuration	1
Instance	: 3 Edit
	OK Cancel

Three 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.

12.3.8.2.3 Editing the Configuration Data Block

While most devices do not expect any configuration it may happen that some data is required. For this, double click on the server in the tree and press the **Edit...** button in the dialog box.

The configuration is entered as a list of items. Use the **Type** button to specify the number of items and their data types. Structured data types defined in the KAS IDE are supported. Simply enter values in the grid. In case you exchange multiple byte integers (e.g. WORD) you must specify whether the device expects little or big endian formatting.

Configuration da	ita				×
Туре:	(none)		Nb. items:		✓ Little endian
Item		Value	Description		
	Type Type: Nb. items:	(none) 1	ОК	Cancel	
				ОК	Cancel

12.3.8.2.4 Configuring an Assembly

Click the Insert Slave icon 🔳 to declare a CIP I/O assembly. Each assembly is identified by:

Identifier	Meaning
Туре	Direction of the I/O assembly. Can be one of:
	Originator to Target (outputs)Target to Originator (inputs)
Instance	Instance of the CIP assembly

Identifier	Meaning	
Config.instance	Type Instance Size Connection t Priority 32 bit header RPI (ms) Description I/O: Inputs (Ta 100 2 Point to point Low 100 Target To Ori I/O: Outputs (101 2 Point to point Low 100 Originator To	
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 or 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 (range is 10 - 10000 ms)	
Description	Optional description text	

12.3.8.2.5 Configuring Variables

IEC61131-3 variables may be mapped on the data of the assembly. For each variable you must specify:

Identifier	Meaning
Symbol	The name of the IEC61131-3 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	Kind of data exchanged through the variable:
	 <i>Data Exchange</i>: a type of input or output data in the assembly <i>Server OK</i>: indicates the status of the IP connection to the server <i>I/O connection OK</i>: indicates the status of the CIP I/O connection <i>Last UCMM error</i>: 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.

12.3.8.3 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.

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. Then select the Ethernet/IP Adapter in the configuration selector

Add Configuration	— ×
Choose a configuration	ОК
. Ethernet/IP	Cancel
 Ethernet/IP Adapter (server) 	
- Ethernet/IP Tag Client	
FlexIO / PointIO	
Profinet IO	

The configuration is represented as a tree:

- Ethernet/IP IO Scanner
 - Served I/Os and objects
 - IO Assembly or Vendor Specific Object (*)
 - Exchanged Variable (*)

(*) The items with this mark can appear several times in the configuration.

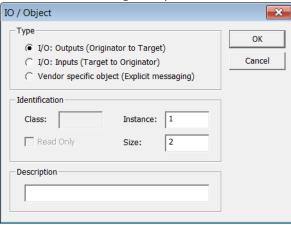
Configuration

The following 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:

646

Identifier	Meaning
Mode	Kind of CIP object. Can be one of:
	 I/O assemblyVendor specific object

Identifier	Meaning		
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.		
	Settlemet/IP Adapter (server) Name Value Mode I/Os (Implicit Messaging) Access I/O: Outputs (Originator to target) Image: Class Instance Instance 1 Symbol Offset Bit Format Range (Low) Range (High) Signal (Low) Signal (High) Signal (High)		
Instance	Instance of the CIP assembly or object		
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, the following attributes are available for scanners:

1 (get only) = size of the object data

3 (get/set) = object data

Then you can map IEC61131-3 variables on the data of the assembly, for each variable you must specify:

Identifier	Meaning
Symbol	The name of the IEC61131-3 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

You can 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.

12.3.8.4 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.

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. Then select the Ethernet/IP Tag Client in the configuration selector

Add Configuration	×
Choose a configuration	ОК
 (All) Ethernet/IP Ethernet/IP I/O Scanner (client) 	Cancel
Ethernet/IP Tag Client	
Profinet IO	

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.

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 IEC61131-3 application. Configuration of the tag and mapping of all array items is performed automatically.
- Alternatively you can click the Insert Slaver icon 🔳 to declare the tag and map some variables later on.

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 (note that 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) SINT (8 bit signed integer) INT (16 bit signed integer) 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.

648

Identifier	Meaning
Period(ms)	You can specify in this parameter a period for continuously sending the request. Enter "0" for a request sent "on demand"
Timeout	Request timeout in milliseconds

IEC61131-3 variables are mapped on the data of the tag, for each variable you must specify:

Identifier	Meaning
Symbol	The name of the IEC61131-3 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	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: will be 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 will be 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.

12.3.8.5 PDMM/PCMM EDS file for EtherNet/IP

The KAS installation contains an EtherNet/IP EDS file for the Kollmorgen controller. This file may be needed by other controllers, (PLCs, PCs etc.) to configure the EtherNet/IP communication with a AKD PDMM.

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:

```
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 up to 4 input and output assemblies when KAS Runtime is configured as adapter (server), up from 1.

Table 8-3: Revision history for AKD PDMM EDS file for EtherNet/IF

12.3.8.5.1 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 according to the user's need. 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. For example, if 40 bytes is sent O->T and 64 bytes is sent T->O, two 64 byte values must be selected.

Edit Device Parameters			
Parameters			
Parameter Name	Value		
Data sizes			
0001 O->T size (Exclusive Owner)	64 Bytes		
0002 T->O size (Exclusive Owner)	64 Bytes 💌		
	1 Byte		
	2 Bytes		
	4 Bytes		
	8 Bytes		
	16 Bytes		
	32 Bytes		
	64 Bytes		
	128 Bytes		
	256 Bytes		
	500 Bytes 👻		
0002 T->O size (Exclusive Owner) Default : 4 Bytes variable size for Consuming			

Figure 8-25: Example of setting byte sizes in a third-party Network Configurator.

	FieldBus Editor*		
E	Ethernet/IP Adapter (server)	Name	Value
品	🖮 🐰 Served I/Os and objects	Mode	I/Os (Implicit Messaging)
_	0 31 [i/o] 1 [64]	Access	I/O: Inputs (Target to originator)
	ī=0 [i/o] 2 [64]	Class	
8		Instance	2
		Size	64
_		Description	
			-
dankin.	and a second sec	and the second second	and the second sec

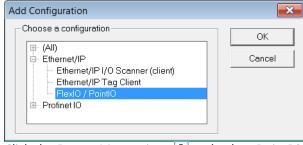
Figure 8-26: Example of setting byte sizes in the KAS IDE Fieldbus Editor

12.3.8.6 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. Then select the FlexIO/PointIO driver in the configuration selector



4. Click the Insert Master icon 📠 and select PointIO

Server	X
Туре:	PointIO
Address:	192 . 168 . 0 . 34
Description:	MyServer Point ID
[OK Cancel

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.

12.3.8.7 Modbus Slave

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" on page 121). Variables defined in the HMI to describe the interface (see Map Variables to HMI) are passed to the AKD PDMM/PCMM this way. 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. As this data has a rather low priority and is interpreted by human feedback, it should never be noticed by the user.

NOTE

Please note that 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. An example of this would be having two separate HMI panels running the same program on a single, large machine.

12.3.8.7.1 Protocol specification

The protocol supported is Open Modbus on Ethernet. The following Modbus function codes are supported:

1	read coils
2	read bit inputs
3	read holding registers

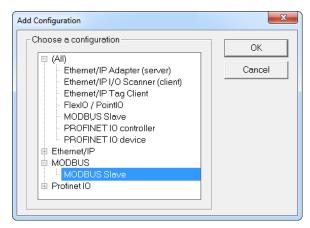
4	read input registers	
5	write 1 coil	
6	write 1 register	
15	write <i>n</i> coils	
16	write <i>n</i> registers	

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.

12.3.8.7.2 Data exchange - configuration

A dedicated configuration tool is integrated in the KAS IDE. To run it,

- 1. Select the **Other Modbus devices** option in the Controller Properties dialog box. See Configure the Controller 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. Then select the **Modbus Slave** in the configuration selector



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.

12.3.8.7.3 Modbus Slave configuration

You need to configure the Modbus Slave in order to make variables visible from external Modbus masters such as SCADA systems. Below is a simple example of slave configuration:

	MoDBUS Slave Holding Registers [16] - Holding Registers Holding Registers [16] - Holding Registers O-1: ModbusCheck/LREAL_Modbus O-1: ModbusCheck/DINT_Modbus O-4-5: ModbusCheck/UDINT_Modbus O-** Coil Bits [11] - Coils O-** Input Registers [11] D-** Input Bits [11]	
--	--	--

Double click on 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. The following kinds of block are available:

- Input Bits: bits read by external masters (function 2).
- Coil Bits: bits forced by by external masters (function 5 or 15).
- Input Registers: words read by external masters (function 4).
- Holding Registers: words forced by external masters (function 6 or 16).

★ TIP

Input Registers require 15-25%less CPU time to exchange data than **Holding Registers** do. For optimal controller performance use Input Registers where possible.

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 will be denied if the range specified in the request does not fit within a data block defined in the configuration. Requests overlapping two data blocks will be denied.

For 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 will be denied and an "address error" exception will be 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.

- For exchanging Boolean variables through Modbus words, a hexadecimal mask is available in order to define to which bit of a word a variable is attached. For example, enter the mask "0001" to map a Boolean variable to the less significant bit of a word.
- For exchanging 32 bit variables (DINT, REAL...), you can select to map the variable on two consecutive words.

You can sort the variables of each data block according to their offset using the 'Sort symbols' menu command at any time.

12.3.8.7.4 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.

- For exchanging Boolean variables through Modbus words, a hexadecimal mask is available in order to define to which bit of a word a variable is attached. For example, enter the mask "0001" to map a Boolean variable to the less significant bit of a word.
- For exchanging 32 bit variables (DINT, REAL...), you can select to map the variable on two consecutive words.

NOTE

• 64 bit variable (LINT and LREAL) cannot be extracted directly without lost of accuracy or data.

12.3.8.8 Profinet IO RT Controller Configuration

The KAS IDE contains a fully integrated configurator for Profinet IO RT Controller.

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. For example, 7 slots containing 16 outputs of 32 bits each, would be 7 * 16 * 32 = 3584 bits = 448 bytes.

IMPORTANT

Referring to the Profinet standard, the units of a Profinet network are named as IO Controllers (Masters) and IO Devices (Slaves).

🗙 TIP

To use some of the additional features of the Profinet controller field bus editor, such as "Browse Network for Slaves", you need to install the "CD Prot" driver. The installer can be found in <Kollmorgen Automation Suite installation directory>\Bin\CDProtDriver.

Please note that your computer will need to be rebooted after installing the driver.

NOTE

Profinet is only supported on AKD PDMMs and PCMMs.

NOTE

Unlike an EtherCAT fieldbus which has 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*, so frequent HTTP or IDE Oscilloscope communication can cause delay in the Profinet communications, and ultimately Profinet frame loss.

12.3.8.8.1 Configuration

The I/Os of the Profinet network must be connected to the variables via a Profinet IO controller. Start the declaration of a Profinet controller with:

1. Insert > Insert Master/Port.

The following window opens.

Properties	Value	ОК
Name	undefined	Cancel
Adapter identification		
Activity timeout factor [100ms]	50	Help
Priority Tagging	No (Ethernet 802.3)	riop
Send clock factor [31,250us]	32	
Watchdog factor	3	
Alarm timeout [100ms]	50	
Alarm retries	3	
IP configuration delay [ms]	500	
Do not wait until connect has been completed		
Start despite disconnected slaves		
I/O communication only from within cycle		

These parameter can be changed:

Parameter	Meaning
Name	 A device name can consists of labels and 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 [az09-] Labels do not start with [-] Labels do not end with [-] The first label does not start with "port-xyz" or "port-xyz-abcde" with a,b,c,d,e, x, y, z = 09 Device names do not have the form n.n.n.n, n = 0999 Labels do only start with 'xn-' if RFC 3490 is applied
Adapter identification	MAC address of the Profinet IO controller or name of the network connection (Windows XP or younger and Windows CE 4.1 or younger). This must be set to tsec0.
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 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 HTTP file transfer occurs between the AKD PDMM or PCMM 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).

Parameter	Meaning
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 hat been competed	On: the application starts immediately. Off: the application starts after all devices are connected.
Start despite disconnected slaves	On: the application starts despite configured but not found devices. Off: the application starts if all configured devices were found only.
I/O communication only from within cycle	On: Run I/O communication from within VM-cycle Off: Run I/O communication outside VM-cycle.

2. Mark the controller and click Insert > Insert Slave/Datablock to add a Profinet IO device.

Properties	Value	OK
Name	<undefined></undefined>	
Adapter MAC address	00:00:00:00:00	Cancel
VLAN ID	0	
IP address	0.0.0.0	Load GSDML file
Subnet mask	0.0.0.0	
Default gateway	0.0.0.0	
Object UUID	dea00000-6c97-11d1-8271-000000000000	
Reduction ratio	1	
Phase	1	
Module IDs		
Module IOs		
Module configuration		
Prefix		

These parameter can be changed:

Parameter	Meaning
Name	 A device name can consists of labels and 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 [az09-] Labels do not start with [-] Labels do not end with [-] The first label does not start with "port-xyz" or "port-xyz-abcde" with a,b,c,d,e, x, y, z = 09 Device names do not have the form n.n.n.n, n = 0999 Labels do 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

Parameter	Meaning
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<br="">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). With the default setting of 16 the data transfer rate is 48ms; for 2 it is 2ms; Most devices support Reduction ratio settings of 1,2,4,8,16,32</reduction></send>
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> * <Reduction ratio> * <Send clock factor> * 31,250 μ s you get the time that may expire between two frames until the device is set back. I.e. for the default settings 3 * 32 * 31,250 μ s the connection will be 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. I.e. the connection will be reset after 24 * 32 * 31,250 μ s = 768 ms. If this data exchange rate is to low please separate the PROFINET IO network from the office network (e.g. by a router).

3. Click Load GSDML file to import the necessary GSDML file.

Properties	Value	OK
Name	750-340 V00.70.00	Cancel
Adapter MAC address	00:00:00:00:00	Cancel
VLAN ID	0	
IP address	0.0.0.0	Load GSDML file
Subnet mask	0.0.0.0	
Default gateway	0.0.0.0	Configure slots
Object UUID	dea00000-6c97-11d1-8271-000102ee011d	
Reduction ratio	1	
Phase	1	
Module IDs	{0000 01000154 [0001 00000000]}	
Module IOs	{0000 [0001 01 01 0000 0000]}	
Module configuration	{0000 [0001 (0080 00000010 00 00 00 00 00 00 cb 43 00 00 00 00 00 00 00 00]}	
Prefix		
Vendor name	WAGO Kontakttechnik GmbH u. Co. KG	
Slave family	I/O - WAGO-I/O-SYSTEM 750/753	
Slave name	750-340 V00.70.00	

4. Do the slot configuration after the GSDML file import.

-v.db(able modules:						
Nan	ne			Module B	Allowed in slots	Decription	ОК
16 D	I			0×00009	101 1-63	16-Channel Digital Input Module; Representatives of Series 750 and 753:	
16 D	0			0×00009	1-63		Cance
2 AI				0×00000	F3 1-63	2-Channel Analog Input Module; Representatives of Series 750 and 753: 452, 454, 456, 461, 462, 465, 466, 467, 469, 472, 474, 4	
2 AC)			0×00000	57 1-63	2-Channel Analog Output Module; Representatives of Series 750 and 753: 550, 552, 554, 556, 560, 585	
2 C1	I/AO, 2 ST/AI			0×00000	57 1-63	2-Channel Analog Output Module; Representatives of Series 750 and 753: 550, 552, 554, 556, 560, 585	
2 DI	(+ 6 PH DI)			0×00008	1-63	2-Channel Digital Input Module; Representatives of Series 750 and 753: 400, 401, 405, 406, 410, 411, 412, 413, 427	
2 DI (+ 6/ 4 PH DI), 2 DO (+ 6 PH DO), DIA Dx000084		1-63	2-Channel Digital Input Module, 1 Bit Diagnosis and 1 Bit Diagnosis Acknowledge per Channel; Representatives of Series 750 and 753				
2 DI	(+ 6/ 4 PH DI), DIA			0x000084	1-63	2-Channel Digital Input Module, 1 Bit Diagnosis per Channel; Representatives of Series 750 and 753: 419, 421, 425	
2 DI	(- 2 PH DI)			0×000083	1-63	2-Channel Digital Input Module; Representatives of Series 750 and 753: 400, 401, 405, 406, 410, 411, 412, 413, 427	
2 DI	(- 2/ 4 PH DI), 2 DO (-	2 PH DO), DIA		80000×0	07 1-63	2-Channel Digital Input Module, 1 Bit Diagnosis and 1 Bit Diagnosis Acknowledge per Channel; Representatives of Series 750 and 755	
2 DI	(- 2/ 4 PH DI), DIA			0×00008	05 1-63	2-Channel Digital Input Module, 1 Bit Diagnosis per Channel; Representatives of Series 750 and 753: 419, 421, 425	
<						>	
Slot	ted modules: Module name	Module ID	Fixed		Parameter data		
Slot			M				Inser
Slot	Module name		N I				Inser
Slot 0 1 2	Module name						Appen
Slot	Module name						
Slot 0 1 2 3 4	Module name						Appen
Slot 0 1 2 3 4 5	Module name						Appen
5lot 0 1 2 3 4	Module name						Appen
5 0 1 2 3 4 5 6 7	Module name						Appen
Slot 0 1 2 3 4 5 6 7 8	Module name						Appen
500 0 1 2 3 4 5 6 7 8 9	Module name						Appen
5lot 0 1 2 3 4 5 6 7 8	Module name						Appen

5. Select the modules in the upper list. With the buttons <code>Insert</code> and <code>Append</code> 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.

(ndex	Parameter record	Parameter	Allowed values	Value	ОК
28	General parameters	Slot Properties		according to device settings	
28	General parameters	Asynch. Diagnosis Message Channel 0		disabled	Cancel
28	General parameters	Substitude Output Value Channel 0		0	
128	General parameters	Asynch. Diagnosis Message Channel 1		disabled	Reset defaults
128	General parameters	Substitude Output Value Channel 1		0	

Figure 8-27: Example of configuring sub-modules.

6. Now you can connect the variables with the I/Os. Use Insert > Insert/Set Variable in order to append a variable to a device.

Properties	Value	OK
Variable name	222	
Area	Input	Cancel
Format	Single bit	
Slot	0	Help
Subslot	0	
Offset	0	
Bit	0	

These parameter 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.

658

Parameter	Description
Format	32 bit float, Signed 16 bit integer, Signed 32 bit integer, Signed 8 bit integer, Single bit, Unsigned 16 bit integer, Unsigned 32 bit integer, Unsigned 8 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.

7. All settings can be changed in the grid too. The information show refers to the items below of the selected item in the configuration tree.

		-	Law				_
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 grid shows the real-time data of the variables.

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. For example, 7 slots containing 16 outputs of 32 bits each, would be 7 * 16 * 32 = 3584 bits = 448 bytes.

12.3.8.8.2 Data types

You can 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.

12.3.8.8.3 Additional features

★ TIP

To use some of the additional features of the Profinet controller field bus editor, such as "Browse Network for Slaves", you need to install the "CD Prot" driver. The installer can be found in <Kollmorgen Automation Suite installation directory>\Bin\CDProtDriver.

Please note that your computer will need to be rebooted after installing the driver.

12.3.8.8.3.1 Browse network for slaves

This command is available in the context menu of the network adapter. All connected Profinet IO devices are listed up.

Device specific	MAC Adr.	Create	Name	IP address	Subnet mask	Default gateway	OK
IM151-3	08:00:06:99:03:F3	~	SIEMENS_1	0.0.0.0	255.255.255.0	0.0.0.1	
WAGO-I/O-SYSTEM 750	00:30:DE:00:41:19	~	WAGO	0.0.0.0	255.255.255.0	0.0.0.1	Cancel
ILB PN 24 DI16 DI016-2TX	00:A0:45:03:99:E4	~	PHOENIX_1	0.0.0.0	255.255.255.0	0.0.0.1	
							Blink
							Help

Select the devices you want to add in the tree via the checkbox in the column Create.

The Network Link (or another) -LED of the marked device blinks for three seconds when pressing the Blink button.

Set the device names (column Name) here. A device name can consist of labels and 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...z0...9-].
- Labels do not start with [-].
- Labels do not end with [-].
- 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 do only start with 'xn-' if RFC 3490 is applied.

Set the IP address of the device here. By clicking into the grid the Default gateway will be set automatically.

Click OK after setting the IP-parameters (address, subnet mask, default gateway).

12.3.8.8.3.2 Configuration of devices

After adding the Profinet devices to the controller they need to be configured. Double click on the referring device. Load the GSDML file. If ambiguous module IDs are found they are shown in a window. Choose the right module in the combo-box.

Slot	Module ID	Module name	^	OK
0	0x00000301	IM151-3 PN (PROFINET I/O device interface module IM 151-3 PN for ET 2005 electronic module, supports packing)		
1	0x00000684	PM-E DC24V (Power module PM-E for electronic modules, DC24V, with diagnostics)		Cancel
2	0x00004021	PM-E DC24V (Power module PM-E for electronic modules, DC24V, with diagnostics)	=	
3	0x00004021	PM-E DC24V (Power module PM-E for electronic modules, DC24V, with diagnostics) PM-E DC24V 5 (Power module PM-E for electronic modules, DC24V, with diagnostics and status)		
4	0x00004021	PM-E DC24V S (Power module PM-E for electronic modules, DC24V, with diagnostics and status)		
5	0x00004014	2DI DC24V ST (Digital input module DI 2xDC24V, Standard)		
6	0x00004014	2DI DC24V ST (Digital input module DI 2xDC24V, Standard)		

NOTE

660

Automatic creation of variables can be done primal after this step.

12.3.8.8.3.3 Set slave station name

With this context menu command it is possible to rename the slave names.

12.3.8.8.3.4 Read module configuration

With this context menu command it is possible to read out the module configuration again.

12.3.8.8.3.5 Create variables

Find this command in the context menu of the device. Based on the defined device modules the referring variables are generated.

Within the same dialog it is possible to define the variables for

- Device diagnosis
- IOxS

12.3.8.8.3.6 Device diagnosis

Based on the Profinet standard the referring variables can be generated:

- CycleCounter [UINT].
- Status [BOOL].
- DataValid [BOOL].
- ProviderState [BOOL].
- StationsProblemIndikator [BOOL].

12.3.8.8.3.7 Create IOxS for slave modules

Based on the defined device modules the referring IOPS- and IOCS-variables are generated.

12.3.8.8.4 How to Resolve Errors

12.3.8.8.4.1 Device is not found

- Check if device is switched on
- Check the network connection
- Ensure the correct name was set on the device

12.3.8.8.4.2 Error setting the IP configuration

- Ensure that the IP configuration is valid and appropriate for your network
- Ensure the VLAN ID is setup correctly in the device settings

12.3.8.8.4.3 Timeout error

- Ensure that the IP configuration is valid and appropriate for your network
- Increase the IP configuration delay

12.3.8.8.4.4 Other errors

- Ensure you use the correct device name
- Ensure you have chosen the correct GSDML file (also check if the GSDML file version is appropriate for the firmware version of the device)

12.3.8.8.4.5 Connect response error

Depending on the PNIO status error check, if the settings listed in the table below are set appropriate for your device.

PNIO status	Check the specified setting
1C010003	Slave\ObjectUUID (correct GDML file)
DB81010A	Master\Activity timeout factor
DB81010B	Master\Controller name
DB81010C	Master\Controller name
DB810207	Master\RT-Class
DB81020A	Master\Send clock factor
DB81020B	Slave\Reduction ratio
DB81020C	Slave\Phase
DB81020F	Master\Watchdog factor
DB810210	Master\Watchdog factor
DB8103,*	Slave\Module IDs,IOs
DB810407	Master\Alarm timeout

PNIO status	Check the specified setting
DB810408	Master\Alarm retries

Table 8-4: PNIO status error codes on connect and the related settings in the configuration

12.3.8.8.4.6 Module configuration is different

- Ensure you have configured the modules present on the device (If you have read the module configuration from the device, ensure you have selected the correct modules in the resolve ambiguous modules dialog displayed after selecting the GSDML file).
- Ensure you have chosen the correct GSDML file (also check if the GSDML file version is appropriate for the firmware version of the device).

12.3.8.8.4.7 Writing parameterization error (with status 0xDF80*)

- Ensure you have configured the modules present on the device (If you have read the module configuration from the device, ensure you have selected the correct modules in the resolve ambiguous modules dialog displayed after selecting the GSDML file).
- Ensure you have chosen the correct GSDML file (also check if the GSDML file version is appropriate for the firmware version of the device).

If your error could not be resolved using the hints above, update the firmware of your device and the GSDML file to the latest version and try again.

For further analyses record the PNIO communication using Wireshark (http://www.wireshark.org/).

- Start Wireshark and select Capture\Options from the menu.
- Choose the network interface used for the PNIO communication and enter the Capture filter below:

ether proto 0x8892 or (ether proto 0x8100 and ether[16:2]=0x8892) or udp port 0x8894

- Then click start.
- Start the runtime with your application several times and then stop the Wireshark capturing by selecting Capture\Stop from the menu. You can save your recorded traffic by selecting File\Save from the menu.

12.3.8.8.5 Coding of PNIO status

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 PNIO status has a value of zero.

Bit	Meaning	
24-31	ErrorCode	
16-23	ErrorDecode	
8-15	ErrorCode1	
0-7	ErrorCode2	

Table 8-5: Coding of PNIO status for negative responses

ErrorCode	Meaning
DB	Error in connect response
DC	Error in release response
DD	Error in control response
DF	Error in write response

Table 8-6: Meaning of ErrorCode for negative responses

ErrorDecode	Meaning
80	Read/Write service
81	Connect, Control, Release service

Table 8-7: Meaning of ErrorDecode for negative responses

ErrorCode1	Meaning
A1	write error
A2	module failure
A3-A6	reserved
A7	busy
A8	version conflict
A9	feature not supported
AA-AF	device specific
BO	invalid index
B1	write length error
B2	invalid slot/subslot
В3	type conflict
B4	invalid area/API
В5	state conflict
B6	access denied
В7	invalid range
B8	invalid parameter
В9	invalid type
BA	backup
BB-BF	device specific
CO	read constrain conflict
C1	write constrain conflict
C2	resource busy
С3	resource unavailable
C4-C7	reserved
C8-CF	device specific

Table 8-8: Meaning of ErrorCode1 for ErrorDecode = 80

ErrorCode1	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
08	Read Write Record Parameter, Error Faulty Record

ErrorCode1	Meaning
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)

Table 8-9: Meaning of ErrorCode1 for ErrorDecode = 81

ErrorCode1	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

Table 8-10: Meaning of ErrorCode2 for ErrorCode1 = 40

For ErrorCode1 <> 40 ErrorCode2 refers to a field in the block specified by ErrorCode1.

ErrorCode2	Meaning
00	Block type
01	Block length
02	Block version (high byte)
03	Block version (low byte)

Table 8-11: Meaning of ErrorCode2 for ErrorCode1 <> 40

ErrorCode2	Meaning
04	AR Type
05	ARUUID
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

Table 8-12: Meaning of ErrorCode2 for ErrorCode1 = 01 (AR block request)

ErrorCode2	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
OF	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
20	Slot
21	Subslot
22	IOCS offset

Table 8-13: Meaning of ErrorCode2 for ErrorCode1 = 02 (IOCR block request)

ErrorCode2	Meaning
04	Number of APIs
05	API
06	Slot
07	Module ident number
08	Module properties
09	Number of submodules
0A	Subslot
0B	Submodule ident number
0C	Submodule properties
0D	Data description
0E	Data length
OF	IOPS length
10	IOCS length

Table 8-14: Meaning of ErrorCode2 for ErrorCode1 = 03 (Expected submodule block request)

ErrorCode2	Meaning
04	Туре
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

Table 8-15: Meaning of ErrorCode2 for ErrorCode1 = 04 (AlarmCR block request)

ErrorCode2	Meaning
04	Sequence number
05	ARUUID
06	API
07	Slot number
08	Subslot number
09	Padding
0A	Index
0B	Data length
ос	Target AR UUID

Table 8-16: Meaning of ErrorCode2 for ErrorCode1 = 8 (Read/write record block request)

ErrorCode2	Meaning
05	Padding
06	Session key
07	Padding
08	Control block command
09	Control block properties

Table 8-17: Meaning of ErrorCode2 for ErrorCode1 = 16 (IOXControl block request)

12.3.8.9 Profinet IO RT Device Configuration

The KAS IDE contains a fully integrated configurator for Profinet IO RT 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. For example, 7 slots containing 16 outputs of 32 bits each, would be 7 * 16 * 32 = 3584 bits = 448 bytes.

IMPORTANT

Referring to the Profinet standard, the units of a Profinet network are named as IO Controllers (Masters) and IO Devices (Slaves).

NOTE

666

Profinet is only supported on AKD PDMMs and PCMMs.

12.3.8.9.1 ProfinetIO RT Device configuration

The Runtime manages a mapping table which contains the Profinet IO Inputs and Outputs. An appropriate configuration tool is integrated in the KAS.

To start the configuration:

- 1. Open the fieldbus configuration window.
- 2. Right click on the windows and choose **Insert > Insert Network**. The following window opens.

Add Configuration	×
Choose a configuration	ОК
(All) Ethernet/IP Adapter (server) Ethernet/IP I/O Scanner (client) Ethernet/IP Tag Client FlexIO / PointIO PROFINET IO controller PROFINET IO device Ethernet/IP	Cancel

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. Start the declaration of a Profinet device by right clicking on the PROFINET IO device network and choosing Insert Master. The following window opens:

PROFINET IO device		X
Properties	Value	ОК
Configuration name	pdmmNAME	Cancel
Adapter identifiaction	tsec0	
Maximum number of ARs	3	Configure slots
Initialize IOxS	v	Cornigure siots
I/O communication only from within cycle		Help
Default station name	pdmmNAME	
Vendor name (ID)	Kollmorgen (0x0147)	
Device ID	0x0002	
Instance ID	0x0001	100
Decription	KAS PN-IO device	
		Kollmorgen KAS PN-IO device

These parameters can be changed:

Parameter	Meaning
Name	A device name can consists of labels and 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 [a209-] Labels do not start with [-] Labels do not end with [-] The first label does not start with "port-xyz" or "port-xyz-abcde" with a,b,c,d,e, x, y, z = 09 Device names do not have the form n.n.n.n, n = 0999 Labels do only start with 'xn-' if RFC 3490 is applied
Adapter identification	Must be "tsec0"
Maximum # of ARs	Maximum number of alarm retries (default 3).
Initialize IOxS	On: Initialize IOxS with good status. Off: No initialization of IOxS.
IO communication only from within cycle	On: Run IO communication from within VM-cycle Off: Run IO communication outside VM-cycle.
Default station name	Name of the station.

5. Click on Configure Slots.

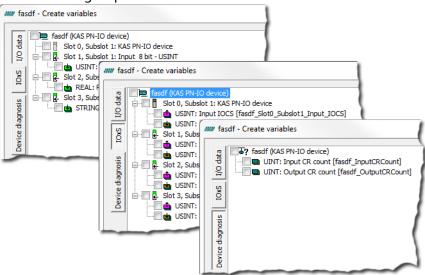
/ailabl	e modules:						
Vame		Module ID	Allowed in slot	ts Decription			ок
- Inp	out 8 bit - SINT	0x0000002	1-127	Signed 8 bit integer input i	module (-128 to 127)		
- Inp	out 8 bit - USINT	0x0000001	1-127	Unsigned 8 bit integer inpu	ut module (0 to 255)		Cancel
- Inp	out 16 bit - INT	0x00000004	1-127	Signed 16 bit integer input	t module (-32768 to 32767)	Ξ	
- Inp	out 16 bit - UINT	0x0000003	1-127	Unsigned 16 bit integer inp	put module (0 to 65535)		
- Inp	out 32 bit - DINT	0x0000006	1-127	Signed 32 bit integer input	t module (-2147483648 to 2147483647)		
- Inp	out 32 bit - REAL	0x0000008	1-127	32 bit floating-point input	module		
- Inp	out 32 bit - UDINT	0x0000005	1-127	Unsigned 32 bit integer inp	put module (0 to 4294967295)		
- Inp	out 64 bit - LINT	0x0000007	1-127	Signed 64 bit integer input	t module		
e Inp	out 64 bit - LREAL	0x0000009	1-127	64 bit floating-point input	module		100 A
Inp	out STRING - 255 byte	0x000000b	1-127	255 byte visible string inpu	ut module		
e Inp	out STRING - 64 byte	0x000000a	1-127	64 byte visible string input	t module		Kollmorgen
+ Ou	tput 8 bit - SINT	0x00001002	1-127	Signed 8 bit integer outpu	t module (-128 to 127)	-	KAS PN-IO device
lot	d modules: Module name	Module ID	Fixed I/O) data	Parameter data	-	Insert
				0 data 00 [0001 01 01 0000 0000]}	Parameter data	•	
	Module name				Parameter data		Insert Append
	Module name		✓ {00		Parameter data		
	Module name		✓ {00		Parameter data		Append
	Module name				Parameter data		Append Remove
	Module name				Parameter data		Append
ilot	Module name				Parameter data		Append Remove
	Module name				Parameter data		Append Remove
	Module name				Parameter data		Append Remove
	Module name				Parameter data		Append Remove

6. Select the modules in the upper list. With the buttons **Insert** and **Append** the modules are copied to the lower list.

668

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.

- 7. Click OK to close the Slot Configuration and click OK to close the device window.
- 8. Right click on the master and select Create Variables. This will automatically populate the variables and groups.



① IMPORTANT

Every Profinet variable is expanded to a set of Boolean variables in PLC by default. A SINT slot, for example, will be mapped to eight PLC BOOL variables. Therefore, if you have many configured slots, many PLC variables will be produced. The AKD PDMM or PCMM will be 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*. Doing so with a SINT slot, for example, will create one SINT variable in KAS instead of eight BOOL variables. This will help reduce the number of PLC variables and reduce the load on the AKD PDMM or PCMM.

🗙 TIP

The *Pack bits* action may be applied to all slots by right clicking on the root node in the *Create variables* dialog.

9. Mark the device and right click on it, then choose Insert Slave to add a group.

📁 Group - P	ROFINET IO device		×
Properties	Value	 	ОК
Name			Cancel
Type (icon)	Group		
			Help

These parameters can be changed:

Parameter	Meaning
Name	Name of the group
Type (icon)	Icon used for the group

10. Now you can connect the variables with the I/Os. Right click on the group and choose **Insert Symbol** in order to append a variable to a device.

Properties	Value	OK
Name		
Area	Input	Cancel
Data type	Unsigned 16 bit integer	Help
Slot	0	
Subslot	0	
Offset	0	
Bit number	0	
String length	0	

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.
Format	32 bit float, Signed 16 bit integer, Signed 32 bit integer, Signed 8 bit integer, Single bit, Unsigned 16 bit integer, Unsigned 32 bit integer, Unsigned 8 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.

All settings can be changed in the grid too. The information show 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
n i i iono i		e de la deserve	<u>^</u>	~	~	~

If KAS is connected to a target system and the system is running, the grid shows the real-time data of the variables.

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. For example, 7 slots containing 16 outputs of 32 bits each, would be 7 * 16 * 32 = 3584 bits = 448 bytes.

12.3.8.9.2 Data types

You can 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.

12.3.8.9.3 Additional features

12.3.8.9.3.1 Create Variables

Find this command in the context menu of the device. Based on the defined device modules the referring variables are generated.

Within the same dialog it is possible to define the variables for

- Device diagnosis
- IOxS

12.3.8.9.3.2 Device Diagnosis

This retrieves the device state information. Based on the Profinet standard the referring variables can be generated:

- InputCRCount [UINT].
- OutputCRCount [UINT].

12.3.8.9.3.3 Create IOxS for Slave Modules

Based on the defined device modules the referring IOPS- and IOCS-variables are generated.

Input modules/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 controller has to ignore it.
- The IOCS state is managed by the controller. The controller can indicate if it can't handle the data.

Output modules:

- The IOPS state is managed by the controller it reflects the status of the data sent by the controller. If the data is invalid the device has to ignore it
- The IOCS state is managed by the device. Le device can indicate to the controller that the data can't be handled.

12.3.8.9.4 How to resolve errors

12.3.8.9.4.1 Device is not found

- Check if device is switched on
- Check the network connection
- Ensure the correct name was set on the device

12.3.8.9.4.2 Error setting the IP configuration

- Ensure that the IP configuration is valid and appropriate for your network
- Ensure the VLAN ID is setup correctly in the device settings

12.3.8.9.4.3 CL-RPC Lookup

12.3.9 Timeout error

- Ensure that the IP configuration is valid and appropriate for your network
- Increase the IP configuration delay

12.3.9.0.0.1 Other errors

- Ensure you use the correct device name
- Ensure you have chosen the correct GSDML file (also check if the GSDML file version is appropriate for the firmware version of the device)

12.3.9.0.0.2 Connect response error

12.3.10 Timeout error

- Ensure that the IP configuration is valid and appropriate for your network
- Increase the IP configuration delay

12.3.10.0.0.1 Connect response error

Depending on the PNIO status error check, if the settings listed in the table below are set appropriate for your device.

PNIO Status	Check the specified setting
1C010003	Slave\ObjectUUID (correct GDML file)
DB81010A	Master\Activity timeout factor
DB81010B	Master\Controller name
DB81010C	Master\Controller name
DB810207	Master\RT-Class
DB81020A	Master\Send clock factor
DB81020B	Slave\Reduction ratio
DB81020C	Slave\Phase
DB81020F	Master\Watchdog factor
DB810210	Master\Watchdog factor
DB8103,*	Slave\Module IDs,IOs
DB810407	Master\Alarm timeout
DB810408	Master\Alarm retries

Table 8-18: PNIO status error codes on connect and the related settings in the configuration

12.3.10.0.0.2 Module configuration is different

- Ensure you have configured the modules present on the device (If you have read the module configuration from the device, ensure you have selected the correct modules in the resolve ambiguous modules dialog displayed after selecting the GSDML file).
- Ensure you have chosen the correct GSDML file (also check if the GSDML file version is appropriate for the firmware version of the device).

12.3.10.0.0.3 Writing parameterization error (with status 0XDF80*)

- Ensure you have configured the modules present on the device (If you have read the module configuration from the device, ensure you have selected the correct modules in the resolve ambiguous modules dialog displayed after selecting the GSDML file).
- Ensure you have chosen the correct GSDML file (also check if the GSDML file version is appropriate for the firmware version of the device).

If your error could not be resolved using the hints above, update the firmware of your device and the GSDML file to the latest version and try again.

For further analyses record the PNIO communication using Wireshark (http://www.wireshark.org/).

- Start Wireshark and select Capture\Options from the menu.
- Choose the network interface used for the PNIO communication and enter the Capture filter below:

```
ether proto 0x8892 or (ether proto 0x8100 and ether[16:2]=0x8892) or udp port 0x8894
```

- Then click start.
- Start the runtime with your application several times and then stop the Wireshark capturing by selecting Capture\Stop from the menu. You can save your recorded traffic by selecting File\Save from the menu.

12.3.10.0.1 Coding of PNIO status

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 PNIO status has a value of zero.

Bit	Meaning
24-31	ErrorCode
16-23	ErrorDecode
8-15	ErrorCode1
0-7	ErrorCode2

Table 8-19: Coding of PNIO status for negative responses

Meaning
Error in connect response
Error in release response
Error in control response
Error in write response

Table 8-20: Meaning of ErrorCode for negative responses

ErrorDecode	Meaning
80	Read/Write service
81	Connect, Control, Release service

 Table 8-21: Meaning of ErrorDecode for negative responses

ErrorCode1	Meaning
A1	write error
A2	module failure
A3-A6	reserved
A7	busy
A8	version conflict
A9	feature not supported
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

ErrorCode1	Meaning
B7	invalid range
B8	invalid parameter
В9	invalid type
BA	backup
BB-BF	device specific
С0	read constrain conflict
C1	write constrain conflict
C2	resource busy
C3	resource unavailable
C4-C7	reserved
C8-CF	device specific

Table 8-22: Meaning of ErrorCode1 for ErrorDecode = 80

ErrorCode1	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
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)

Table 8-23: Meaning of ErrorCode1 for ErrorDecode = 81

ErrorCode1	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

Table 8-24: Meaning of ErrorCode2 for ErrorCode1 = 40

For ErrorCode1 <> 40 ErrorCode2 refers to a field in the block specified by ErrorCode1.

ErrorCode2	Meaning
00	Block type
01	Block length
02	Block version (high byte)
03	Block version (low byte)

Table 8-25: Meaning of ErrorCode2 for ErrorCode1 <> 40

ErrorCode2	Meaning
04	AR Type
05	ARUUID
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

Table 8-26: Meaning of ErrorCode2 for ErrorCode1 = 01 (AR block request)

ErrorCode2	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
OF	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

ErrorCode2	Meaning
18	IO data object offset
19	Number of IOCS
20	Slot
21	Subslot
22	IOCS offset

Table 8-27: Meaning of ErrorCode2 for ErrorCode1 = 02 (IOCR block request)

ErrorCode2	Meaning
04	Number of APIs
05	API
06	Slot
07	Module ident number
08	Module properties
09	Number of submodules
0A	Subslot
0B	Submodule ident number
0C	Submodule properties
0D	Data description
0E	Data length
OF	IOPS length
10	IOCS length

Table 8-28: Meaning of ErrorCode2 for ErrorCode1 = 03 (Expected submodule block request)

ErrorCode2	Meaning
04	Туре
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

Table 8-29: Meaning of ErrorCode2 for ErrorCode1 = 04 (AlarmCR block request)

ErrorCode2	Meaning
04	Sequence number
05	ARUUID
06	API
07	Slot number
08	Subslot number
09	Padding

676

ErrorCode2	Meaning
0A	Index
0B	Data length
0C	Target AR UUID

Table 8-30: Meaning of ErrorCode2 for ErrorCode1 = 8 (Read/write record block request)

ErrorCode2	Meaning
05	Padding
06	Session key
07	Padding
08	Control block command
09	Control block properties

Table 8-31: Meaning of ErrorCode2 for ErrorCode1 = 16 (IOXControl block request)

12.3.11 Performance Guidelines

Additional controller CPU usage is required when fieldbuses are used to transmit and receive data between network devices and PLC variables. The impact to PLC execution time and the CPU load can be estimated as follows.

```
PLC Time = (Mapped PLC Variable Count * Exchange Time per Variable) + PLC 
Overhead
```

AKD PDMM or PCMM model Exchange Time per Variable

800 MHz	0.4 microseconds
1.2 GHz	0.2 microseconds

Table 8-32: Average Exchange Time per Variable by AKD PDMM / PCMM model used to calculate performance.

NOTE

The exact *Exchange Time per Variable* is dependent on the data type. BOOL variables take less time, while REAL variables require more time.

12.3.11.1 EtherNet/IP

```
CPU Usage % = (PLC Time / ECAT Cycle Time) + (Assembly Count * Load per Assembly)
```

AKD PDMM or PCMM model	PLC Overhead	Load per Assembly
800 MHz	20 microseconds	0.0075 (0.75%)
1.2 GHz	10 microseconds	0.00375 (0.375%)

Table 8-33: Values by AKD PDMM and PCMM models used to calculate performance with EtherNet/IP.

The exact percent *Load per Assembly* is dependent on the data size per assembly and the *Requested Packet Interval (RPI)*. The *Load per Assembly* values listed above represent the maximum percentage, with 496 bytes per Assembly and *RPI* = 10 milliseconds.

12.3.11.1.1 EtherNet/IP Performance Example

This example uses an 800 MHz AKD PDMM with 4x Assemblies, 372 mapped PLC variables, and a 1000 microsecond EtherCAT Cycle Time.

```
PLC Time = (372 * 0.4) + 20 = 169 microseconds
CPU Usage % = (169 / 1000) + (4 * .0075) = 0.20 (20%)
```

Thus, exchanging data with 372 PLC variables, over EtherNet/IP, will increase the PLC execution time by 169 microseconds and reduce the CPU Idle Time by 20%.

12.4 Project Structure Guidelines

12.4.1 Introduction

By implementing a predefined structure for new projects, KAS tries to achieve the following 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)

12.4.2 External Files

Some items that belongs to your application (displayed in the **Project Explorer**) are not embedded into the project file. For the domains listed below, KAS IDE also uses some resources that are stored in external files.

Domain	Description	File
	Using Kollmorgen HMI, simply 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	KVB Project File
PLC	The PLC programming environment gives you the possibility to create reusable components (UDFB), and template applications which can be customized to suit any given application	Create Custom Libraries Read Common Constants
Motion	The CAM editor lets you create complex CAM profiles online using a "graphical" interface. It is also possible to import existing CAM profile points into the CAM editor to allow you to reuse your existing machine building experience seamlessly	Import Cam Profile How Do I Export the Oscilloscope Data?
Fieldbus	Kollmorgen Automation Suite tightly integrates the EtherCAT motion bus (standard Ethernet-based cabling) to define all the network description	Import or Export EtherCAT ENI File

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	Download AKD Firmware

Table 8-34: - File location

🗙 TIP

The hyperlinks bring you to the relevant topic that contains more details.

12.4.3 Application Software Structure - Definitions

12.4.3.1 Modules to build up the Structure

12.4.3.1.1 Structure Overview

You normally write the PLC program. Whereas Kollmorgen application team members create in most cases the motion control part.

The global software structure is built up with different modules placed on two different levels as showed on the figure below:

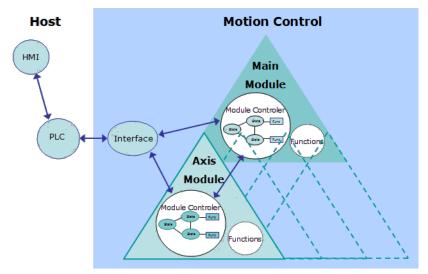


Figure 8-28: Software Structure Overview

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.

12.4.3.1.2 Main Module description

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.

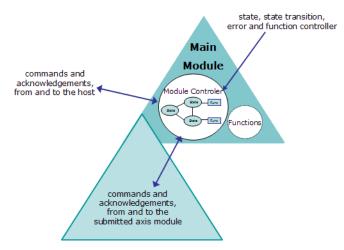


Figure 8-29: Main Module Description

As shown on the figure, the main module consists of two parts:

- the **module controller** 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 acknowledgements. 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 acknowledgement. The main module controller also manages the error status of the submitted modules and performs the needed actions.
- the **functional part** 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).

12.4.3.1.3 Axis Module description

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 acknowledgements.

The axis module also communicates with its main module via the internally defined data channel.

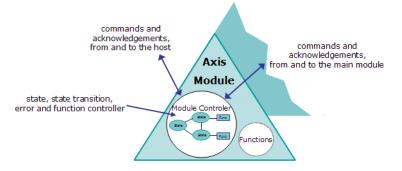


Figure 8-30: Axis Module Description

As shown on the figure, the axis module consists of the same two parts as the main module:

- the **module controller** part is responsible for state, state transition, error and functions handling. 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 acknowledgements. If 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.
- the **functional part** consists of all functionalities needed for the current physical axis. These functions can be state dependent (e.g. single axes functions) or state independent (e.g. increase a speed value).

12.4.3.2 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 fulfil specific application usage (see also How to add a new state and How to add a new function).

12.4.3.2.1 State transition Diagram

The following state machine has been defined.

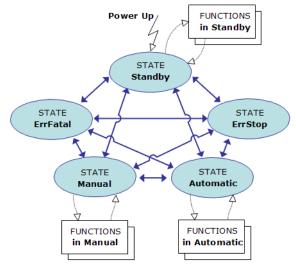


Figure 8-31: State Machine

Legend

682

STATE	Represents the current state of one or more modules (e.g. Standby state means that the module is waiting for commands)
	In addition to the static states, there are two other possible 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
FUNCTIONS	Any needed machine or single-axis functionality (e.g. single-axis jog), which can be called out of one or several states

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.

12.4.3.2.2 State, state transitions and functions descriptions

The structure is built in such a way that 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. That 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.

12.4.4 Application Software Structure - Implementation

This chapter 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.

12.4.4.1 SFC children building up the software

The following files contain all the data to build up the application. They are all required to ensure a successful compilation.

Parent SFC

Main System start up and SFC children call	
--	--

Main module SFC children

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

Axis module SFC children

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

With i = 1... n

12.4.4.2 Variables for the Interface

12.4.4.2.1 List of variables

- M1_CmdState
- bM1_CallStandbyFunction1
- bM1_CallStandbyFunction2
- bM1_CallManualFunction1
- bM1_CallManualFunction2

- bM1_CallAutomaticFunction1
- bM1_CallAutomaticFunction2
- bAi_CallStandbyFunction1
- bAi_CallStandbyFunction2
- bAi_CallManualFunction1
- bAi_CallManualFunction2
- bAi_CallAutomaticFunction1
- bAi_CallAutomaticFunction2
- bErrorReset

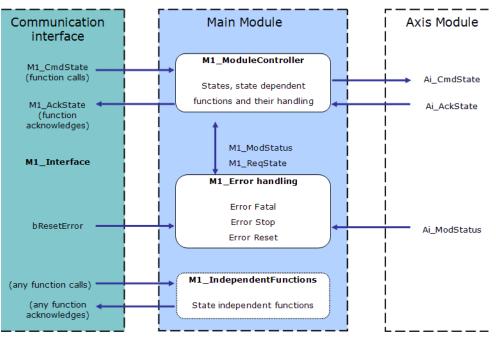
12.4.4.2.2 List of output variables

- M1_AckState
- M1_StatusWord
- bM1_Running
- Ai_StatusWord
- bAi_Running

12.4.4.3 Main module implementation description

In the main module, all necessary state, state transition, error and function handling facilities are implemented for this level.

Context diagram for the main module



The following objects (variables, tasks...) are defined in the structure of the main module.

12.4.4.3.1 M1_CmdState

12.4.4.3.1.1 Description

This internal word variable contains the actual state command value. It is automatically set to state '**Standby**' during power up.

<pre>//** State Defines ** //*********************************</pre>	//*****	******	* * *
#define DEF_StateUndefined0#define DEF_StateStandby1#define DEF_StateManual2#define DEF_StateAutomatic3#define DEF_StateBusy4#define DEF_StateErrorStop5	//**	State Defines	* *
#define DEF_StateStandby1#define DEF_StateManual2#define DEF_StateAutomatic3#define DEF_StateBusy4#define DEF_StateErrorStop5	//*****	* * * * * * * * * * * * * * * * * * * *	* * *
#define DEF_StateManual2#define DEF_StateAutomatic3#define DEF_StateBusy4#define DEF_StateErrorStop5	#define	DEF_StateUndefined	0
#define DEF_StateAutomatic3#define DEF_StateBusy4#define DEF_StateErrorStop5	#define	DEF_StateStandby	1
#define DEF_StateBusy4#define DEF_StateErrorStop5	#define	DEF_StateManual	2
#define DEF_StateErrorStop 5	#define	DEF_StateAutomatic	3
_	#define	DEF_StateBusy	4
<pre>#define DEF_StateErrorFatal 6</pre>	#define	DEF_StateErrorStop	5
	#define	DEF_StateErrorFatal	6

12.4.4.3.1.2 Usage

These 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, then the definitions described above can be modified accordingly.

12.4.4.3.2 M1_AckState

12.4.4.3.2.1 Description

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 above).

12.4.4.3.2.2 Usage

Out of this value the corresponding acknowledgements for the PLC can be created in the communication interface.

12.4.4.3.3 M1_ReqState

12.4.4.3.3.1 Description

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 above).

12.4.4.3.3.2 Usage

Used by system, do not use it for application purpose.

12.4.4.3.3.3 Description

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 as follows:

| Bit |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Bits 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)

12.4.4.3.3.4 Usage

While 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).

12.4.4.3.4 bErrorReset

12.4.4.3.4.1 Description

This internal flag variable is used as the error reset command for the main and axis modules. It is reset during power up.

12.4.4.3.4.2 Usage

Set and reset this flag to activate a reset of the module errors (M1_StatusWord, Ai_StatusWord).

12.4.4.3.5 M1_ErrorHandling

12.4.4.3.5.1 Description

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. This causes the error reaction bits (MSK_Mi_StatusErrorStop, MSK_Mi_StatusErrorFatal) to be set in the module status word.

12.4.4.3.5.2 Usage

Any additional error which needs to be treated has to be included in this program. Do not forget to modify the corresponding masks (**MSK_M1_StatusErrorFatal**, **MSK_M1_StatusErrorStop**) to cause the correct reaction on errors.

12.4.4.3.6 M1_ModuleController

12.4.4.3.6.1 Description

This program is the heart of the whole controller and contains:

- a state manager sequence
- all state sequences
- and state dependent function sequences of the main module

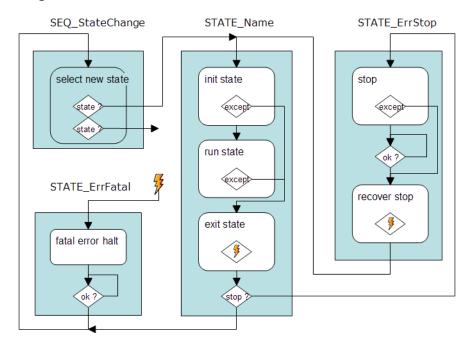
12.4.4.3.6.2 Usage

Some rules have to be followed, when using and changing states and functions (see also How to add a new state and How to add a new function).

12.4.4.4 States and Errors

12.4.4.1 How States and Errors are treated

The figure below shows how states and errors are treated.



StateChange (state manager)

Activates the new state required by M1_ReqState

StateName (state macro)

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 (init variables, pipes,)
	- Sends commands to the submitted axis modules by setting Ai_CmdState to StateName and waits for their acknowledgement in Ai_AckState
	- Acknowledges end of initialization by setting M1_AckState to M1_ReqState
run state	- Waits for any function calls, activate 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

12.4.4.2 How to add a new state

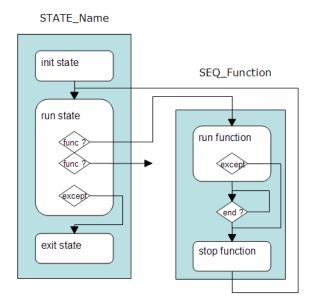
To add a new state, do as follows:

- 1. copy a similar existing state sequence
- 2. replace the old state name by the new one (e.g. 'Standby' by '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

12.4.4.5 Functions linked to states

12.4.4.5.1 How Functions are treated

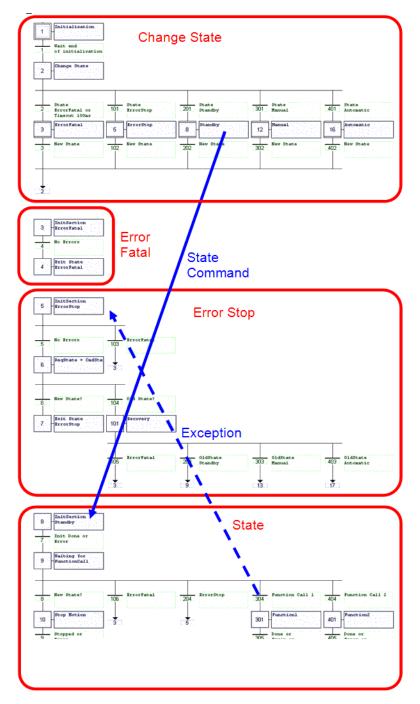
The figure below shows how functions (that are state dependent) are treated.



Function (function step)

688

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



12.4.4.5.2 How to add a new function

To add a new function, do as follows:

- 1. copy a similar existing function sequence
- 2. replace the old function name by the new one (e.g. 'Running' by '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

12.5 Project Templates

KAS provides start up templates to help you get started (see how to use the project setup wizard here). There are separate templates for the Pipe Network motion engine and for the PLCopen motion engine:

Template Type	Template name	Description
PipeNetwork	1 Axis FFLD	Simple initialization, relative position move
	1 Axis ST	Simple initialization, relative position move
	2 Axes FFLD	Position and velocity moves, gearing, camming, and drive status
	2 Axes ST	Position and velocity moves, gearing, camming, and drive status
	2 Axes SFC	Position and velocity moves, gearing, camming, and drive status with optimized performance.
PLCopen	1 Axis FFLD	Simple initialization, relative position move
	1 Axis ST	Simple initialization, relative position move
	2 Axes FFLD	Position and velocity moves, gearing, camming, and drive status
	2 Axes ST	Position and velocity moves, gearing, camming, and drive status
	2 Axes SFC	Position and velocity moves, gearing, camming, and drive status with optimized performance.
Coordinated Motion	2 Axes - Linear / Circular	Raster Scan Motion Path, 2 PLCopen axes
	3 Axes - Linear / Circular	Raster Scan Motion Path, 2 PLCopen axes and 1 PipeNetwork axis
	3 Axes - Linear (3D)	Diamond/Square Motion Path, 3 PLCopen axes
KAS Runtime	Library	Allows you to create a custom library (Create and Use Custom Libraries)

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.

Dictionary						
Controller:PLC 🛛 👻	📃 Track Sele	ection				
ነA Name	Туре	Dim.	Attrib.	Init value		
TravelSp	LREAL			LREAL		
MasterA	LREAL			LREAL		
MasterDe	LREAL			LREAL		
Machine	LREAL			LREAL		
Axis1Sta	DINT					
Axis2Sta	DINT					
Machine	DINT			0		
bMaster	BOOL			FALSE		
bMaster	BOOL			FALSE		
bEStop	BOOL			FALSE		
bLedStat	BOOL	[03]				
Profiles	ProfilesCo					
EtherCAT	EtherCAT		Read Only			
PipeNet	PNCode					
💾 Retain variables						
🗆 💼 MachineLogic						
lastTrave	LREAL			LREAL		
lastMach	LREAL			LREAL		

690

Control Panel		
Kollmorgen Au Contro	DI Panel	ialised wer nning ULT
Travel Speed	Machine Speed	
Rel Move		
NOTE		

Additionally, the templates contain a Control Panel for ease of running motion.

You may need to modify the templates to run on your specific hardware.

12.5.1 Pipe Network 1-Axis Template with FFLD only

The 1-axes Pipe Network template has a FFLD program (called **Main**) that initializes, starts and runs the motion. Also included are controls to enable/disable the drive, clear faults, and display fault and axis position information in the Control Panel.

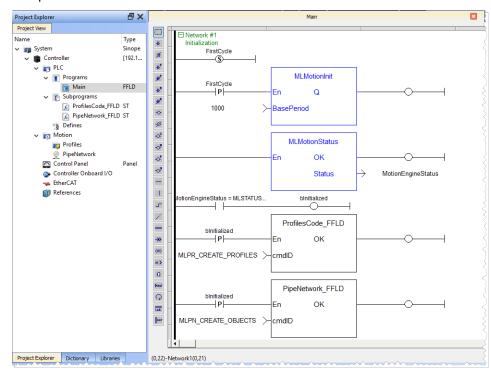


Figure 8-32: PN Template with FFLD - Main

Kollmorgen Automation Suite

EtherCAT Enable Enabled Clear Fault Axis Position Move Distance Relative Move	Control Panel						
Move Distance deg			Disable				
	Axis Position		deg				
	Move Distance	Relative Move	deg				

12.5.2 Pipe Network 1-Axis Template with ST only

The 1-axis Pipe Network template has a ST program (called **Main**) that initializes, starts and runs the motion. Also included are controls to enable/disable the drive, clear faults, and display fault and axis position information in the Control Panel.

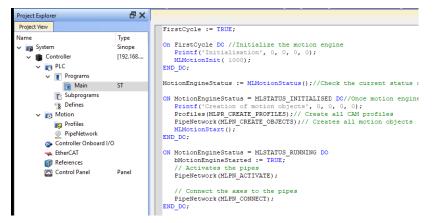


Figure 8-33: PN Template with ST - Main

Kollmorgen Automation Suite

 Ether(Enable 		Enable Disable Clear Fault
Axis Position		deg
Move Distance		deg
	Relative Move	

12.5.3 Pipe Network 2-Axes Template with SFC, ST, FFLD, and FBD

12.5.3.1 PLC Programs

The 2-axes Pipe Network template has an SFC program (called **Main**) that initializes and starts the motion.

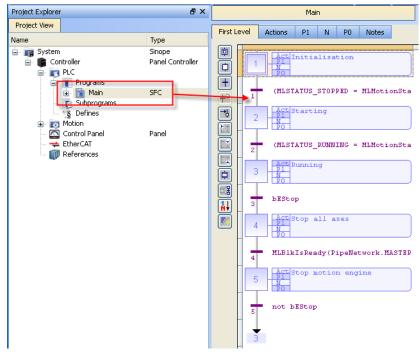


Figure 8-34: PN Template - Main

The Pipe Network Template contains an SFC child program called Machine Logic for running motion.

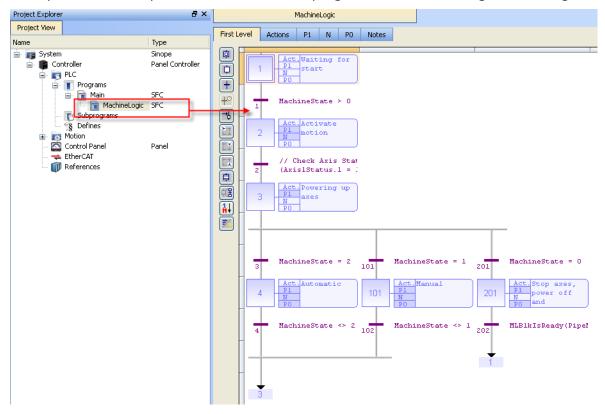
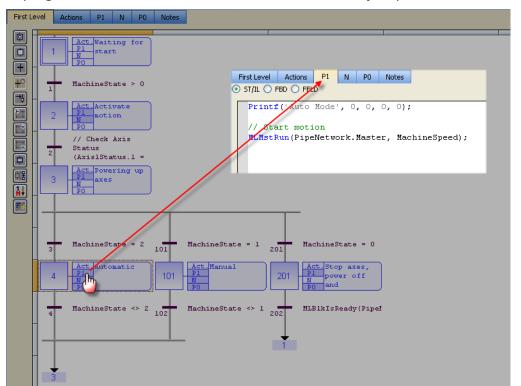
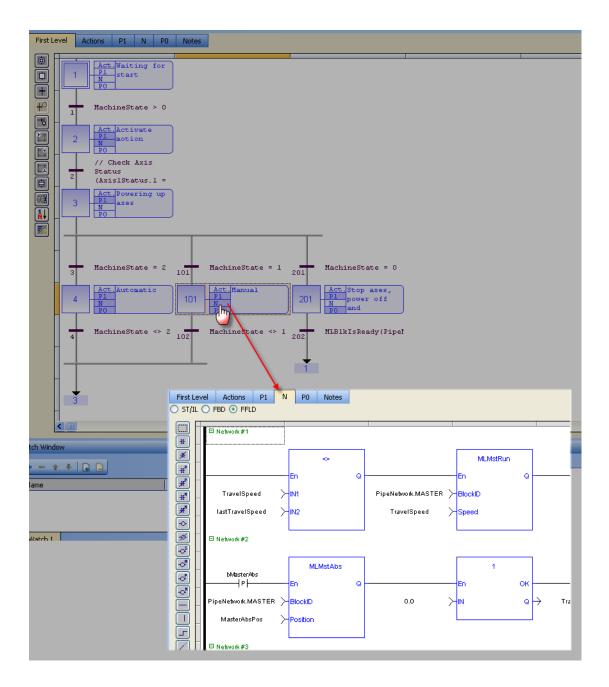


Figure 8-35: PN Template - MachineLogic

ST programs can be found in the P1 and P0 actions for many steps



FFLD programs can be found in the N action for steps 4 and 101



12.5.4 Motion

The template has a motion profile defined with the graphical Pipe Network editor.

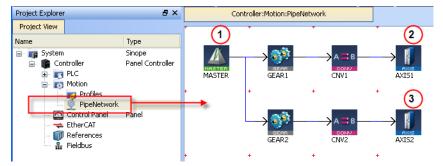


Figure 8-36: PN Template - Motion

The motion profile contains four 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

12.5.5 Control Panel

For more details, see Design the Control Panel with the Internal Control Panel Editor

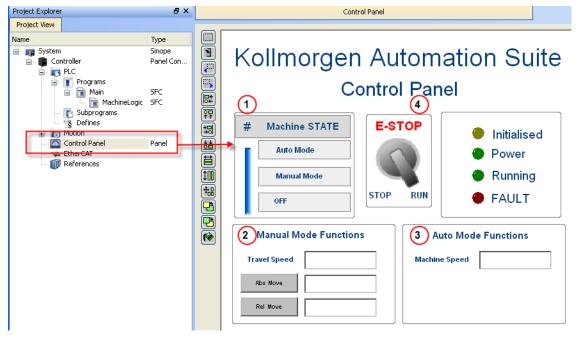


Figure 8-37: PN Template - Control Panel

Call out#	Description
1	Allows to choose how to run the axes between automatic and manual modes
2	In manual mode, you can set the speed. You can also 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
3	In automatic mode, you can set the speed
4	When you click the emergency button, the machine state becomes OFF (see call out \fbox)
	and the two axes stop running
Tabla	9 25: DN Tamplata Control Danal

Table 8-35: PN Template - Control Panel

Based on the template, the project can be run:

- using the KAS Simulator
- with actual drives and motors (in this case, you first have to set up the axes in the EtherCAT part. For more details, see "Configuring EtherCAT" on page 320)

12.5.5.1 HMI

Also included in the project template is an HMI screen for the AKI terminals created in the KVB (Kollmorgen Visualization Builder) environment.

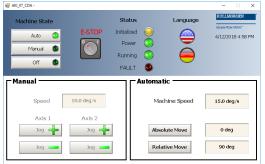
The KVB software is a separate installation from the KAS IDE. KVB can be opened from within KAS IDE by double-clicking on the **KVBProject** item in the project tree.



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.



A screen, localized in English and German, is included which duplicate the controls on the IDE Internal Control panel screen.



12.5.6 Pipe Network 2-Axes Template with ST only

12.5.6.1 PLC Programs

The 2-axes Pipe Network template has a ST program (called **Main**) that initializes, starts and runs the motion.

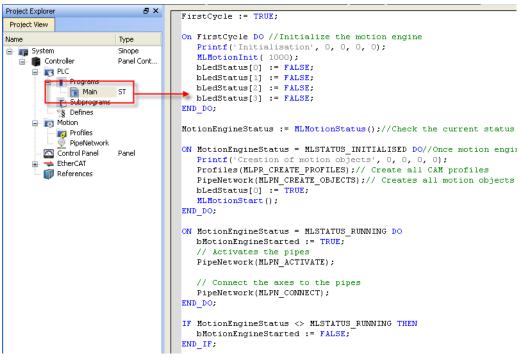


Figure 8-38: PN Template with ST - Main

12.5.7 Motion

698

The template has a motion profile defined with the graphical Pipe Network editor.

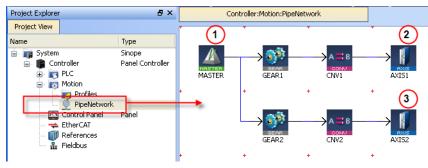


Figure 8-39: PN Template - Motion

The motion profile contains four 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

12.5.8 Control Panel

For more details, see Design the Control Panel with the Internal Control Panel Editor

Project Explorer	₽×	Control Panel	
Project View			
Name Controller Controller Controller Main Main Subprograms Subprograms Control Panel Control Panel Control Panel Cher CAT References	Type Sinope Panel Con SFC SFC Panel	State Kollmorgen Automation Suit Control Panel Image: Contr	e

Figure 8-40: PN Template - Control Panel

Call out#	Description
1	Allows to choose how to run the axes between automatic and manual modes
2	In manual mode, you can set the speed. You can also 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
3	In automatic mode, you can set the speed
4	When you click the emergency button, the machine state becomes OFF (see call out $\overbrace{1}$)
	and the two axes stop running
Tabla	8 26: DN Tamplata Control Danal

Table 8-36: PN Template - Control Panel

Based on the template, the project can be run:

- using the KAS Simulator
- with actual drives and motors (in this case, you first have to set up the axes in the EtherCAT part. For more details, see "Configuring EtherCAT" on page 320)

12.5.8.1 HMI

Also included in the project template is an HMI screen for the AKI terminals created in the KVB (Kollmorgen Visualization Builder) environment.

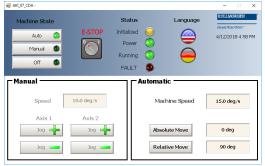
The KVB software is a separate installation from the KAS IDE. KVB can be opened from within KAS IDE by double-clicking on the **KVBProject** item in the project tree.



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.



A screen, localized in English and German, is included which duplicate the controls on the IDE Internal Control panel screen.



12.5.9 Pipe Network 2-Axes Template with FFLD only

12.5.9.1 PLC Programs

The 2-axes Pipe Network template has a FFLD program (called Main) that initializes, starts and runs the motion.

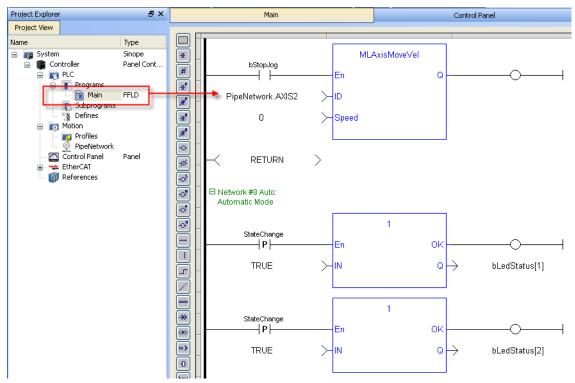


Figure 8-41: PN Template with FFLD - Main

12.5.10 Motion

The template has a motion profile defined with the graphical Pipe Network editor.

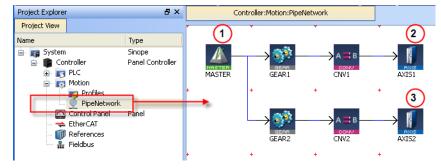


Figure 8-42: PN Template - Motion

The motion profile contains four 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

12.5.11 Control Panel

For more details, see Design the Control Panel with the Internal Control Panel Editor

Project Explorer	₽×	Control Panel
Project View		
Name Controller Controller Programs Main Subprograms Subprograms Control Panel Control Panel Control Panel Control Panel Control Panel Control Panel Control Panel	Type Sinope Panel Con SFC SFC Panel	Kollmorgen Automation Suite Control Panel # Machine STATE Auto Mode Manual Mode OFF OFF OFF OFF Manual Mode Functions Travel Speed Bos Move Rel Move 3 Auto Mode Functions Machine Speed

Figure 8-43: PN Template - Control Panel

Call out#	Description
1	Allows to choose how to run the axes between automatic and manual modes
2	In manual mode, you can set the speed. You can also 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
3	In automatic mode, you can set the speed
4	When you click the emergency button, the machine state becomes OFF (see call out \bigcirc) and the two axes stop running

Table 8-37: PN Template - Control Panel

Based on the template, the project can be run:

- using the KAS Simulator
- with actual drives and motors (in this case, you first have to set up the axes in the EtherCAT part. For more details, see "Configuring EtherCAT" on page 320)

12.5.11.1 HMI

702

Also included in the project template is an HMI screen for the AKI terminals created in the KVB (Kollmorgen Visualization Builder) environment.

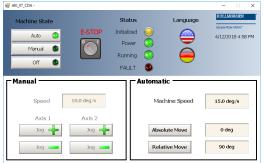
The KVB software is a separate installation from the KAS IDE. KVB can be opened from within KAS IDE by double-clicking on the **KVBProject** item in the project tree.



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.



A screen, localized in English and German, is included which duplicate the controls on the IDE Internal Control panel screen.



12.5.12 PLCopen 1-Axis Template with FFLD

The 1-axis PLCopen template has a FFLD program (called **Main**) that initializes and starts the motion. Also included are controls 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.

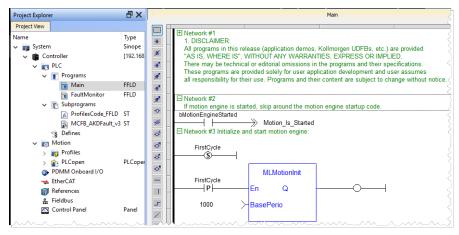


Figure 8-44: PLCopen Template with FFLD - Main

_	Machine Control	
	E-Stop Clear Fault Ethercat Networ	* ●
	Single Axis Motion Axis 1	
	Position: 4122.55 deg Fault 🔶 Enabled 🔵	
l	Velocity deg/s	
I	Acc/Dec deg/s^2 Rei Move deg	
•	Acc/Dec deg/s ⁴ 2	

12.5.13 PLCopen 1-Axis Template with ST

This 1-axis PLCopen template has a ST program (called **Main**) that initializes, starts and runs the motion. Also included are controls to enable/disable the drives, 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.

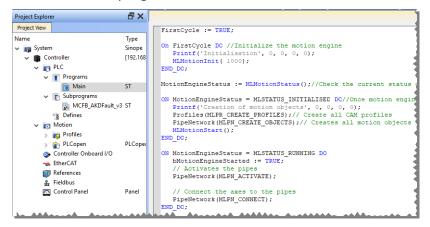
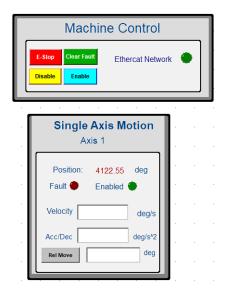


Figure 8-45: PLCopen Template with ST - Main



12.5.14 PLCopen 2-Axes Template with SFC and FFLD

This template project contains a two-axis template for getting started with a KAS application. It contains key project elements that will save development time. Additional axes can be added to the template applications that have three, four, five, or more axes. The templates are designed to easily add code to the programs and Control Panels that are in the template or to new ones. A project can contain many programs, and each program can be written in ST, SFC, FFLD, FBD, or IL programming languages. This template program can be run on the KAS Simulator or with a AKD PDMM or PCMM and AKD drives.

This project contains two axes where Axis 2 is slaved to Axis 1 at a 2:1 ratio.

12.5.14.1 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. Also included are controls to enable/disable the drives, clear faults, and display fault and axis position information in the Control Panel.

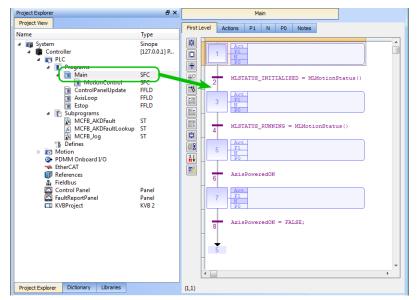


Figure 8-46: PLCopen - Template Main

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.

KAS IDE User Guide | 12 Advanced Topics

		Ma	ain*					
	First Level Actions P1 N P0 Notes							
		○ FBD						
		Network #8 Run Axis 1 when the Run/Stop switch is	s set					
	_	GearlnActive2			MoveVelocity		MoveVe	elinVel1
*	H			Execute	InVelocity		(MoveVe)
∃£" }#"	$\left \right $	Axis1	\succ	Axis	Busy			Active1
•	$\left \right $	45	\succ	Velocity	Active		(D
ø		30	\succ	Acceleration	CommandAborted		()
-~"		30	\succ	Deceleration	Error		(
-		0	\succ	Jerk	ErrorID	\rightarrow	MoveVe	elErrorID1
-		Dir	\succ	Direction				
		0	\succ	BufferMode				
F								
		Network #9 Stop both axes when the Run/Stop swit	ch is	set to Stop				
							Halt1	Liet
	H	StartMove			HaltDone1			Halt
in >	Н						Execute	Done -
Ð	H				Axis1	\succ	Axis	Busy-
} ■₩					30	\succ	Deceleration	Active -
C.P.					0	\succ	Jerk	CommandAborted -
	_				0	\succ	BufferMode	Error -

Figure 8-47: PLCopen Template - Step 5 of the Main

12.5.14.2 Motion

706

The template contains two PLCopen Servo axes where User Units, Update Rate, Rollover Position, and Axis Limits are defined as follows:

Project Explorer	8 ×	PLCopen Axis Data
Project View Name System	F × Type Sinope [10.50.68.39] PDMM PLCopen Panel AKI Touchscreen KVB 2	PLCopen Axis Data
Project Explorer Dictionary Libraries		OK Cancel

Figure 8-48: PLCopen Template - Motion

For more details on PLcopen axis parameters, see "Axis Data Parameters" on page 158

12.5.14.3 Control Panel

The template contains a Control Panel which works inside of the IDE when running a project, and serves as a useful tool to run and debug programs. The Control Panel consists of two screens, named **Control Panel** and **Fault Report Panel**.

Control Panel consists of three sections.

- Multi-Axis Commands which controls motion and shows gearing
- Single Axis Motion Commands which shows motion at the single axis level and resetting the axis position
- Axis Status which shows basic axis information

For more details, see Design the Control Panel with the Internal Control Panel Editor

Ethercal Network	atus Axis 2 Position: 3258.64 deg Fault Enabled
System Sorope Tropper Sorope Tropper Sorope Tropper Sorope Tropper Sorope Tropper Sorope S	Position: 3258.64 deg
Programs Subprograms E. Stop Clear Fault Ethercat Network Position: 4122.55 deg P	
Weisen Image: Control in the second	ault Num 0
	Fault Description Position Error
Engage Geening Run/Stop 3 Axis 1 Single Axis N Engage Camming 4 4 4 4 4	Motion Axis 2
Disengage	cc/Dec deg/s ⁴ 2
bet Move deg	Rel Move deg
	Set Zero

Figure 8-49: PLCopen Template - Control Panel

Call out#	Description
	Allows you to set the speed
2	Displays the actual position for each axis
3	Control the axis motion
4	Start or stop the motion on the condition that the axes are enable (the green light must be switched on)
5	Allows for enabling or disabling the axes. After an emergency stop, you need to select the Reset and Enable commands before running the axes
Table	8-38. Pl Copen Template - Control Papel elements

Table 8-38: PLCopen Template - Control Panel elements

Fault Report Panel provides more detailed information about when a drive fault occurs.

Axis 1	Axis 2
Fault 🔴	Fault 🔴
Fault Num	Fault Num
Fault Description	Fault Description
Axis 1 Position:	Axis 2 Position:
Position Error 0.0	Position Error .00240
Position Error Limit	Position Error Limit
Limit Exceeded 🛛 🔴	Limit Exceeded 🔶

Figure 8-50: PLCopen Template - Fault Report Panel

Based on the template, the project can be run:

- using the KAS Simulator
- with actual drives and motors (in this case, you first have to set up the axes in the EtherCAT part. For more details, see "Configuring EtherCAT" on page 320)

12.5.14.4 HMI

Also 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. KVB can be opened from within KAS IDE by double-clicking on the **KVBProject** item in the project tree.

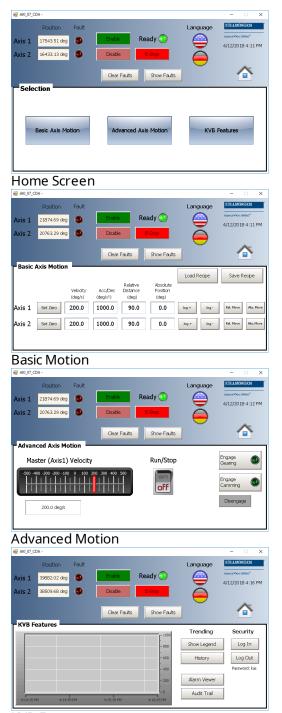


The project can be run in the KVB programming environment after the KAS IDE program has started running. Click on the Run button in the Project ribbon to start the program.



708

Four screens, localized in English and German, are included that duplicate the controls on the IDE Internal Control panel screens.



KVB Features

12.5.15 PLCopen 2-Axes Template with ST

This template project contains a two-axis template for getting started with a KAS application. It contains key project elements that will save development time. Additional axes can be added to the template applications that have three, four, five, or more axes. The templates are designed to easily add code to the programs and Control Panels that are in the template or to new ones. A project can contain many programs, and each program can be written in ST, SFC, FFLD, FBD, or IL programming languages. This template program can be run on the KAS Simulator or with a AKD PDMM or PCMM and AKD drives.

12.5.15.1 PLC Programs

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.

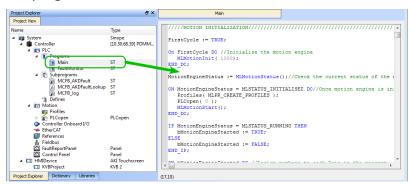


Figure 8-51: PLCopen Template with ST - Main

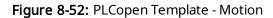
Also included are three subprograms (**MCFB_AKDFault**, **MCFB_AKDFaultLookup**, and **MCFB_Jog**) that are Kollmorgen UDFBs and can be found in the Project Library section of the IDE.

12.5.15.2 Motion

710

The template contains two PLCopen Servo axes where User Units, Update Rate, Rollover Position, and Axis Limits are defined as follows:

Project Explorer	8 ×	PI Copen Axis Data
Project Explorer Project View Name	G × Type Sinope [10.50.68.39] PDMM PLCopen Panel Panel Panel Panel KVB 2	PLCopen Axis Name: PLCOpen Axis 1 Type: Servo Number: 1 Servo Axis Servo Axis User Units: 360 UU Drive Interface Units: 10-48576 IU Update Rate: 250 usec ms Rollover Position: 0 UU
Project Explorer Dictionary Libraries		OK Cancel



For more details on PLcopen axis parameters, see "Axis Data Parameters" on page 158

12.5.15.3 Control Panel

The template contains a Control Panel which works inside of the IDE when running a project, and serves as a useful tool to run and debug programs. The Control Panel consists of two screens, named **Control Panel** and **Fault Report Panel**.

Control Panel consists of three sections.

- Multi-Axis Commands which controls motion and shows gearing
- Single Axis Motion Commands which shows motion at the single axis level and resetting the axis position
- Axis Status which shows basic axis information

For more details, see Design the Control Panel with the Internal Control Panel Editor

Project View		Conto or Parties
Name Typ		
🗸 📷 System Sinc		Machine Control
>	Copen	Example Ethercat Network Position: 4122.55 deg Position: 3258.64 deg Fault S Fault Enabled Fault Enabled Fault Enabled Fault S Fault Enabled Fault Fault Fault Fault Fault
EtherCAT References Fieldbus Control Panel Pan Point Panel Panel Point Panel Panel Point Panel Panel Point Panel Panel Point Panel Po	14	Multi-Axis Motion Axis 2 Syncet to Axis 1)
FoultReportPonel Pon KVBProject KVB	B 2	Construction Real of the second
		Impage Ceering Run/Stop Impage Ceering Run/Stop Impage Ceering Velocity Impage Ceering Velocity
		tisengage AcciDec degis/2 AcciDec degis/2
		ept_
		Abs Move deg deg deg
		¢
Project Explorer Dictionary Libraries		(65533.6533.0)

Figure 8-53: PLCopen Template - Control Panel

Call out#	Description
	Allows you to set the speed
2	Displays the actual position for each axis
3	Control the axis motion
4	Start or stop the motion on the condition that the axes are enable (the green light must be switched on)
5	Allows for enabling or disabling the axes. After an emergency stop, you need to select the Reset and Enable commands before running the axes
Table	e 8-39: PLCopen Template - Control Panel elements

Fault Report Panel provides more detailed information about when a drive fault occurs.

Axis 1	Axis 2
Fault 🔴	Fault 🔴
Fault Num	Fault Num
Fault Description	Fault Description
Axis 1 Position:	Axis 2 Position:
Position Error 0.0	Position Error .00240
Position Error Limit	Position Error Limit
Limit Exceeded 🔴	Limit Exceeded 🔶

Figure 8-54: PLCopen Template - Fault Report Panel

12.5.15.4 HMI

Also included in the project template are HMI screens for the AKI terminals created in the KVB (Kollmorgen Visualization Builder) environment.

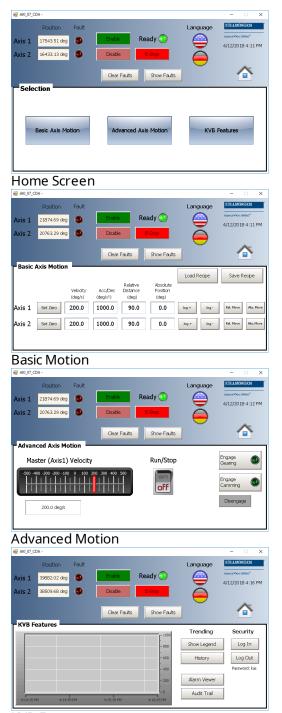
The KVB software is a separate installation from the KAS IDE. KVB can be opened from within KAS IDE by double-clicking on the **KVBProject** item in the project tree.

Project View	
Name	Туре
 Image System Controller Image PLC Image Motion Controller Onboard I/O EtherCAT Image References Image Fieldbus 	Sinope [127.0.0.1] PDMM or PCMM
FaultReportPanel Control Panel HMIDevice KVBProject	Panel Panel AKI Touchscreen KVB 2

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.



Four screens, localized in English and German, are included that duplicate the controls on the IDE Internal Control panel screens.



KVB Features

12.5.16 PLCopen 2-Axes Template with FFLD

This template project contains a two-axis template for getting started with a KAS application. It contains key project elements that will save development time. Additional axes can be added to the template applications that have three, four, five, or more axes. The templates are designed to easily add code to the programs and Control Panels that are in the template or to new ones. A project can contain many programs, and each program can be written in ST, SFC, FFLD, FBD, or IL programming languages. This template program can be run on the KAS Simulator or with a AKD PDMM or PCMM and AKD drives.

This project contains two axes where Axis 2 is slaved to Axis 1 at a 2:1 ratio.

12.5.16.1 PLC Programs

The 2-axes PLCopen template has a 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. Also included are controls to enable/disable the drives, clear faults, and display fault and axis position information in the Control Panel.

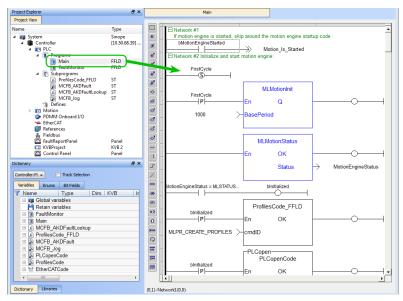


Figure 8-55: PLCopen Template with FFLD - Main

12.5.16.2 Motion

714

The template contains two PLCopen Servo axes where User Units, Update Rate, Rollover Position, and Axis Limits are defined as follows:

Project Explorer	8 ×	PLCopen Axis Data
Project View		
Name	Туре	Axis
🔺 📷 System	Sinope	Name: PLCOpenAxis1 Type: Servo 👻
Controller	[10.50.68.39] PDMM	Number: 1
 PLC Motion 		
- Profiles		
🔺 🎪 PLCopen	PLCopen	
PLCOpenAxis1		
Controller Onboard I/O		Servo Axis
🔫 EtherCAT		Axis Data Axis Limits
References # Fieldbus		
Fieldbus FaultReportPanel	Panel	User Units: 360 UU
Control Panel	Panel	
▲ I HMIDevice	AKI Touchscreen	Drive Interface Units: 1048576 IU
KVBProject	KVB 2	
		Update Rate: 250 usec 🔻 ms
		Rollover Position: 0 UU
		Rollover Position: 0 00
Project Explorer Dictionary Libraries		OK Cancel
Information and Logs		

Figure 8-56: PLCopen Template - Motion

For more details on PLcopen axis parameters, see "Axis Data Parameters" on page 158

12.5.16.3 Control Panel

The template contains a Control Panel which works inside of the IDE when running a project, and serves as a useful tool to run and debug programs. The Control Panel consists of two screens, named **Control Panel** and **Fault Report Panel**.

Control Panel consists of three sections.

- Multi-Axis Commands which controls motion and shows gearing
- **Single Axis Motion Commands** which shows motion at the single axis level and resetting the axis position
- Axis Status which shows basic axis information

For more details, see Design the Control Panel with the Internal Control Panel Editor

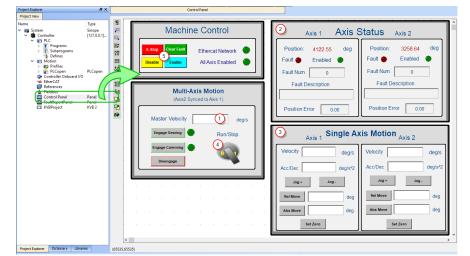


Figure 8-57: PLCopen Template - Control Panel

Call out#	Description
1	Allows you to set the speed
2	Displays the actual position for each axis
3	Control the axis motion
4	Start or stop the motion on the condition that the axes are enable (the green light must be switched on)
5	Allows for enabling or disabling the axes. After an emergency stop, you need to select the Reset and Enable commands before running the axes
Table 8-40: PLCopen Template - Control Panel elements	

Fault Report Panel provides more detailed information about when a drive fault occurs.

Axis 1	Axis 2
Fault 🔴	Fault 🔴
Fault Num	Fault Num
Fault Description	Fault Description
Axis 1 Position:	Axis 2 Position:
Position Error 0.0	Position Error .00240
Position Error Limit	Position Error Limit
Limit Exceeded 🛛 🔴	Limit Exceeded 🔶

Figure 8-58: PLCopen Template - Fault Report Panel

Based on the template, the project can be run:

- using the KAS Simulator
- with actual drives and motors (in this case, you first have to set up the axes in the EtherCAT part. For more details, see "Configuring EtherCAT" on page 320)

12.5.16.4 HMI

Also included in the project template are HMI screens for the AKI terminals created in the KVB (Kollmorgen Visualization Builder) environment.

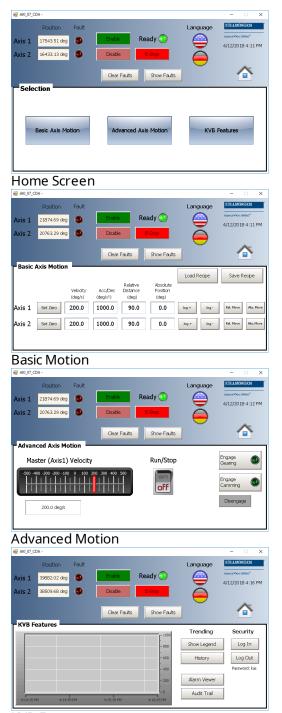
The KVB software is a separate installation from the KAS IDE. KVB can be opened from within KAS IDE by double-clicking on the **KVBProject** item in the project tree.



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.



Four screens, localized in English and German, are included that duplicate the controls on the IDE Internal Control panel screens.



KVB Features

12.5.17 Coordinated Motion 2-Axis Template

This project controls two axes in coordinated motion (PLCOpenAxis1 and PLCOpenAxis2).

12.5.17.1 Programs

The program is Sequential Function Chart (SFC) containing both Structured Text (ST) and Free Form Ladder Diagram (FFLD) code.

The first five steps of the SFC create and initialize the axes and the coordinated motion axes group. Step six 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.

12.5.17.2 Motion

To run the coordinated motion pattern, perform the following steps:

- 1. Download and start the application.
- 2. Press the "Enable" button to enable the axes and the axes group.
- 3. Press the "Home" button to move the axes to their zero position. (optional)
- 4. After the "Ready" light is illuminated, turn the "Cycle Start" switch to "1" and the axes will begin moving in programmed pattern.

12.5.17.3 Control Panel

#	Description	GUI
	Allows you to change the coordinated motion feedrate from 0% to 200%	1 Axes Group Feedrate Override
2	Displays the axis positions	0 25 50 75 100 125 150 175 200
3	Indicates that the three axes are enabled and the axes group is ready and enabled for coordinated motion	
4	Begins the coordinated motion pattern	Axis 1 Position: Axis 2 Position:
5	Causes an Emergency Stop for all axes. The axes will be disabled.	5 6
6	Resets the Emergency Stop and enables the axes	3 A Run/Stop E-Stop Reset 7 Disable Enable 8
	Disables the two axes	Home (9) (10
8	Enables the two axes and the axes group	
9	Moves all axes to their zero position	
10	Causes a reference by setting the axes' position to zero at their current location.	

12.5.18 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.

12.5.18.1 PLC Programs

The Coordinated Motion 3-Axis template has a Sequential Function Chart program (SFC) containing both Structured Text (ST) and Free Form Ladder Diagram (FFLD) code.

The first five steps 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 also performs basic moves for the third (Pipe Network) axis, to move down/up before and after the coordinated motion pattern.

12.5.18.2 Motion

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. As this template demonstrates, PLCopen axes that perform coordinated motion can be mixed with independent Pipe Network axes.

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.

To run the complete motion pattern:

- 1. Download and start the application.
- 2. Press the "Enable" button to enable the axes and the axes group.
- 3. Press the "Home" button to move the axes to their zero position. (optional)
- 4. After the "Ready" light is on, turn the "Cycle Start" switch to "1" and the axes will begin moving in the programmed pattern.

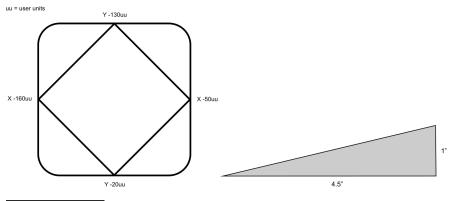
12.5.18.3 Control Panel

#	Description	GUI
1	Allows you to change the coordinated motion feedrate from 0% to 200% Displays the axis positions	1 Axes Group Feedrate Override 0 25 50 75 100 125 150 175 200 0%
3	Indicator lights. Ready indicates that the three axes are enabled and the axes group is ready and enabled for coordinated motion. Error Stopped indicates that the axes group has been stopped and disabled by the E-Stop button.	2 Axis 1 Position: 0.0000 Axis 2 Position: 0.0000
4	Begins the coordinated motion pattern	Vertical Axis Pos: 0.0000
5	Causes an Emergency Stop for all axes. The axes will be disabled.	Ready Run/Stop 5 6 E-Stop Reset
6	Resets the Emergency Stop and clears axis errors.	Error Stopped
$\overline{)}$	Disables the three axes	Home Ref (9) (10)
8	Enables the three axes and the axes group	
9	Moves all axes to their zero position	
10	Causes a reference by setting the axes' position to zero at their current location.	

12.5.19 Coordinated Motion 3-Axis (3D) Template

This project template controls three axes in coordinated motion (PLCOpenAxis1, PLCOpenAxis2 and PLCOpenAxis3) and 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.

The pattern and the platform for the plane are displayed below.



NOTE

This program was 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, click on the **Sim Mode** button on the Control Panel. This will cause the program to execute in simulator mode, which will bypass the homing sequence. Also, the axis scaling parameters (entered in the PLCopen Axis Data dialogs) may need to be changed to accommodate the different hardware.

12.5.19.1 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).

- The first five steps 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.

12.5.19.2 Motion

To run the complete motion pattern:

- 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" and the axes will begin moving in the programmed pattern.

12.5.19.3 Control Panel

#	Description
1	Allows you to change the coordinated motion feedrate from 0% to 200%
2	Displays the axis positions
3	Ready Indicator light - "Ready" indicates that the three axes are enabled, and the axes group is ready and enabled for coordinated motion.
4	Homed Indicator light: "Homed" indicates that each axis has been referenced, and the axis group has moved to its zero position.
5	Simulation Mode indicator light: "Sim Mode" indicates that the homing functions have been bypassed and 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 will be disabled. "Ready" and "Homed" lights will be turned off.
8	Resets the Emergency Stop and clears axis errors.
9	Disables the axes group. The axes have to be at standstill to disable the group
10	Enables the three axes and the axes group. This will turn on the "Ready" light.
11	Starts a homing function. Each axis will be referenced and the axis group will move to its zero position. Homing must be completed before executing a cycle start. When homing is complete, the "Homed" light will be turned on.
12	"Sim Mode" bypasses the homing function. The current position of each axis will be set to zero position. The homing function will not be performed. The "Sim Mode" light and "Homed" light will be turned on. This mode can be used when running on a simulator or when the hardware reference switches are not available.
	1 Axes Group Feedrate Override
	0 25 50 75 100 125 150 175 200
	0%
	2 X Axis Position: 0.0000
	Y Axis Position: 0.0000
	Z Axis Position: 0.0000
3	

Enable (10)

2

9

11

Disable

4

722

Homed

5 Sim Mode

13 PDMM and PCMM Errors and Alarms

When an Error or Alarm occurs, always check the controller log messages. The log messages will provide more details about the failure and the history of events leading up to the failure. From the log messages, you can determine the specifics about the cause of the failure to correct the underlying problem.

13.1 PCMM and PDMM Errors

Code	Description	Cause	Remedy	Clear‡
E01	Critical temperat ure exceeded. The controller operation is stopped after 20 seconds, CPU will be put to sleep.	CPU temperature exceeded safe operating temperature limit.	Power-off. Check airflow and operating environment are within hardware specifications. Allow unit to cool before power-on.	HW
E02	Out of memory. KAS runtime is stopping.	Memory leak, memory corrupted, or hardware memory failure.	Power-off/on. If problem is recurrent, check release notes for firmware updates or return hardware for repair.	HW
E03	Fan failure.	CPU cooling fan was not able to operate properly.	Check temperature and monitor for High temp alarm (see A01). Return hardware for fan replacement.	HW
E10	Firmware is corrupted.	Flash memory corrupted during firmware download or flash hardware failure.	Re-download firmware or boot into recovery mode, download firmware, and power-off/on. If problem persists, return hardware for repair.	SW
E11	Flash is corrupted, no filesystem is available.	At startup the filesystem could not be mounted on the flash.	Reset to factory defaults. If problem persists, return 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 for retained variables.	NVRAM is full.	Change application to reduce the amount of retained variables.	SW
E14	Reset to Factory Defaults failed.	Flash memory could not be formatted during a Reset to Factory Defaults procedure.	Try reset to factory defaults again from power-on. If problem persists, return hardware for repair.	SW

Code	Description	Cause	Remedy	Clear [‡]
E15	Cannot read/write files from/to a SD card	SD card is not plugged in or the file system is corrupt and cannot be mounted. PLC function failures will not cause this error.	Insert a valid SD card or reformat the SD card using Settings > SD Card > Format button.	SW
E16	Not enough space available on the SD card	SD card is full, unable to write to the SD card. PLC function failures.	Clean-up the SD card space by deleting files or re-format the card using Settings > SD Card > Format button.	SW
E17	Cannot connect to a shared directory	Improper shared directory Check if the computer's shared		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.	 Check the web server for other alarms or errors, such as temperature alerts, and resolve them. Power cycle the device If E20 is still being reported, reset to factory defaults. If the problem is recurrent, check release notes for firmware updates, or download firmware. 	ΗW
E21	Runtime process, thread, or driver failed to respond during operation.	KAS runtime code failed during normal operation.	Power-off/on. If problem is recurrent, check 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. See CPU Overload (E23).	Either the motion engine did not complete or the PLC program did not complete within the timeout period due to excessive CPU load.	Stop the application or power- off/on. Reduce the sample rate, simplify the application, or reduce the application cycles and restart the application.	SW

724

Code	Description	Cause	Remedy	Clear‡
E24	PLC application cannot be started	 PLC application cannot be started, due to an existing condition. Possible reasons: Maintenance operation is in progress. Controller is in Online Configuration Mode- AKD Restore failed. The IDE version of the compiled PLC code and controller runtime version do not match. Previous download failed. 	 Check the following: Controller web-server home page for any maintenance operation in-progress. Wait for the operation to finish. Connect to the controller with the IDE and disable <u>Online</u> Configuration Mode. 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 restore. IDE version (only major.minor.micro) should match with runtime version. To correct, install the correct version of IDE or Runtime. Connect 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 IDE Controller log for more information	 Read AKD 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 network operation failed due to a network communication error.	Check the EtherCAT network wiring and devices state. Re-start the application.	SW
E32	EtherCAT EtherCAT network operation communication failed due to a network failure during communication error. bootstrap mode.		Check the EtherCAT network wiring and devices state. Re-start the application.	SW
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 IDE Controller log for more information.	Check the EtherCAT network wiring and devices state. Re-start the application.	SW
E34	EtherCAT failed to initialize into preop mode.	EtherCAT network initialization failed due to a network communication error.	Check the EtherCAT network wiring and devices state. Re-start the application.	SW

Code	Description	Cause	Remedy	Clear‡
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	failed tofailed due to a mismatchwiring order. Correct the devdiscover thebetween the discovered andorder wiring or re-scan theexpectedexpected devices.network, re-compile, and dov		network, re-compile, and download the updated application. Re-start	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
E50	Backup to SD card failed	An unrecoverable error occurred during the backup operation.	Repeat the backup to SD card operation. If it fails again, replace the SD card.	SW
E51	Restore from SD card 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 hardware for repair.	SW
E52	SD Backup files The restore operation failed Perform a backup operation are missing or due to missing, incomplete, or before the restore or use and corrupt corrupt files on the SD card. SD card with valid backup files.		SW	
E53	SD Backup files are not compatible	Backup files on the SD card are not compatible.	Use a different SD card with Backup file(s) from a compatible controller.	SW

‡ Items labeled "SW" can be cleared from the web server. Items labeled "HW" require a reboot to be cleared.

13.2 PCMM and PDMM Alarms

Code	Description	Cause	Remedy	Clear ‡
A01	High temperature exceeded	CPU temperature near the safe operating temperature limit (120°C).	Check airflow and operating environment are within hardware specifications.	SW
A02	Low on memory.	Memory leak or corruption.	Power-off/on. If problem is recurrent, check release notes for firmware updates or return 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
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

726

Code	Description	Cause	Remedy	Clear ‡
A21	Recoverable process or thread failed to respond during operation.	KAS non-runtime code failed during normal operation and was automatically restarted.	If problem is recurrent, power-off/on. Check 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.	EtherCAT master was unable to send a frame for one or more cycles.	5	SW
A31	EtherCAT received a frame with a wrong Working Counter value.	 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. 	 Check the EtherCAT cables and the connectors. Check the EtherCAT devices for any device errors. 	
A38	ready. EtherCAT EtherCAT master did missed a not receive, or receive frame received too late, a during frame for one or more operation cycles. mode.		 Read AKD 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, or 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
A53	The Controller was replaced with a higher performance model.	The 1.2GHz model was restored using backup files from an 800MHz model.	Either replace the controller with an 800MHz model or use the functionally compatible, higher performance 1.2GHz model.	SW

‡ Items labeled "SW" can be cleared from the web server. Items labeled "HW" require a reboot to be cleared.

13.3 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 error will be flashed on the 7-segment display.

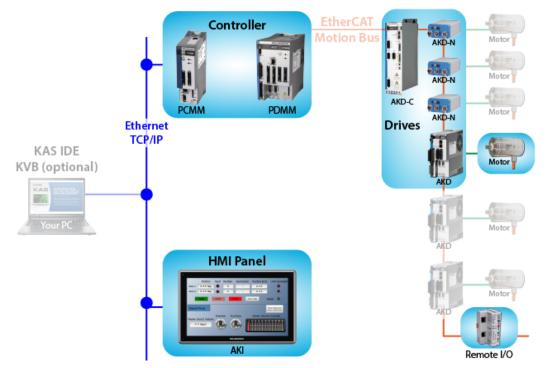
Process	Timeout
Motion Engine	200 milliseconds
PLC Program (VM)	10 seconds

The Real-Time operation for EtherCAT and the Motion Engine have the highest priority in the controller. The PLC Program (VM) has the second highest priority in the controller. These processes will 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 pushbuttons, web-server, Modbus, and communications with the KAS IDE. The 7-Segment will indicate a CPU overload or frozen software task by displaying one of the following patterns:

Display	Meaning
B	CPU overload is extreme.
	Motion thread is no longer able to execute.
B	EtherCAT Rx thread is no longer able to execute.
	PLC thread is no longer able to execute.
	Non-realtime threads are no longer able to execute.

To recover from an E23, stop the application from the 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. In this case, you will need to power-off/on the controller. If the PCMM or AKD PDMM is configured for Auto-start, press and hold the B3 menu button at boot-time to prevent the application from automatically re-starting. Then, you will be able to connect to the Controller with the IDE.



14 KAS Component Manuals

14.1 HMI

HMI Part Number	Description	KVB	Manual
AKI2G-CDA-MOD-05T-000	Graphical Display 7" TFT LCD, Touchscreen	v2.20	Ø
AKI2G-CDA-MOD-07T-000	Graphical Display 5" TFT LCD, Touchscreen	v2.20	Ø
AKI2G-CDB-MOD-07T-000	Graphical Display 7" TFT LCD, Touchscreen	v2.20	Ø
AKI2G-CDB-MOD-12T-000	Graphical Display 12" TFT LCD, Touchscreen	v2.20	Ø
AKI-CDC-MOD-12T-000	Graphical Display 12.1" TFT LCD, Touchscreen	v2.0	Ø
AKI-CDC-MOD-15T-000	Graphical Display 15.4" TFT LCD, Touchscreen	v2.0	Ø
AKI-CDC-MOD-21T-000	Graphical Display 21.5" TFT LCD, Touchscreen	v2.0	Ø

Table 9-1: List of KAS HMI

NOTE

Refer to our Web site for up-to-date information and material that are available in the Automation Component Solutions section.

Note that you first need to log in before accessing KAS Literature.

Scan this QR code to access our web site with your mobile device.



General Description	Part Number	630005105
Certifications	General	CE, FCC, KCC
	Marine	_
	UL	UL 61010-2-201
Mechanical	Mechanical size	170 x 107 x 49 mm
	Touch type	Resistive
	Cut-out size	161 x 93 mm
	Weight	0.5 kg
	Housing material	Plastic (PC+ABS), Gray
Power	Input Voltage	+24 V DC (18 - 32V DC). CE: The power supply must conform with the requirements according to IEC 60950 and IEC 61558-2-4. UL and cUL: The power supply must conform with the requirements for class II power supplies.
	Power consumption	6W
	Input fuse	Internal DC fuse
System	CPU	ARM9 400 MHz
	RAM	128 MB
	Flash	256 MB, 200 MB free for application storage
Display	Size, diagonal	5" diagonal
	Resolution	800 x 480 pixels
	Backlight	LED backlight
	Backlight life time	20,000 hours
	Backlight brightness	300 cd/m ²
	Backlight dimming	Industrial dimming
	Display type	TFT-LCD with LED backlight
	Display pixel error	Class 1 (ISO9241-307)
Communication, serial	Number of serial ports	2 Port 9pin DSUB
	Serial port 1	RS 232 (RTS/CTS)
	Serial port 2	RS 422/485
	Serial port 3	RS 232
	Serial port 4	RS 485
Communication, Ethernet	Number of Ethernet ports	1
	Ethernet port 1	1 x 10/100 Base-T (shielded RJ45)
	Ethernet port 2	_
Expansion interface	Expansion port	No

14.1.1 AKI2G-CDA-MOD-05T-000, 5" LCD Display

KAS IDE User Guide | 14 KAS Component Manuals

	SD card	No
	USB	1 x USB 2.0 500 mA
Environmental	Operating temperature	-10°C to +50°C
	Storage temperature	-20°C to +60°C
	Shock	15g, half-sine, 11ms according to IEC 60068-2-27
	Vibration	1g, according to IEC 60068-2-6, Test Fc
	Sealing front	IP65
	Sealing back	IP20
	Humidity	5% - 85% non-condensed

General Description	Part Number	630005205
Certifications	General	CE, FCC, KCC
	Marine	_
	UL	UL 61010-2-201
Mechanical	Mechanical size	196 x 146 x 52mm
	Touch type	Resistive
	Cut-out size	186 x 136 mm
	Weight	0.7 kg
	Housing material	Plastic (PC+ABS), Gray
Power	Input Voltage	+24 V DC (18 - 32V DC). CE: The power supply must conform with the requirements according to IEC 60950 and IEC 61558-2-4. UL and cUL: The power supply must conform with the requirements for class II power supplies.
	Power consumption	9.6W
	Input fuse	Internal DC fuse
System	CPU	ARM9 400 MHz
	RAM	128 MB
	Flash	256 MB, 200 MB free for application storage
Display	Size, diagonal	7" diagonal
	Resolution	800 x 480 pixels
	Backlight	LED backlight
	Backlight life time	20,000 hours
	Backlight brightness	400 cd/m ²
	Backlight dimming	Industrial dimming
	Display type	TFT-LCD with LED backlight
	Display pixel error	Class 1 (ISO9241-307)
Communication, serial	Number of serial ports	2 Port 9pin DSUB
	Serial port 1	RS 232 (RTS/CTS)
	Serial port 2	RS 422/485
	Serial port 3	RS 232
	Serial port 4	RS 485
Communication, Ethernet	Number of Ethernet ports	1
	Ethernet port 1	1 x 10/100 Base-T (shielded RJ45)
	Ethernet port 2	_

14.1.2 AKI2G-CDA-MOD-07T-000, 7" LCD Display

Expansion interface	Expansion port	No
	SD card	No
	USB	1 x USB 2.0 500 mA
Environmental	Operating temperature	-10°C to +50°C
	Storage temperature	-20°C to +60°C
	Shock	15g, half-sine, 11ms according to IEC 60068-2-27
	Vibration	1g, according to IEC 60068-2-6, Test Fc
	Sealing front	IP65
	Sealing back	IP20
	Humidity	5% - 85% non-condensed

	D-1410D-071-000	u, /" LCD Display
General Description	Part Number	630000205
Certifications	General	CE, FCC, KCC
	Marine	DNV, KR, GL, LR, ABS, CCS
	UL	UL 61010-2-201
Mechanical	Mechanical size	204 x 143 x 50 mm
	Touch type	Resistive
	Cut-out size	189 x 128 mm
	Weight	0.8 kg
	Housing material	Powder-coated aluminum, Gray
Power	Input Voltage	+24 V DC (18 - 32V DC). CE: The power supply must conform with the requirements according to IEC 60950 and IEC 61558-2-4. UL and cUL: The power supply must conform with the requirements for class II power supplies.
	Power consumption	14.4W
	Input fuse	Internal DC fuse
System	CPU	i.MX6Solo Single Cortex-A9 1.0GHz 512kB L2 cache
	RAM	512 MB
	Flash	2GB SSD (eMMC), 1.5GB free for application storage
Display	Size, diagonal	7" diagonal
	Resolution	800 x 480 pixels
	Backlight	LED backlight
	Backlight life time	20,000 hours
	Backlight brightness	350 cd/m ²
	Backlight dimming	Industrial dimming
	Display type	TFT-LCD with LED backlight
	Display pixel error	Class 1 (ISO9241-307)
Communication, serial	Number of serial ports	1 Port 9pin DSUB
	Serial port 1	RS 232 (RTS/CTS)
	Serial port 2	RS 422/485
	Serial port 3	RS 485 (only if COM 2 is RS 485)
Communication, Ethernet	Number of Ethernet ports	1
	Ethernet port 1	1 x 10/100 Base-T (shielded RJ45)
	Ethernet port 2	—

14.1.3 AKI2G-CDB-MOD-07T-000, 7" LCD Display

Expansion interface	Expansion port	Yes, ciX expansion module
	SD card	SD and SDHC
	USB	1 x USB 2.0 500 mA
Environmental	Operating temperature	-10°C to +60°C
	Storage temperature	-20°C to +70°C
	Shock	15g, half-sine, 11ms according to IEC 60068-2-27
	Vibration	1g, according to IEC 60068-2-6, Test Fc
	Sealing front	IP65, NEMA 4X/12 and UL Type 4X/12
	Sealing back	IP20
	Humidity	5% - 85% non-condensed

General Description	Part Number	640000205
Certifications	General	CE, FCC, KCC
	Marine	DNV, KR, GL, LR, ABS, CCS
	UL	UL 61010-2-201
Mechanical	Mechanical size	340 x 242 x 57mm
	Touch type	Resistive
	Cut-out size	324 x 226mm
	Weight	2.6 kg
	Housing material	Powder-coated aluminum, Gray
Power	Input Voltage	+24 V DC (18 - 32V DC). CE: The power supply must conform with the requirements according to IEC 60950 and IEC 61558-2-4. UL and cUL: The power supply must conform with the requirements for class II power supplies.
	Power consumption	28.8W
	Input fuse	Internal DC fuse
System	CPU	i.MX6DualLite, Dual Cortex-A9 1.0GHz 512kB L2 cache
	RAM	1 GB
	Flash	2GB SSD (eMMC), 1.5GB free for application storage
Display	Size, diagonal	12.1" diagonal
	Resolution	1280 x 800 pixels
	Backlight	LED backlight
	Backlight life time	50,000 hours
	Backlight brightness	400 cd/m ²
	Backlight dimming	Industrial dimming
	Display type	TFT-LCD with LED backlight
	Display pixel error	Class 1 (ISO9241-307)
Communication, serial	Number of serial ports	1 Port 9pin DSUB
	Serial port 1	RS 232 (RTS/CTS)
	Serial port 2	RS 422/485
	Serial port 3	RS 485 (only if COM 2 is RS 485)
	Serial port 4	1
Communication, Ethernet	Number of Ethernet ports	2
	Ethernet port 1	1 x 10/100 Base-T (shielded RJ45)
	Ethernet port 2	1 x 10/100 Base-T (shielded RJ45)

14.1.4 AKI2G-CDB-MOD-12T-000, 12" LCD Display

Expansion interface	Expansion port	Yes, ciX expansion module
	SD card	SD and SDHC
	USB	2 x USB 2.0 500 mA
Environmental	Operating temperature	-10°C to +60°C
	Storage temperature	-20°C to +70°C
	Shock	15g, half-sine, 11ms according to IEC 60068-2-27
	Vibration	1g, according to IEC 60068-2-6, Test Fc
	Sealing front	IP65, NEMA 4X/12 and UL Type 4X/12
	Sealing back	IP20
	Humidity	5% - 85% non-condensed

14.1.5 AKI-CDC-MOD-12T-000

Aluminum frame and casing in appealing design, powder-coated in steel-gray color, provides protection and gives a timeless look. The 4-color foil print further enhances the sense of depth in the operator panel. The front enclosure is IP65 classified, which means that it is waterproof even when exposed to washing. The operator panel is configured with the user-friendly software tool. With 1280x800 pixels, this high-resolution 12" TFT-display ensures optimal viewing in all conditions as well as wide viewing angles.

Electrical and Ha	rdware Specifications	AKI-CDC-MOD-12T
System Info	Processor	Intel® Celeron® B810E (2 × 1.6GHz), 2MBL2 Cache, Intel® QM67 Chipset <i>Optional</i> : Intel® Core [™] i3 2310E (2 × 2.1GHz) (Hyperthreading), 3MB L2 Cache, QM67 Chipset <i>Optional</i> : Intel® Core [™] i7 2715QE (4 × 2.1GHz) (Turbo 2.0, Hyperthreading), 6MB L2Cache, QM67 Chipset
	RAM	2 GB* / 4GB* DDR-3 SO-DIMM 1333MHz *depending on Processor Module
	Application Storage	64 GB 2.5" SSD
	Real time clock	Yes (on chip)
	Real time clock battery	Lithium battery type BR2032 (or CR 2032), exchangeable
	Operating System	Windows® 7
Display	Туре	TFT-LCD with LED backlight
	Size / Active display (W x H)	12" / 261.1 x 163.2mm
	Pixel count	1280 x 800 pixels
	Bit depth	24-bit, 16.7M colors
	VGA	1 ×VGA: resolution max. 2048 × 1536 @ 75Hz
	DVI	1 ×DVI-D single Link: Resolution max. 1600 × 1200 or 1920 × 1200 (with reduced blanking)
Interaction	Touchscreen material	Touch screen: Polyester on glass, resistive. Overlay: Autotex F157 or F207
	LED	1 x multi-color, software programmable
	External control	USB
Power	Power consumption at rated voltage	107 W
	Power supply	+24 V DC (18 - 32V DC). CE: The power supply must conform with the requirements according to IEC 60950 and IEC 61558-2-4. UL and cUL: The power supply must conform with the requirements for class II power supplies.
	Fuse	Internal DC fuse, 10 A Slow
Mechanical	Front panel, W x H x D	340 x 242 x 79mm
	Cut out dimensions	324 x 226mm
	Mounting depth	72mm (172 mm including clearance)
	Weight	4.2 kg
	Housing material	Powder-coated aluminum

Electrical and Ha	rdware Specifications	AKI-CDC-MOD-12T
	Standalone mounting	VESA 100x100
Communications	Ethernet	2 x 10 /100/1000 Base-T, shielded RJ 45
	Serial port RS422/RS485	9-pin D-sub contact with RS252 RTS/CTS, chassis- mounted female with standard locking screws 4-40 UNC
	Serial port RS232C	9-pin D-sub contact with RS252 RTS/CTS, chassis- mounted female with standard locking screws 4-40 UNC
Expansion	USB	4 x USB Host 2.0, max output current 500 mA
	Memory expansion	1 x SD. Only compatible with the standard SD format with up to 2GB storage capacity.
	Field buses (expansion modules)	1 x Extension Module Slot for Fieldbus (optional)
Certification	CE approvals	Noise tested according to EN61000-6-4emission and EN61000-6-2 immunity.
	UL, cUL approvals (when product or packing is marked)	UL 508
Environmental	Front /Rear seal	IP 65 / IP 20
	Relative operating humidity	5% - 85% non-condensed
	Operating temperature	0°C to +50° C
	Storage temperature	-20° to +70°C

14.1.6 AKI-CDC-MOD-15T-000

Aluminum frame and casing in appealing design, powder-coated in steel-gray color, provides protection and gives a timeless look. The 4-color foil print further enhances the sense of depth in the operator panel. The front enclosure is IP65 classified, which means that it is waterproof even when exposed to washing. The operator panel is configured with the user-friendly software tool. With 1280x800 pixels, this high-resolution 15" TFT-display ensures optimal viewing in all conditions as well as wide viewing angles.

Electrical and Hardware Specifications		AKI-CDC-MOD-15T
System Info	Processor	Intel® Celeron® B810E (2 × 1.6GHz), 2MBL2 Cache, Intel® QM67 Chipset <i>Optional</i> : Intel® Core™ i3 2310E (2 × 2.1GHz) (Hyperthreading), 3MB L2 Cache, QM67 Chipset <i>Optional</i> : Intel® Core™ i7 2715QE (4 × 2.1GHz) (Turbo 2.0, Hyperthreading), 6MB L2Cache, QM67 Chipset
	RAM	2 GB* / 4GB* DDR-3 SO-DIMM 1333MHz *depending on Processor Module
	Application Storage	64 GB 2.5" SSD
	Real time clock	Yes (on chip)
	Real time clock battery	Lithium battery type BR2032 (or CR 2032), exchangeable
	Operating System	Windows® 7
Display	Туре	TFT-LCD with LED backlight
	Size / Active display (W x H)	15" / 331.2 x 207.0mmmm
	Pixel count	1280 x 800 pixels
	Bit depth	24-bit, 16.7M colors
	VGA	1 ×VGA: resolution max. 2048 × 1536 @ 75Hz
	DVI	1 ×DVI-D single Link: Resolution max. 1600 × 1200 or 1920 × 1200 (with reduced blanking)
Interaction	Touchscreen material	Touch screen: Polyester on glass, resistive. Overlay: Autotex F157 or F207
	LED	1 x multi-color, software programmable
	External control	USB
Power	Power consumption at rated voltage	114 W
	Power supply	+24 V DC (18 - 32V DC). CE: The power supply must conform with the requirements according to IEC 60950 and IEC 61558-2-4. UL and cUL: The power supply must conform with the requirements for class II power supplies.
	Fuse	Internal DC fuse, 10 A Slow

Electrical and Ha Specifications	rdware	AKI-CDC-MOD-15T
Mechanical	Front panel, W x H x D	410 x 286 x 83mm
	Cut out dimensions	394 x 270mm
	Mounting depth	76mm (176 mm including clearance)
	Weight	5.4 kg
	Housing material	Powder-coated aluminum
	Standalone mounting	VESA 100×100
Communications	Ethernet	2 x 10 /100/1000 Base-T, shielded RJ 45
	Serial port RS422/RS485	9-pin D-sub contact with RS252 RTS/CTS, chassis-mounted female with standard locking screws 4-40 UNC
	Serial port RS232C	9-pin D-sub contact with RS252 RTS/CTS, chassis-mounted female with standard locking screws 4-40 UNC
Expansion	USB	4 x USB Host 2.0, max output current 500 mA
	Memory expansion	1 x SD. Only compatible with the standard SD format with up to 2GB storage capacity.
	Field buses (expansion modules)	1 x Extension Module Slot for Fieldbus (optional)
Certification	CE approvals	Noise tested according to EN61000-6-4emission and EN61000-6-2 immunity.
	UL, cUL approvals (when product or packing is marked)	UL 508
Environmental	Front /Rear seal	IP 65 / IP 20
	Relative operating humidity	5% - 85% non-condensed
	Operating temperature	0°C to +50° C
	Storage temperature	-20° to +70°C

14.1.7 AKI-CDC-MOD-21T-000

Aluminum frame and casing in appealing design, powder-coated in steel-gray color, provides protection and gives a timeless look. The 4-color foil print further enhances the sense of depth in the operator panel. The front enclosure is IP65 classified, which means that it is waterproof even when exposed to washing. The operator panel is configured with the user-friendly software tool. With 1920 x 1080 pixels, this high-resolution 21" TFT-display ensures optimal viewing in all conditions as well as wide viewing angles.

Electrical and Hardware Specifications		AKI-CDC-MOD-15T
System Info	Processor	Intel® Celeron® B810E (2 × 1.6GHz), 2MBL2 Cache, Intel® QM67 Chipset <i>Optional:</i> Intel® Core™ i3 2310E (2 × 2.1GHz) (Hyperthreading), 3MB L2 Cache, QM67 Chipset <i>Optional:</i> Intel® Core™ i7 2715QE (4 × 2.1GHz) (Turbo 2.0, Hyperthreading), 6MB L2Cache, QM67 Chipset
	RAM	2 GB* / 4GB* DDR-3 SO-DIMM 1333MHz *depending on Processor Module
	Application Storage	64 GB 2.5" SSD
	Real time clock	Yes (on chip)
	Real time clock battery	Lithium battery type BR2032 (or CR 2032), exchangeable
	Operating System	Windows® 7
Display	Туре	TFT-LCD with LED backlight
	Size / Active display (W x H)	21" / 473.6 x 268.1mm
	Pixel count	1920 x 1080pixels
	Bit depth	24-bit, 16.7M colors
	VGA	1 ×VGA: resolution max. 2048 × 1536 @ 75Hz
	DVI	1 ×DVI-D single Link: Resolution max. 1600 × 1200 or 1920 × 1200 (with reduced blanking)
Interaction	Touchscreen material	Touch screen: Polyester on glass, resistive. Overlay: Autotex F157 or F207
	LED	1 x multi-color, software programmable
	External control	USB
Power	Power consumption at rated voltage	125 W
	Power supply	+24 V DC (18 - 32V DC). CE: The power supply must conform with the requirements according to IEC 60950 and IEC 61558-2-4. UL and cUL: The power supply must conform with the requirements for class II power supplies.
	Fuse	Internal DC fuse, 10 A Slow

Electrical and Hardware Specifications		AKI-CDC-MOD-15T
Mechanical	Front panel, W x H x D	556 x 347 x 87mm
	Cut out dimensions	539 x 331mm
	Mounting depth	79mm (179 mm including clearance)
	Weight	8.1 kg
	Housing material	Powder-coated aluminum
	Standalone mounting	VESA 100×100
Communications	Ethernet	2 x 10 /100/1000 Base-T, shielded RJ 45
	Serial port RS422/RS485	9-pin D-sub contact with RS252 RTS/CTS, chassis-mounted female with standard locking screws 4-40 UNC
	Serial port RS232C	9-pin D-sub contact with RS252 RTS/CTS, chassis-mounted female with standard locking screws 4-40 UNC
Expansion	USB	4 x USB Host 2.0, max output current 500 mA
	Memory expansion	1 x SD. Only compatible with the standard SD format with up to 2GB storage capacity.
	Field buses (expansion modules)	1 x Extension Module Slot for Fieldbus (optional)
Certification	CE approvals	Noise tested according to EN61000-6-4emission and EN61000-6-2 immunity.
	UL, cUL approvals (when product or packing is marked)	UL 508
Environmental	Front /Rear seal	IP 65 / IP 20
	Relative operating humidity	5% - 85% non-condensed
	Operating temperature	0°C to +50° C
	Storage temperature	-20° to +70°C

14.1.8 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
- 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

14.2 Controllers

Description	Model Number	Main Characteristics
PCMM, 800 MHz single core	AKC-PCM-MC-080-00N-00-000	Standard Mutli-axis Controller
PCMM, 1.2 GHz single core	AKC-PCM-M1-120-00N-00-000	High Performance Multi-Axis Controller
PCMM, 1.2 GHz dual core	AKC-PCM-M2-120-00N-00-000	High Performance Multi-Axis Controller
AKD PDMM, 800 MHz	AKD-M0xxxx-MCEC-0000	Standard Drive Resident Controller
AKD PDMM, 1.2 GHz	AKD-M0xxxx-M1EC-0000	High Performance Drive Resident Controller

Table 9-2: List of KAS Controllers

14.2.1 PCMM Specifications

The PCMM programmable motion controller delivers a small yet powerful and cost-effective hardware platform ideally suited for modular or stand-alone machines that want the maximum in flexibility and performance.

Processor	800MHz (MC model), 1.2GHz (M1 model), or dual-core 1.2GHz (M2 model)
Internal Memory	64 MB Flash memory for program storage
External Memory	Removable SD Card slot (SD Card not included)
Input Power	24 Vdc @ 1.25 A
Operating Temperature	0 °C - 55 °C
High Temperature Limit	120°C
Critical Temperature Limit	127°C
Sealing	IP20
Local I/O	6 digital inputs, 2 digital outputs
Motion Network	EtherCAT, max 4kHz update rate
Networking Protocols	100BaseT connection supporting UDP, HTTP, Modbus TCP, Ethernet/IP, Profinet

Dimensions	174 mm (H) x 50 mm (W) x 111.5 mm (D)
Certifications	CE, UL

14.2.2 AKD PDMM Specifications

The AKD PDMM combines an AKD servo drive with a powerful, embedded motion controller in a single, compact package that can control up to 64 axes on your machine. It provides integrated control for multiple high-performance axes, complete I/O and HMI interfaces.

Model	Voltage	Continuous Current (Arms)	Peak Current (Arms)	Dimensions
AKD-M00306- xxEC-0000	120/240 VAC 1- and 3-Phase	3	9	168x89x156
AKD-M00606- xxEC-0000	120/240 VAC 1- and 3-Phase	6	18	168x89x156
AKD-M01206- xxEC-0000	120/240 VAC 1- and 3-Phase	12	30	192x107x187
AKD-M02406- xxEC-0000	120/240 VAC 1- and 3-Phase	24	48	248x96x228
AKD-M00307- xxEC-0000	240/400/480 VAC 3- Phase	3	9	256x99x185
AKD-M00607- xxEC-0000	240/400/480 VAC 3- Phase	6	18	256x99x185
AKD-M01207- xxEC-0000	240/400/480 VAC 3- Phase	12	30	256x99x185
AKD-M02407- xxEC-0000	240/400/480 VAC 3- Phase	24	48	306x99x228
AKD-M04807- xxEC-0000	240/400/480 VAC 3- Phase	48	96	385x185x225

Processor	800MHz (MCEC model) or 1.2GHz (M1EC model)
Internal Memory	64 MB Flash memory for program storage
External Memory	Removable SD card slot (not included)
Input Power	24 Vdc @ 1.25 A
Operating Temperature	0 °C - 55 °C
High Temperature Limit	120°C
Critical Temperature Limit	127°C
Sealing	IP20
Local I/O	6 digital inputs, 2 digital outputs
Motion Network	EtherCAT, max 4kHz update rate
Networking Protocols	100BaseT connection supporting UDP, HTTP, Modbus TCP, Ethernet/IP, Profinet
Dimensions	174 mm (H) x 50 mm (W) x 111.5 mm (D)
Certifications	CE, UL

14.2.3 NVRAM

746

KAS uses the NVRAM (non-volatile memory) to save Retain Variables.

Hardware Type	NVRAM Size Allocation
Simulator	128 Kbytes
AKD PDMM or PCMM	32 Kbytes

Table 9-3: NVRAM Size Depending on Hardware

IMPORTANT

Part of the NVRAM allocation is reserved to store some internal data (144 bytes). As a consequence, not all 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 2 seconds when the application is running, and when you shutdown the application.

Life expectancy of the NVRAM

- Data Retention: At maximum operating temperature, the data written to the memory will be retained for 10 years.
- Endurance: The NVRAM memory has an endurance limit of 1014 read/write accesses, (or more than 6,000,000 years of being accessed every 2 seconds).

NOTE

Using the retain variables is highly cycle time consuming. As a consequence, Kollmorgen strongly recommends to carefully monitor the system load with the TraceTimes command.

14.2.3.1 How can I check the NVRAM space is enough to store my retain variables?

To calculate the NVRAM space, you have to add the size of each retained variable according to:

- its data type as described here
- the numbers of elements in case you declare the variable as an array

Do not forget to add the 144 bytes as stated in the Warning above.

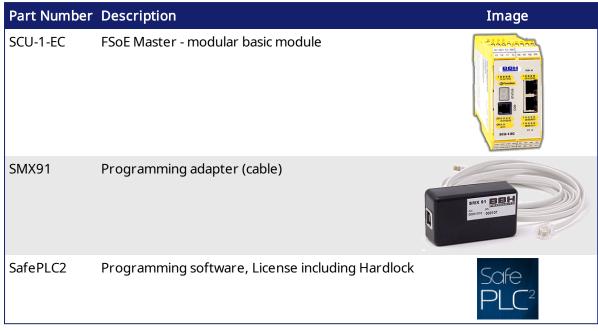
In the following example, the total size is: **3684 bits** (which is less than 0.5 Kbytes)

Pictionary		
Controller:PLC	*	Track Selection
)A Name	🛆 Type	Dim.
🗉 🧊 Global variabl	es	
🗆 💾 Retain variabl	es	
bLedStatus	BOOL	[03]
MasterAbsPos	LREAL	
MyString	STRING	6(100) [02]
TravelSpeed	LREAL	
💼 Main		

Variable	Size / element	Element no.	Total Size / variable
bLedStatus	1 bit	4	4
MasterAbsPos	64 bits	1	64
MyString	800 bits (100 bytes)	3	2400
TravelSpeed	64 bits	1	64
Internal data	1152 bits (144 bytes)	na	1152

14.3 BBH FSoE Master

The following BBH Safety equipment is required to program and operate the SCU-1-EC FSoE Safety master with the AKD2G, AKT2G-SDI-004-000, AKT2G-SDO-004-000, PCMM/AKD PDMM, and/or KAS IDE software.



These items are available for order from Kollmorgen.

14.4 Kollmorgen Safety Solutions

KSM-compact Safety Module

- For 1 or 2 safe axes KSM 11
- Up to 2 expansion modules KSM12
- Base unit with 16 safe I/O KSM 31
- KSM 21R • Expandable up to 60 safe I/O
- KSM 5x
- 1 safe relay output, expandable • 2 pulse outputs, 2 signal outputs
- Expandable up to 6 pulse and 6 signal ouputs
- Up to 800 function blocks
- Space-saving, compact design

KSM-modular Safety PLC

- Up to 12 safe axes KSM 100-1
 - Up to 8 expansion modules
- KSM 100-2
 KSM 100-2
 Base unit with 56 safe I/O
- KSM 121/-2 Expandable up to 200 safe I/O KSM 122/-2 • 1 safe relay output, expandable
- KSM 122A
 - 2 pulse outputs, up to 10 signal outputs
 - Expandable up to 14 pulse and 22 signal outputs
- KSM 131 • Up to 3000 function blocks KSM 131R
 - For applications with large number of interfaces

Available Safety Functions (PLe & SIL3)

- Safe Stop Function: STO, SS1, SS2, SOS
- Safe Velocity Function: SLS, SSM, SSR, SMS
- Safe Position Function: SLP, SCA, SLI
- Safe Direction Function: SDI
- Safe Brake Function: SBC

Learn more about the safety functions on the Kollmorgen Developer Network.

Connectivity

- EtherCAT
- CANopen
- Profinet
- PROFIsafe and EtherCAT FSoE slave

14.5 Mounting and Wiring of I/O Terminals

14.5.1 Instructions for ESD Protection

NOTE

Desctruction of the devices by electrostatic discharge possible!

The devices contain components at risk from electrostatic discharge caused by improper handling.

- Please ensure you are electrostatically discharged and avoid touching the contacts of the device directly.
- Avoid contact with highly insulating materials (synthetic fibers, plastic film, etc.).
- Surroundings (working place, packaging, and personnel) should be grounded properly when handling the devices.
- Each assembly must be terminated at the right hand end with an AKT2G-EM-000-000 bus end cap to ensure the protection class and ESD protection.



Figure 9-1: Spring contacts of the I/O components.

14.5.2 Installation on mounting rails

Risk of electric shock and damage of device!

Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the bus terminals!

14.5.2.1 Assembly

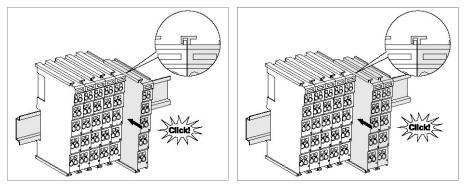


Figure 9-2: Attaching on mounting rail

The bus coupler and bus terminals are attached to commercially available 35 mm mounting rails (DIN rails according to EN 60715) by applying slight pressure:

- 1. First attach the fieldbus coupler to the mounting rail.
- 2. The bus terminals are now attached on the right-hand side of the fieldbus coupler. Join the components with tongue and groove and push the terminals against the mounting rail, until the lock clicks onto the mounting rail.

If the terminals are clipped onto the mounting rail first and then pushed together without tongue and groove, the connection will not be operational! When correctly assembled, no significant gap should be visible between the housings.

★ TIP

Fixing of mounting rails

The locking mechanism of the terminals and couplers extends to the profile of the mounting rail. At the installation, the locking mechanism of the components must not come into conflict with the fixing bolts of the mounting rail. To mount the mounting rails with a height of 7.5 mm under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

14.5.2.2 Disassembly

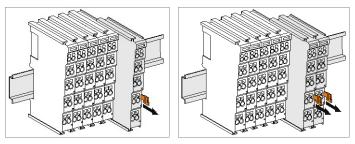


Figure 9-3: Disassembling of terminal

Each terminal is secured by a lock on the mounting rail, which must be released for disassembly:

- 1. Pull the terminal by its orange-colored lugs approximately 1 cm away from the mounting rail. In doing so for this terminal the mounting rail lock is released automatically and you can pull the terminal out of the bus terminal block easily without excessive force.
- 2. Grasp the released terminal with thumb and index finger simultaneous at the upper and lower grooved housing surfaces and pull the terminal out of the bus terminal block.

14.5.2.3 Connections within a bus terminal block

The electric connections between the Bus Coupler and the Bus Terminals are automatically realized by joining the components:

- The six spring contacts of the E-Bus/K-Bus deal with the transfer of the data and the supply of the Bus Terminal electronics.
- The power contacts deal with the supply for the field electronics and thus represent a supply rail within the bus terminal block. The power contacts are supplied via terminals on the Bus Coupler (up to 24 V) or for higher voltages via power feed terminals.

NOTE

Power Contacts

During the design of a bus terminal block, the pin assignment of the individual Bus Terminals must be taken account of, since some types (e.g. analog Bus Terminals or digital 4-channel Bus Terminals) do not or not fully loop through the power contacts. Power Feed Terminals (AKT2G-PSF-024-000) interrupt the power contacts and thus represent the start of a new supply rail.

14.5.2.4 PE power contact

The power contact labeled PE can be used as a protective earth. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

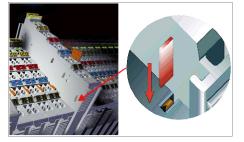


Figure 9-4: Power contact on left side

NOTE

Possible damage of the device

Note that, for reasons of electromagnetic compatibility, the PE contacts are capacitatively coupled to the mounting rail. This may lead to incorrect results during insulation testing or to damage on the terminal (e.g. disruptive discharge to the PE line during insulation testing of a consumer with a nominal voltage of 230 V). For insulation testing, disconnect the PE supply line at the Bus Coupler or the Power Feed Terminal! In order to decouple further feed points for testing, these Power Feed Terminals can be released and pulled at least 10 mm from the group of terminals.

752

Risk of electric shock!

The PE power contact must not be used for other potentials!

14.5.3 Installation instructions for enhanced mechanical load capacity

Risk of injury through electric shock and damage to the device!

Bring the Bus Terminal system into a safe, de-energized state before starting mounting, disassembly or wiring of the Bus Terminals!

14.5.3.1 Additional checks

The terminals have undergone the following additional tests:

Verification	Explanation
Vibration	10 frequency runs in 3 axes
	6 Hz < f < 60 Hz displacement 0.35 mm, constant amplitude
	60.1 Hz < f < 500 Hz acceleration 5 g, constant amplitude
Shocks	1000 shocks in each direction, in 3 axes
	25 g, 6 ms

14.5.3.2 Additional installation instructions

For terminals with enhanced mechanical load capacity, the following additional installation instructions apply:

- The enhanced mechanical load capacity is valid for all permissible installation positions
- Use a mounting rail according to EN 60715 TH35-15
- Fix the terminal segment on both sides of the mounting rail with a mechanical fixture, e.g. an earth terminal or reinforced end clamp
- The maximum total extension of the terminal segment (without coupler) is: 64 terminals (12 mm mounting with) or 32 terminals (24 mm mounting with)
- Avoid deformation, twisting, crushing and bending of the mounting rail during edging and installation of the rail
- The mounting points of the mounting rail must be set at 5 cm intervals
- Use countersunk head screws to fasten the mounting rail
- The free length between the strain relief and the wire connection should be kept as short as possible. A distance of approx. 10 cm should be maintained to the cable duct.

14.5.4 Connection

14.5.4.1 Connection system

Risk of electric shock and damage of device!

Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the bus terminals!

14.5.4.1.1 Overview

The Bus Terminal system offers different connection options for optimum adaptation to the respective application:

• The terminals of AKT2G and AKT series with standard wiring include electronics and connection level in a single enclosure.

14.5.4.1.2 Standard wiring (AKT2G/AKT-xx)



Figure 9-5: Standard wiring

The terminals of AKT2G and AKT series have been tried and tested for years.

They feature integrated screwless spring force technology for fast and simple assembly.

14.5.4.2 Wiring

754

Risk of electric shock and damage of device!

Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

14.5.4.2.1 Terminals for standard wiring AKT2G-xx/AKT-xx

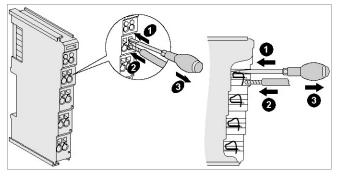


Figure 9-6: Connecting a cable on a terminal point

Up to eight terminal points enable the connection of solid or finely stranded cables to the Bus Terminal. The terminal points are implemented in spring force technology. Connect the cables as follows:

- 1. Open a terminal point by pushing a screwdriver straight against the stop into the square opening above the terminal point. Do not turn the screwdriver or move it alternately (don't toggle).
- 2. The wire can now be inserted into the round terminal opening without any force.
- 3. The terminal point closes automatically when the pressure is released, holding the wire securely and permanently.

See the following table for the suitable wire size width.

Terminal housing	AKT2G _{-XX} , AKT-XX
Wire size width (single core wires)	0.08 2.5 mm ²
Wire size width (fine-wire conductors)	0.08 2.5 mm ²
Wire size width (conductors with a wire end sleeve)	0.14 1.5 mm ²
Wire stripping length	8 9 mm

14.5.4.3 Shielding

NOTE

Shielding

Encoder, analog sensors and actors should always be connected with shielded, twisted paired wires.

Observe the special conditions for the intended use of Kollmorgen fieldbus components with extended temperature range (ET) in potentially explosive areas (directive 2014/34/EU)!

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60079-15! The environmental conditions during use are thereby to be taken into account!
- For dust (only the fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9): The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to EN 60079-0 for group IIIA or IIIB and IP6X for group IIC, taking into account the environmental conditions under which the equipment is used.
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of -25 to 60°C for the use of Kollmorgen fieldbus components with extended temperature range (ET) in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The fuses of the AKT2G-PSF-024-000 power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

14.5.5 Installation positions

NOTE

Constraints regarding installation position and operating temperature range

Please refer to the technical data for a terminal to ascertain whether any restrictions regarding the installation position and/or the operating temperature range have been specified. When installing high power dissipation terminals ensure that an adequate spacing is maintained between other components above and below the terminal in order to guarantee adequate ventilation!

14.5.5.1 Optimum installation position (standard)

The optimum installation position requires the mounting rail to be installed horizontally and the connection surfaces of the AKT2G terminals to face forward (see Figure 9-7: Recommended distances for standard installation position). The terminals are ventilated from below, which enables optimum cooling of the electronics through convection. "From below" is relative to the acceleration of gravity.

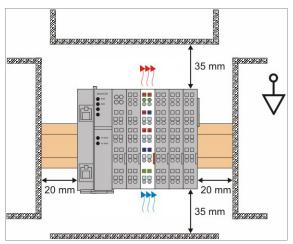


Figure 9-7: Recommended distances for standard installation position

Compliance with the distances shown in Fig. Recommended distances for standard installation position is recommended.

14.5.5.2 Other installation positions

All other installation positions are characterized by different spatial arrangement of the mounting rail - see Figure 9-8: Other installation positions.

The minimum distances to ambient specified above also apply to these installation positions.

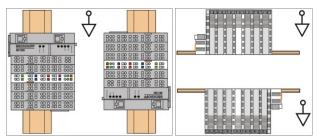


Figure 9-8: Other installation positions

756

14.5.6 Positioning of passive Terminals

★ TIP

Hint for positioning of passive terminals in the bus terminal block

EtherCAT Terminals (AKT2G-xx), which do not take an active part in data transfer within the bus terminal block are so called passive terminals. The passive terminals have no current consumption out of the E-Bus.

To ensure an optimal data transfer, you must not directly string together more than two passive terminals!

14.5.6.1 Examples for positioning of passive terminals (highlighted)

ſ		0			F	<u> </u>	F	F	F	p - 4	P	
	0 0				0000	0000	0000	0000	0000	0000		
	$\overline{(}$	Us 24V 0V ① ⑤	88	88	88	88	88	88	88	88	88	
		24V Up 2-6	88	88	88	88	88	88	88	88	88	
\odot		0V Up 3-7	88	88	88	88	88	88	88	88	88	\otimes
		4-3	88	88	88	88	88	88	88	88	88	
l			EK1100 BECKHOFF		EL9186 BECKHOFF	EL9187 BECKHOFF		EL9186 BECKHOFF	EL9187 BECKHOFF	EL2004 BECKHOFF	EL9011 BECKHOFF	

Figure 9-9: Correct positioning

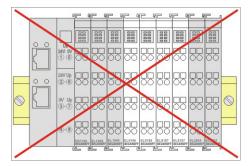


Figure 9-10: Incorrect positioning

14.5.7 UL Notice

Application

US Kollmorgen EtherCAT modules are intended for use with Kollmorgen's UL Listed EtherCAT System only.

Examination

US For cULus examination, the Kollmorgen I/O System has only been investigated for risk of fire and electrical shock (in accordance with UL508 and CSA C22.2 No. 142).

For devices with Ethernet connectors

US Not for connection to telecommunication circuits.

14.5.7.1 Basic principles

UL certification according to UL508. Devices with this kind of certification are marked by this sign:



14.5.8 Continuative documentation about explosion protection



Explosion protection for terminal systems

Pay also attention to the continuative documentation

14.5.9 Safety Instructions

14.5.9.1 Delivery state

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Kollmorgen.

14.5.9.2 Operator's obligation to exercise diligence

The operator must ensure that

- the safety products are only used as intended (see chapter Product description);
- the safety products are only operated in sound condition and in working order.
- the safety products are operated only by suitably qualified and authorized personnel.
- the personnel is instructed regularly about relevant occupational safety and environmental protection aspects, and is familiar with the operating instructions and in particular the safety instructions contained herein.
- the operating instructions are in good condition and complete, and always available for reference at the location where the safety products are used.
- none of the safety and warning notes attached to the safety products are removed, and all notes remain legible.

14.5.9.3 Description of safety symbols

In these operating instructions the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

758

A DANGER	Serious risk of injury!
A DANGER	Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.
WARNING	Risk of injury!
	Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.
	Personal injuries!
	Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.
NOTICE	Damage to the environment or devices
Attention	Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.
NOTE	Tip or pointer
	This symbol indicates information that contributes to better understanding.

14.5.10 Safety Operation

14.5.10.1 Environmental Conditions

Please ensure that the safety components are only transported, stored and operated under the specified conditions (see technical data)!

Risk of injury!

The safety components must not be used under the following operating conditions.

- under the influence of ionizing radiation (that exceeds the level of the natural environmental radiation)
- in corrosive environments
- in an environment that leads to unacceptable soiling of the safety component

NOTICE

Electromagnetic compatibility

The safety components comply with the current standards on electromagnetic compatibility with regard to spurious radiation and immunity to interference in particular.

However, in cases where devices such as mobile phones, radio equipment, transmitters or highfrequency systems that exceed the interference emissions limits specified in the standards are operated near safety components, the function of the safety components may be impaired.

14.5.10.2 Safety Instructions

Before installing and commissioning the safety components please read the Safety Instructions in this documentation.

Commissioning Test

Before the AKT2G-SDI-004-000/AKT2G-SDO-004-000 can be used for the safety task, the user must carry out a commissioning test so that sensor and actuator wiring errors can be ruled out.

14.5.10.3 Transport / Storage

Use the original packaging in which the components were delivered for transporting and storing the safety components.

Note the specified environmental conditions

Please ensure that the digital safety components are only transported and stored under the specified environmental conditions (see technical data).

14.5.10.4 Mechanical Installation

Risk of injury!

Bring the bus system into a safe, de-energized state before starting installation, disassembly or wiring of the devices!

Use ferrules with plastic collars

When using fine-wire cables for signal connections, use ferrules with plastic collars. This leads to a higher system availability when test pulses for the corresponding channels are switched off.

14.5.10.4.1 Control cabinet / terminal box

The safety terminals must be installed in a control cabinet or terminal box with IP54 protection class according to IEC 60529 as a minimum.

14.5.10.4.2 Installation position and minimum distances

For the prescribed installation position the mounting rail is installed horizontally and the mating surfaces of the terminals point toward the front (see illustration below). The terminals are ventilated from below, which enables optimum cooling of the electronics through convection. The direction indication "down" corresponds to the direction of positive acceleration due to gravity.

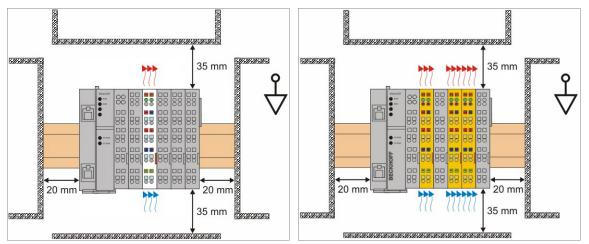


Figure 9-11: Installation position and minimum distances

In order to ensure optimum convection cooling, the distances to neighboring devices and to control cabinet walls must not be smaller than those shown in the diagram.

NOTE

External heat sources / radiant heat / impaired convection

The maximum permissible ambient temperature of 55°C was checked with the above example configuration. Impaired convection, an unfavorable location near heat sources or an unfavorable configuration of the EtherCAT Terminals may result in overheating of the terminals.

The key parameter is always the maximum permitted internally measured temperature of 95°C, above which the safety terminals switch to safe state and report an error. The internal temperature can be read from the safety components via CoE.

14.5.11 Safety Terminal Reaction Times

The safety terminals form a modular safety system that exchanges safety-oriented data via the Safety-over-EtherCAT protocol. This topic is intended to help you determine the system's reaction time from the change of signal at the sensor to the reaction at the actuator.

14.5.11.1 Typical Reaction Time

The typical reaction time is the time that is required to transmit information from the sensor to the actuator, if the overall system is working without error in normal operation.

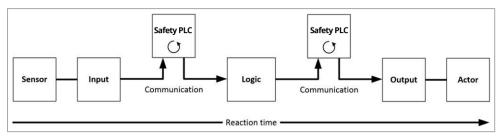


Figure 9-12: Typical reaction time

Definition	Description
RTSensor	Reaction time of the sensor until the signal is provided at the interface. Typically supplied by the sensor manufacturer.
RTInput	Reaction time of the safe input, such as AKT2G-SDI-004. This time can be found in the technical data. In the case of the AKT2G-SDI-004 it is 4 ms.
RTComm	Reaction time of the communication This is typically 3x the EtherCAT cycle time, because new data can only be sent in a new Safety-over-EtherCAT telegram. These times depend directly on the higher-level standard controller (cycle time of the PLC).
RTLogic	Reaction time of the logic terminal. This is the cycle time of the safety PLC, depending on the size of the safety project.
RTOutput	Reaction time of the output terminal. This typically lies within the range of 2 to 3 ms.
RTActor	Reaction time of the actuator. This information is typically supplied by the actuator manufacturer
WDComm	Watchdog time of the communication

This results in the following equation for the typical reaction time:

 $ReactionTime_{typ} = RT_{Sensor} + RT_{Input} + 3 * RT_{comm} + RT_{Logic} + 3 * RT_{comm} + RT_{output} + RT_{Actor}$

with, for example

 $ReactionTime_{typ} = 5ms + 4ms + 3 * 1ms + 10ms + 3 * 1ms + 3ms + 20ms = 48ms$

14.5.11.2 Worst-Case Reaction Time

The worst case reaction time is the maximum time required to switch off the actuator in the case of an error.

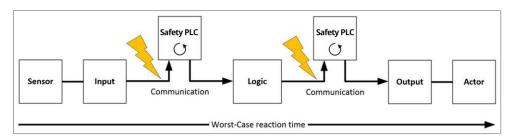


Figure 9-13: Worst-case reaction time

This assumes that a signal change occurs at the sensor and is transmitted to the input. A communication error occurs at precisely the moment when the signal is to be transferred to the communication interface. This is detected by the logic following the watchdog time of the communication link. This information should then be transferred to the output, but a further communication error occurs here. This error is detected at the output following the expiry of the watchdog time and leads to the switch-off.

This results in the following equation for the worst-case reaction:

 $ReactionTime_{max} = WD_{comm} + WD_{comm} + RT_{Actor}$

with, for example

ReactionTime_{max} = 2 * 15ms + 20ms = 50ms

14.5.12 Safety I/O Maintenance

Maintenance

The safety components are maintenance-free!

Environmental conditions

Observe the specified environmental conditions!

Please ensure that the safety components are only stored and operated under the specified conditions (see technical data).

If the safety component is operated outside the permitted temperature range it will switch to **Global Shutdown** state.

Cleaning

Protect the safety component from unacceptable soling during operation and storage!

If the safety component was subjected to unacceptable soiling it may no longer be operated!

Have soiled terminals checked!

Cleaning of the safety component by the user is not permitted!

Please send soiled terminals to the manufacturer for inspection and cleaning!

14.5.13 Interference-Free EtherCAT Terminals

Use of interference-free EtherCAT Terminals in safety applications

If an EtherCAT Terminal is described as interference-free, this means that the consecutive terminal behaves passively in a safety application (e.g. in the case of the all-pole switch-off of a po- tential group).

In this case the terminals do not represent an active part of the safety controller and do not affect the Safety Integrity Level (SIL) or Performance Level (PL) attained in the safety application.

14.6 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

Please find in the table below the list of each I/O component available.

AKT2G (EtherCAT) Terminals

I/O Terminal Part Number	I/O Terminal Description					
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, pres breakage detection, 16 bit	4-channel thermocouple input terminal, preset to type K, with wire breakage detection, 16 bit				
AKT2G-AN-430-000	4-channel analog input, parameterisable, -10/0+10 V, -20/0/+4+20 mA, 16 bit					
AKT2G-AT-410-000	4-channel analog output terminal 010 V, 12	4-channel analog output terminal 010 V, 12 bit, 1-wire system				
AKT2G-AT-425-000	4-channel analog output terminal -10 V+10	4-channel analog output terminal -10 V+10 V, 12 bit, 4 x 2-wire system				
AKT2G-DN-002-000	Up/down counter 24 V DC, 100 kHz, 32 bit co	unter	depth			
AKT2G-DN-008-000	8-channel digital input terminal 24 V DC, filter	⁻ 3.0 n	ns, 1-wire system			
AKT2G-DNH-008-00	0 8-channel digital input terminal 24 V DC, filter	⁻ 10 µs	s, 1-wire system			
AKT2G-DT-008-000	8-channel digital output terminal 24 V DC, 0.5	5 A, 1-	wire system			
AKT2G-ECT-000-000	EtherCAT Coupler for E-bus terminals	EtherCAT Coupler for E-bus terminals				
AKT2G-EM-000-000	Bus end cover for E-bus terminals, cover for power and E-bus contacts, grey					
AKT2G-ENC-180-000	1-channel incremental encoder interface, 32 bit					
AKT2G-ENC-190-000	Incremental encoder interface with differential input, 16/32 bit					
AKT2G-PSF-024-000	Power supply terminal with fuse, 24 V DC					
AKT2G-SDI-004-000	4-channel digital input terminal, Safety, 24 V DC					
AKT2G-SDO-004-000		/ DC, (0.5 A			
AKT (K-Bus) Terminal						
I/O Terminal Part Number	Description	Link	Replaced By			
AKT-AN-200-000	2 Channel Thermocouple Input Module	Ø	AKT2G-AN-240-000			
AKT-AN-400-000	4 Channel Thermocouple Input Module	Ø	AKT2G-AN-400-000			
AKT-AN-410-000	4 Channel Analog Input Module, 0-10 VDC	Ø	AKT2G-AN-430-000			
AKT-AN-420-000	4 Channel Analog Input Module, 0-20 mA	Ø	AKT2G-AN-430-000			
AKT-AN-810-000	8 Channel Analog Input Module, 0-10 VDC	Ø	AKT2G-AN-430-000			
AKT-AN-820-000	8 Channel Analog Input Module, 0-20 mA	Ø	AKT2G-AN-430-000			
AKT-AT-220-000	2 Channel Analog Output Module, 0-20 mA	Ø	—			
AKT-AT-410-000	4 Channel Analog Output Module, 0-10 VDC	Ø	-			
AKT-AT-420-000	4 Channel Analog Output Module, 0-20 mA	Ø	_			
AKT-AT-810-000	8 Channel Analog Output Module, 0-10 VDC	Ø	AKT2G-AT-410-000			

I/O Terminal Part Number	Description	Link	Replaced By
AKT-AT-820-000	8 Channel Analog Output Module, 0-20 mA	Ø	—
AKT-DN-004-000	4 Channel Digital Input Module, 24 VDC 3ms	Ø	AKT2G-DN-008-000
AKT-DN-008-000	8 Channel Digital Input Module, 24 VDC 3ms	Ø	AKT2G-DN-008-000
AKT-DNH-004-000	4 Channel Digital Input Module, 24 VDC 0.2ms	Ø	AKT2G-DNH-008-000
AKT-DNH-008-000	8 Channel Digital Input Module, 24 VDC 0.2ms	Ø	AKT2G-DNH-008-000
AKT-DT-004-000	4 Channel Digital Output Module, 24 VDC 0.5A	Ø	AKT2G-DT-008-000
AKT-DT-008-000	8 Channel Digital Output Module, 24 VDC 0.5A	Ø	AKT2G-DT-008-000
AKT-DT-2RT-000	2 Channel Relay Output Module, 230 V AC 2.0A Rel.2NO PotFree	Ø	-
AKT-ECT-000-000	EtherCAT Bus Coupler	Ø	AKT2G-ECT-000-000
AKT-EM-000-000	Standard-Bus End Terminal	Ø	AKT2G-EM-000-000
AKT-IM-000-000	Isolation / Separation Terminal	Ø	—
AKT-PS-024-000	Power Supply, 24 VDC	Ø	AKT2G-PSF-024-000
AKT-PSF-024-000	Fused Power Supply with diagnostics, 24 VDC	Ø	AKT2G-PSF-024-000
AKT-SM-L15-000	Stepper Motor Terminal, 24 VDC, 1.5 A	Ø	-
AKT-SM-L50-000	Stepper Motor Terminal, 50 VDC, 5 A	Ø	_

IMPORTANT

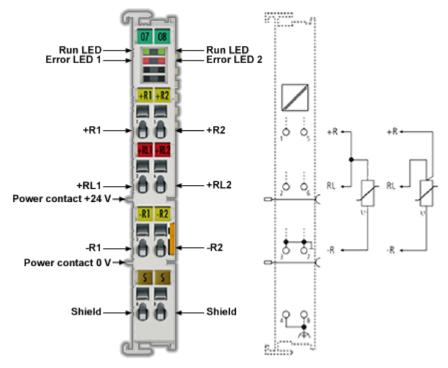
24-volt power is not passed through AKT-AN-200-000 and AKT-AN-400-000 thermocouple modules. To get 24VDC power to devices that need it (such as an AKT_AT-410-000 Output module) there are two possible solutions.

- Place the module requiring 24VDC before the thermocouple module.
- Add a power feed module (AKT-PS-024-000 or AKT-PSF-024-000) after the thermocouple module.

Table 9-4: List of KAS I/O Terminals

14.6.1 AKT2G-AN-240-000

2-channel input terminal PT100 (RTD) for resistance sensors, 16 bit, 2-, 3-wire system

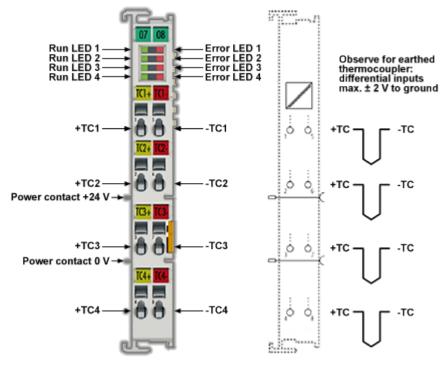


Technical Data	AKT2G-AN-240-000
Number of inputs	2
Sensor types	Pt100, Pt200, Pt500, Pt1000, Ni100, Ni120, Ni1000 KT/KTY from firmware 06 Resistance measurement 10 Ω to 1 k Ω or 10 Ω to 4 k Ω (e.g. for potentiometer connection)
Connection method	2-, 3-wire (Preset: 3-wire)
Temperature range	Range-dependent: -200+850°C (Pt sensors); -60+250°C (Ni sensors)
Resolution (default)	0.1°C per digit
Conversion time	approx. 800 ms - 2 ms (configurable), depending on configuration and filter setting approx. 85 ms, preset
Measuring current (depending on the sensor element and temperature)	typ. < 0.5 mA
Measuring error	for Pt sensors: < ± 0.5 °C at ambient temperature 0°C ± 55 °C < ± 1.5 °C in the extended temperature range
Width in the process image	max. 8-byte input
Power supply for electronics	via the E-Bus
Current consumption from the Ebus	typ. 190 mA
Electrical isolation	500 V (E-bus/field voltage)
Weight	approx. 60 g

Technical Data	AKT2G-AN-240-000
Permissible ambient temperature range during operation	-25°C to +60°C (extended temperature range)
Permissible ambient temperature range during storage	-40°C +85°C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (connected width: 12 mm)
Installation	on 35 mm mounting rail according to EN 60715
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also Installation instructions for enhanced mechanical load capacity.
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE ATEX cULus

14.6.2 AKT2G-AN-400-000

4-channel thermocouple input terminal, preset to type K, with wire breakage detection, 16 bit

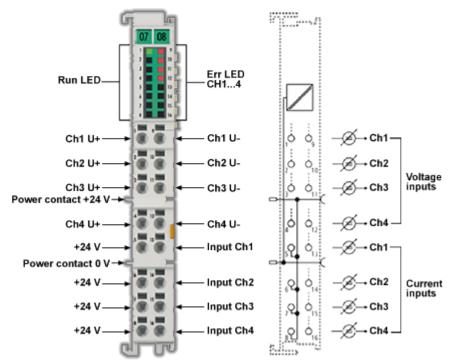


Technical Data	AKT2G-AN-400-000
Number of inputs	4
Thermocouple sensor types	Types J, K, L, B, E, N, R, S, T, U, C (default setting type K), mV measurement
Input filter limit frequency	1 kHz typ.; depending on sensor length, conversion time, sensor type
Connection technology	2-wire
Maximum cable length to the thermocouple	30 m
Measuring range, FSV	in the range defined in each case for the sensor (default setting: type K; -200 +1370°C)
	Voltage: ± 30 mV (1 μ V resolution) up to ± 75 mV (4 μ V resolution)
Resolution	Internal: 16 bit
	Temperature representation: 0.1/0.01 °C per digit, default 0.1°C
	Note: 16 bit is used for FSV calculation; so, value leaps >0.01°C will occur at resolution 0.01°C depending of which thermocouple is set; e.g. type K: approx. 0.04°C
Supports NoCoeStorage function	yes, from firmware 01
Wiring fail indication	yes
Conversion time	approx. 2.5 s to 20 ms,
	depending on configuration and fil ter setting, default: approx. 250 ms

Technical Data	AKT2G-AN-400-000		
Measuring error	< ±0.3 % (relative to full scale value)		
Voltage supply for electronics	via the E-bus		
Distributed Clocks	-		
Current consumption via E-bus	typ. 200 mA		
Bit width in the process data image	max. 16 byte input, max. 8 byte output		
Max. potential ±TC against ground	2 V, important e.g. when operating with grounded thermocouples		
Max. differential voltage between the ±TC inputs	±15 V permanent		
Electrical isolation	500 V (E-bus/field voltage)		
Configuration	via TwinCAT System Manager		
Weight	approx. 60 g		
Permissible ambient temperature range during operation	-25°C +60°C (extended temperature range), from firmware 06		
Permissible ambient temperature range during storage	-40°C +85°C		
Permissible relative humidity	95%, no condensation		
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)		
Mounting	on 35 mm mounting rail conforms to EN 60715		
Vibration/shock	conforms to EN 60068-2-6 / EN 60068-2-27,		
resistance	see also installation instructions for terminals with increased mechanical load capacity		
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4		
Protection class	IP20		
Installation position	variable		
Approval	CE, ATEX, cULus, IECEx		

14.6.3 AKT2G-AN-430-000

4-channel analog input, parameterisable, -10/0...+10 V, -20/0/+4...+20 mA, 16 bit



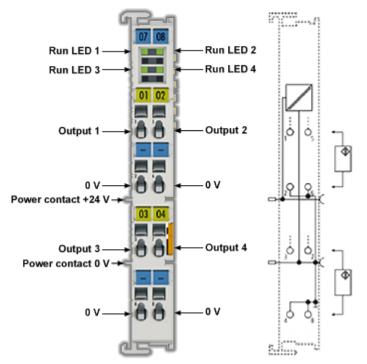
Technical Data	AKT2G-AN-430-000			
Analog inputs		4 (U differential, I single-ended)		
Conversion type		simultaneous		
ADC type		SAR		
Signal voltage		-10/0+10 V		
Signal current		-20/0/+4+20 mA		
Measuring range, nominal (Full Scale Value)	Voltage measurement range	-10/0+10 V		
	Current measurement range	-20/0/+4+20 mA		
Measuring range, technical Voltage measuring range		-10.73+10.73 V		
Current measur range		-21.47+21.47 mA		
Measuring error		< ±0.3 % (relative to full scale value)		
(full measuring range)				
Distributed Clocks		yes		

Technical Data		AKT2G-AN-430-000		
Distributed Clocks precision	<< 1 µs			
Support NoCoeStorage	yes			
Resolution		16 bit (incl. sign)		
Internal resistance		Voltage measurement: > 200 k Ω Current measurement: 85 Ω typ.		
Input filter limit frequency		5 kHz		
Common-mode voltage UCM		35 V max. (voltage measurement)		
Minimal EtherCAT cycle time		200 µs		
Overcurrent protection		50 mA typ.		
Bit width of the process image		Inputs: 16 Byte		
Configuration		no address or configuration settings required		
MTBF (+55°C)		-		
Special features		U/I parameterisable, Extended Range, standard and compact process image, activatable FIR/IIR filters		
Supply voltage for electronic		via the E-bus		
Current consumption via E-bus		typ. 170 mA		
Electrical isolation		500 V (E bus/ fieldbus voltage)		
Recommended operating voltage range (ground related to GND/ 0V power contact)	Voltage measurement range	UCM 35 V max.		
	Current	single ended,		
	measurement range	dielectric strength max. 30 V		
Recommended signal range	Voltage measurement range	Extended Range (107%), differential		
	Current measurement range	Extended Range (107%), single ended		
Destruction limit (ground related to GND/ 0V power contact)	Voltage measurement range	50 V		
,	Current measurement range	30 V		

Technical Data	AKT2G-AN-430-000			
Destruction limit (differential)	Voltage measurement range	50 V		
	Current measurement range	n.a.		
Weight		approx. 65 g		
Permissible ambient temperature range of operation	during	-25+60 °C		
Permissible ambient temperature range of	during storage	-40+85 °C		
Permissible relative humidity		95 %, no condensation		
Design		HD (High Density) housing with signal LED		
Dimensions (W x H x D)		approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)		
Mounting		on 35 mm mounting rail conforms to EN 60715		
Vibration/shock resistance		conforms to EN 60068-2-6/EN 60068-2-27		
EMC immunity/emission		conforms to EN 61000-6-2/EN 61000-6-4		
Protection class		IP20		
Installation position		variable		
Approval		CE, cULus		

14.6.4 AKT2G-AT-410-000

4-channel analog output terminal 0...10 V, 12 bit, 1-wire system

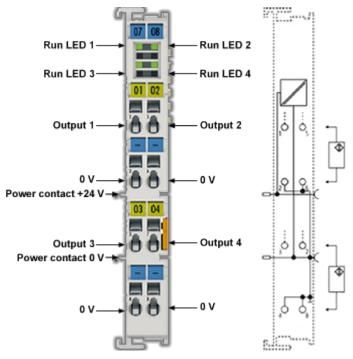


Technical Data	AKT2G-AT-410-000			
Number of outputs	4			
Power supply	24 V DC via the power contacts			
Signal voltage	010 V			
Load	> 5 kΩ (short-circuit-proof)			
Measuring error	< ± 0.1% (at 0 °C +55 °C, relative to the full scale value)			
	$< \pm 0.2\%$ (when the extended temperature range is used)			
Resolution	12 bit			
Conversion time	~ 250 µs			
Power supply for electronics	via the E-bus			
Distributed Clocks	yes			
Current consumption via Ebus	typ. 140 mA			
Electrical isolation	500 V (E-bus/field voltage)			
Bit width in process image	4 x 16-bit AO output			
Configuration	via TwinCAT System Manager			
Weight	approx. 60 g			
Permissible ambient temperature range during	-25 °C +60 °C			
operation	(extended temperature range)			

Technical Data	AKT2G-AT-410-000		
Permissible ambient temperature range during storage	-40 °C +85 °C		
Permissible relative humid- ity	95%, no condensation		
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)		
Mounting	on 35 mm mounting rail conforms to EN 60715		
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27,		
	see also installation instructions for enhanced mechanical load capacity		
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4		
Protection class	IP20		
Installation position	variable		
Approval	CE, ATEX, cULus		

14.6.5 AKT2G-AT-425-000

4-channel analog output terminal -10 V...+10 V, 12 bit, 4 x 2-wire system

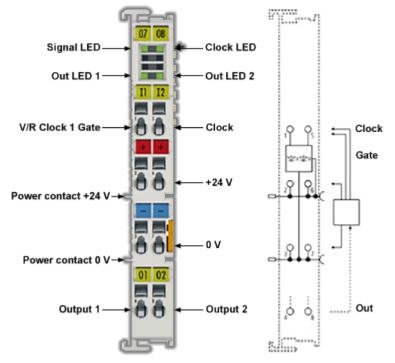


Technical Data	AKT2G-AT-425-000		
Number of outputs	4		
Power supply	24 V DC via the power contacts		
Signal voltage	-10+10 V		
Load	> 5 kΩ (short-circuit-proof)		
Measuring error	< ± 0.1% (at 0 °C +55 °C, relative to the full scale value)		
	$< \pm 0.2\%$ (when the extended temperature range is used)		
Resolution	12 bit		
Conversion time	~ 250 µs		
Power supply for electronics	via the E-bus		
Distributed Clocks	yes		
Current consumption via Ebus	typ. 140 mA		
Electrical isolation	500 V (E-bus/field voltage)		
Bit width in process image	4 x 16-bit AO output		
Configuration	via TwinCAT System Manager		
Weight	approx. 60 g		
Permissible ambient temperature range during	-25 °C +60 °C		
operation	(extended temperature range)		

776

Technical Data	AKT2G-AT-425-000
Permissible ambient temperature range during storage	-40 °C +85 °C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Mounting	on 35 mm mounting rail conforms to EN 60715
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27,
	see also installation instructions for enhanced mechanical load capacity
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE, ATEX, cULus

14.6.6 AKT2G-DN-002-000



14.6.6.1 Up/down counter 24 V_{DC}, 100 kHz, 32 bit counter depth

The up/down counter counts binary pulses, and transmits the counter state, in an electrically isolated form, to the higher-level automation device.

The AKT2G-DN-002 EtherCAT Terminal can alternatively be operated as:

- a single-channel counter (32 bit) that can be toggled between counting up and down via the U/D input (delivery state)
- a single-channel counter (32 bit) that is controlled via the gate connection
- two separate logic counters (32 bit) that can count in one direction only with the Clock 1 and Clock 2 inputs

The signal state of the inputs and outputs is indicated by light emitting diodes.

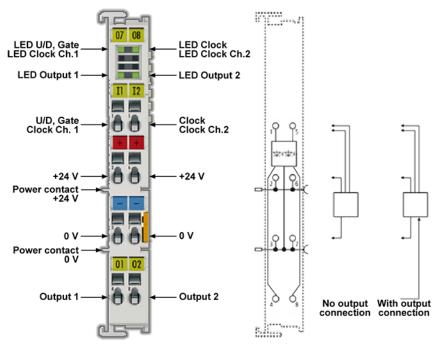
The two outputs (Output 1 and Output 2) are switched in relation to the counter value and can thus be used as fast control signals for field devices.

The AKT2G-DN-002 supports distributed clocks, i.e. the input data can be monitored synchronously with other data that are also linked to distributed clock terminals. The accuracy across the system is < 100 ns.

14.6.6.2 Technical Data

Technical Data	AKT2G-DN-002-000
Number of counters	1 or 2
Rated voltage	24 VDC (-15%/+20%)
Signal voltage "0"	-3 V 5 V (EN 61131-2, type 1)
Signal voltage "1"	15 V 30 V (EN 61131-2, type 1)
Counting frequency	100 kHz
Counter depth	32 bit
Input current	typ. 5 mA (EN 61131-2, type 1)
Output current (per channel)	max. 0.5 A (short-circuit-proof)
Distributed Clocks (DC)	yes
Current consumption power contacts	typ. 14 mA + load
Current consumption via E-bus	typ. 130 mA
Electrical isolation	500 V (E-bus/field voltage)
Supports NoCoeStorage function	yes
Weight	approx. 50 g
Permissible ambient temperature range during operation	-25°C +60°C (extended temperature range)
Permissible ambient temperature range during storage	-40°C +85°C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm
Mounting	on 35 mm mounting rail conforms to EN 60715
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also Installation instructions for enhanced mechanical load capacity
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE ATEX, cULus

14.6.6.3 LEDs and Connection



LEDs

LED	Color	Meaning
UP/DOWN, GATE CLOCK CH.1	green	Signal at U/D, Gate input (operating mode 32 bit up/down counter, 32 bit gated counter)
		Signal at the Clock input, channel 1 (operating mode 2 x 32 bit counter)
		see "Operating Mode Selection" on page 781
CLOCK CLOCK CH.2	green	Signal at Clock input (operating mode 32 bit up/down counter, 32 bit gated counter)
		Signal at the Clock input, channel 2 (operating mode 2 x 32 bit counter
		see "Operating Mode Selection" on page 781
OUTPUT1	green	Signal at the corresponding output
OUTPUT2		

Connection

780

Terminal Point		Description		
Name No.				
U/D, Gate	1	Up/down input (operating mode 32 bit up/down counter),		
Clock Ch.1		Gate input (operating mode 32 bit counter with gate function) Clock 1 input (operating mode 2 x 32 bit counter)		
		see "Operating Mode Selection" on page 781		
+24 V 2		+24 V (internally connected to terminal point 6 and positive power contact)		
0 V 3		0 V (internally connected to terminal point 7 and negative power contact)		

Terminal Point		Description		
Name	No.			
Output 1	4	Output 1		
Clock 5		Clock input (operating mode 32 bit up/down counter) and (operating mode 32		
Clock Ch.2		bit counter with gate function)		
		Clock 2 input (operating mode 2 x 32 bit counter)		
		see "Operating Mode Selection" on page 781		
+24 V	6	+24 V (internally connected to terminal point 2 and positive power contact)		
0 V 7 0 V (internally connected to terminal point 3 and negative power conta		0 V (internally connected to terminal point 3 and negative power contact)		
Output 2	8	Output 2		

14.6.6.4 Basic Function Principles

The AKT2G-DN-002 input terminals count binary pulses and transfer the current value to the higher-level controller.

In addition to the 32 bit up/down counter, further available operating modes are a 32 bit gated counter or two 32 bit counters. In gated counter mode, a low or high level at the Gate input inhibits the counting function of the terminal.

If two 32 bit counters are active, the U/D input (terminal point 1) is configured as the input for the first counter and the Clock 2 input (terminal point 5) as the input for the second counter.

Beyond that, two digital outputs can be set.

The maximum input frequency is limited to 100 kHz for the AKT2G-DN-002; the counters react to the rising edge of the input signal.

14.6.6.5 Operating Mode Selection

The following operation modes are possible.

Operation mode	Predefined PDO Assignment	Setting of the counting direction via CoE directory	Switchable outputs
1 (default)	1Ch. +/- Counter:	Index 0x8020:05:	Output 1
	0x1A02 – CNT Inputs	0: Enable UD counter	
+ 0x1602 – CNT Outputs		 UD input (terminal point 1): sets the counting direction: High level: up; Low level: Down 	
		 Clock input (terminal point 5): indicates the individual pulses. 	

Operation mode	Predefined PDO Assignment	Setting of the counting direction via CoE directory	Switchable outputs
2	1Ch. +/- Counter:	Index 0x8020:05:	Output 1
	0x1A02 – CNT Inputs	1: Enable pos. gate	
	+ 0x1602 – CNT Outputs	 Gate is inhibited by a positive level on the gate input (terminal point 1) Clock input (terminal point 5): 	
		indicates the individual pulses.	
		+	
		Index 0x8020:04:	
		0: up counter	
		1: down counter	
3	1Ch. +/- Counter:	Index 0x8020:05:	Output 1
	0x1A02 – CNT Inputs	2: Enable neg. gate	
	+ 0x1602 – CNT Outputs	 Gate is inhibited by a negative level on the gate input (terminal point 1) Clock input (terminal point 5): indicates the individual pulses. 	
		+	
		Index 0x8020:04:	
		0: up counter	
		1: down counter	
4	2Ch. Counter: 0x1A00 – CNT Inputs	Index 0x8000:04 (Channel1) and Index 0x8010:04 (Channel2)	Counter 1 → Output 1
	Channel 1	0:up counter	Counter 2 \rightarrow
	0x1A01 – CNT Inputs Channel 2	1:down counter	Output 2
	+		
	0x1600 – CNT Outputs Channel 1		
	0x1601 – CNT Outputs Channel 2		

In addition, the distributed clock function may be activated for the AKT2G-DN-002.

Single-channel up/down counter, gated counter (operating mode 1-3)

- 1. Selection of the PDOs for "1Ch.+/-Counter"
- 2. CoE Init-command to configure index 0x8020:05:
 - "Enable UD counter" single-channel up/down counter (operating mode 1)
 If a high signal level is encountered at the up/down input of the terminal (terminal point
 1), the counter counts up in the event of positive edges at the clock input (terminal point
 5), with a low signal level it counts down.
 - "Enable pos. gate" single-channel gated counter closes in the case of a high level (operating mode 2)

The counter is inhibited if a high level is encountered at the gate input of the terminal (terminal point 1).

The counting direction is set by index 0x8020:04 (0: up, 1: down). The clock input (terminal point 5) indicates the individual pulses.

"Enable neg. gate" – single-channel gated counter closes in the case of a low level (operating mode 3)

The counter is inhibited if a low level is encountered at the gate input of the terminal (terminal point 1). The counting direction is set by index 0x8020:04 (0: up, 1: down). The clock input (terminal point 5) indicates the individual pulses.

Two-channel up/down counter (operating mode 4)

- Selection of the PDOs for "2Ch. Counter" The terminal points 1 or 5 serve as clock input for 32 bit counter 1 or 2.
- 2. CoE Init-commands to configure the indices 0x8000:04 for channel 1 and 0x8010:04 for channel. Two options are available per channel:
 - 0: up counter
 - 1: down counter

14.6.6.6 Objects for Commissioning

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore default parameters	UINT8	RO	0x01 (1dec)
1011:01	SubIndex 001	If this object is set to "0x64616F6C" in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0dec)

Index 8000 CNT Settings Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:0	CNT Settings Ch.1	Maximum subindex	UINT8	RO	0x13 (19dec)
8000:01	Enable function to set output	Activates the function for setting Output 1	BOOLEAN	RW	0x00 (0dec)
8000:02	Enable function to reset output	Activates the function for resetting Output 1	BOOLEAN	RW	0x00 (0dec)
8000:03	Enable reload	The counter counts to the value in index 0x8000:13	BOOLEAN	RW	0x00 (0dec)
8000:04	Count down	Counting direction: • 0: Up • 1: Down	BOOLEAN	RW	0x00 (0dec)
8000:11	Switch on threshold value	Switch-on threshold value for Output 1	UINT32	RW	0x00000000 (0dec)
8000:12	Switch off threshold value	Switch-off threshold value for Output 1	UINT32	RW	0x00000000 (0dec)
8000:13	Counter reload value	The limit that can be activated via "Enable reload" (index 0x8000:03). The counter counts to this limit and, on exceeding it, begins again at zero.	UINT32	RW	0x00000001 (1dec)

Index 8010 CNT Settings Ch.2

784

Index (hex)	Name	Meaning	Data type	Flags	Default
8010:0	CNT Settings Ch.2	Maximum subindex	UINT8	RO	0x13 (19dec)
8010:01	Enable function to set output	Activates the function for setting Output 2	BOOLEAN	RW	0x00 (0dec)

Index (hex)	Name	Meaning	Data type	Flags	Default
8010:02	Enable function to reset output	Activates the function for resetting Output 2	BOOLEAN	RW	0x00 (0dec)
8010:03	Enable reload	The counter counts to the value in index 0x8010:13	BOOLEAN	RW	0x00 (0dec)
8010:04	Count down	Counting direction 0: Up 1: Down 	BOOLEAN	RW	0x00 (0dec)
8010:11	Switch on threshold value	Switch-on threshold value for Output 2	UINT32	RW	0x00000000 (0dec)
8010:12	Switch off threshold value	Switch-off threshold value for Output 2	UINT32	RW	0x00000000 (0dec)
8010:13	Counter reload value	The limit that can be activated via "Enable reload" (index 0x8010:03). The counter counts to this limit and, on exceeding it, begins again at zero.	UINT32	RW	0x00000001 (1dec)

Index 8020 CNT Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8020:0	CNT Settings	Maximum subindex	UINT8	RO	0x13 (19dec)
8020:01	Enable function to set output	Activates the function for setting Output 1	BOOLEAN	RW	0x00 (0dec)
8020:02	Enable function to reset output	Activates the function for resetting Output 1	BOOLEAN	RW	0x00 (0dec)
8020:03	Enable reload	The counter counts to the value in index 0x8020:13	BOOLEAN	RW	0x00 (0dec)
8020:04	Count down	Counting direction 0: Up 1: Down 	BOOLEAN	RW	0x00 (0dec)
8020:05	Operating mode	 Operating mode 0: Enable UD counter U/D input (terminal point 1) specifies the counting direction: High level: up, low level: down 1: Enable pos. gate (gate inhibits with positive level) 2: Enable neg. gate (gate inhibits with negative level) 	BIT2	RW	0x00 (0dec)

KAS IDE User Guide | 14 KAS Component Manuals

Index (hex)	Name	Meaning	Data type	Flags	Default
8020:11	Switch on threshold value	Switch-on threshold value for Output 1	UINT32	RW	0x00000000 (0dec)
8020:12	Switch off threshold value	Switch-off threshold value for Output 1	UINT32	RW	0x00000000 (0dec)
8020:13	Counter reload value	The limit that can be activated via "Enable reload" (index 0x8020:03). The counter counts to this limit and, on exceeding it, begins again at zero.	UINT32	RW	0x00000001 (1dec)

14.6.6.7 Profile-specific objects (0x6000-0xFFFF)

The profile-specific objects have the same meaning for all EtherCAT slaves that support the profile 5001.

Index 6000 CNT Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	CNT Inputs Ch.1	Maximum subindex	UINT8	RO	0x11 (17dec)
6000:01	Output functions enabled	This bit indicates that the internal functions for the output have been enabled	BOOLEAN	RO	0x00 (0dec)
6000:02	Status of output	Status of the output	BOOLEAN	RO	0x00 (0dec)
6000:03	Set counter done	The counter was set	BOOLEAN	RO	0x00 (0dec)
6000:04	Counter inhibited	The counter is stopped for as long as this bit is set	BOOLEAN	RO	0x00 (0dec)
6000:06	Status of input clock	State of the Clock input (high level applied)	BOOLEAN	RO	0x00 (0dec)
6000:0E	Sync Error	Synchronization error	BOOLEAN	RO	0x00 (0dec)
6000:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated	BOOLEAN	RO	0x00 (0dec)
6000:11	Counter value	Counter value	UINT32	RO	0x00000000 (0dec)

Index 6010 CNT Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:0	CNT Inputs Ch.2	Maximum subindex	UINT8	RO	0x11 (17dec)
6010:01	Output functions enabled	This bit indicates that the internal functions for the output have been enabled	BOOLEAN	RO	0x00 (0dec)
6010:02	Status of output	Status of the output	BOOLEAN	RO	0x00 (0dec)
6010:03	Set counter done	The counter was set	BOOLEAN	RO	0x00 (0dec)
6010:04	Counter inhibited	The counter is stopped for as long as this bit is set	BOOLEAN	RO	0x00 (0dec)
6010:06	Status of input clock	State of the Clock input (high level applied)	BOOLEAN	RO	0x00 (0dec)
6010:0E	Sync Error	Synchronization error	BOOLEAN	RO	0x00 (0dec)

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated	BOOLEAN	RO	0x00 (0dec)
6010:11	Counter value	Counter value	UINT32	RO	0x00000000 (0dec)

Index 6020 CNT Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6020:0	CNT Inputs	Maximum subindex	UINT8	RO	0x11 (17dec)
6020:01	Output functions enabled	This bit indicates that the internal functions for the output have been enabled	BOOLEAN	RO	0x00 (0dec)
6020:02	Status of output	Status of the output	BOOLEAN	RO	0x00 (0dec)
6020:03	Set counter done	The counter was set.	BOOLEAN	RO	0x00 (0dec)
6020:04	Counter inhibited	The counter is stopped for as long as this bit is set	BOOLEAN	RO	0x00 (0dec)
6020:05	Status of input UD	State of the Up/Down input (high level applied)	BOOLEAN	RO	0x00 (0dec)
6020:06	Status of input clock	State of the Clock input (high level applied)	BOOLEAN	RO	0x00 (0dec)
6020:0E	Sync Error	Synchronization error	BOOLEAN	RO	0x00 (0dec)
6020:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated	BOOLEAN	RO	0x00 (0dec)
6020:11	Counter value	Counter value	UINT32	RO	0x00000000 (0dec)

Index 7000 CNT Outputs Ch.1

788

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	CNT Outputs Ch.1	Maximum subindex	UINT8	RO	0x11 (17dec)
7000:01	Enable output functions	The internal functions for the output are enabled via this bit	BOOLEAN	RO	0x00 (0dec)
7000:02	Set output	Set output	BOOLEAN	RO	0x00 (0dec)
7000:03	Set counter	Set counter	BOOLEAN	RO	0x00 (0dec)

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:04	Inhibit counter	The counter is stopped as long as this bit is active. The previous counter state is retained.	BOOLEAN	RO	0x00 (0dec)
7000:11	Set counter value	This is the counter value to be set via "Set counter" (index 0x7000:03).	UINT32	RO	0x00000000 (0dec)

Index 7010 CNT Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
7010:0	CNT Outputs Ch.2	Maximum subindex	UINT8	RO	0x11 (17dec)
7010:01	Enable output functions	The internal functions for the output are enabled via this bit	BOOLEAN	RO	0x00 (0dec)
7010:02	Set output	Set output	BOOLEAN	RO	0x00 (0dec)
7010:03	Set counter	Set counter	BOOLEAN	RO	0x00 (0dec)
7010:04	Inhibit counter	The counter is stopped as long as this bit is active. The previous counter state is retained.	BOOLEAN	RO	0x00 (0dec)
7010:11	Set counter value	This is the counter value to be set via "Set counter" (index 0x7010:03).	UINT32	RO	0x00000000 (0dec)

Index 7020 CNT Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7020:0	CNT Outputs	Maximum subindex	UINT8	RO	0x11 (17dec)
7020:01	Enable output functions	The internal functions for the output are enabled via this bit	BOOLEAN	RO	0x00 (0dec)
7020:02	Set output	Set output	BOOLEAN	RO	0x00 (0dec)
7020:03	Set counter	Set counter	BOOLEAN	RO	0x00 (0dec)
7020:04	Inhibit counter	The counter is stopped as long as this bit is active. The previous counter state is retained.	BOOLEAN	RO	0x00 (0dec)
7020:11	Set counter value	This is the counter value to be set via "Set counter" (index 0x7020:03).	UINT32	RO	0x00000000 (0dec)

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2dec)

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:01	Module index distance	Index spacing of the objects of the individual channels	UINT16	RO	0x0010 (16dec)
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0003 (3dec)

Index F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default
F008:0*	Code	NoCoeStorage function:	UINT32	RW	0x0000000
	word	The input code of the code word 0x12345678 activates the NoCoeStorage function:			(0dec)
		Changes to the CoE directory are not saved if the function is active. The function is deactivated by: 1.) changing the code word or 2.) restarting the terminal.			

* Function NoCoeStorage from Firmware 03

NOTE

Code word

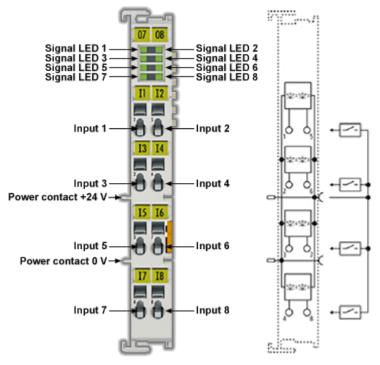
The vendor reserves the authority for the basic calibration of the terminals. The code word is therefore at present reserved.

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Maximum subindex	UINT8	RO	0x03 (3dec)
F010:01	SubIndex 001	reserved	UINT16	RO	0x0096
					(150dec)
F010:02	SubIndex 002	reserved	UINT16	RO	0x0096
					(150dec)
F010:03	SubIndex 003	reserved	UINT16	RO	0x0096
					(150dec)

14.6.7 AKT2G-DN-008-000

8-channel digital input terminal 24 V DC, filter 3.0 ms, 1-wire system

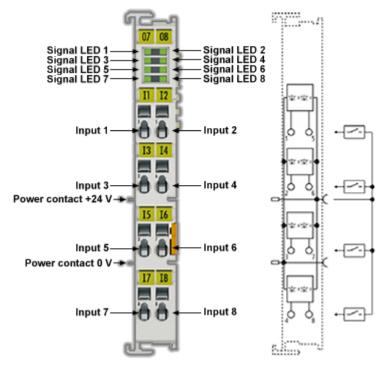


Technical Data	AKT2G-DN-008-000
Number of inputs	8
Number of simultaneously controllable in- puts, depending on the ambient temperature	8 (-25°C +55°C) 4 (> +55°C) (aligned in horizontal installation position)
Nominal voltage of the inputs	24 VDC (-15% / +20%)
Signal voltage "0"	-3 V 5 V (EN 61131-2, type 1/3)
Signal voltage "1"	11 V 30 V (EN 61131-2, type 1/3)
Input filter	3 ms
Input current	typically 3 mA (EN 61131-2, type 1/3)
Current consumption power contacts	typ. 2 mA + load
Current consumption via E-bus	typ. 90 mA
Electrical isolation	500 V (E-bus/field voltage)
Bit width in the process image	8 input bits
Configuration	no address setting, configuration via TwinCAT System Manager
Weight	approx. 55 g
Permissible ambient temperature range during operation	-25°C +60°C (extended temperature range, aligned in horizontal installation position) -25°C +45°C (all other installation positions)

Technical Data	AKT2G-DN-008-000
Permissible ambient temperature range during storage	-40°C +85°C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Mounting	on 35 mm mounting rail conforms to EN 60715
Vibration/shock resistance	according to EN 60068-2-6/EN 60068-2-27, see also Installation instructions for terminals with increased mechanical load capacity
EMC resistance burst/ESD	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	see note Constraints regarding installation position and operating temperature range
Approval	CE, cULus, ATEX, IECEx

14.6.8 AKT2G-DNH-008-000

8-channel digital input terminal 24 V DC, filter 10 µs, 1-wire system

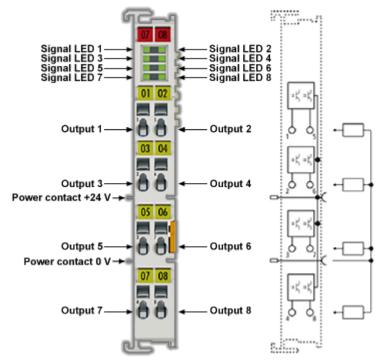


Technical Data	AKT2G-DNH-008-000	
Number of imputs	8	
Number of simultaneously controllable inputs,	8 (-25°C +55°C)	
depending on the ambient temperature	4 (> +55°C)	
	(aligned in horizontal installation position)	
Nominal voltage of the inputs	24 VDC (-15% / +20%)	
Signal voltage "0"	-3 V5 V (EN 61131-2, type 1/3)	
Signal voltage "1"	11 V 30 V (EN 61131-2, type 1/3)	
Input filter	10 µs typ. (1050 µs)	
Input current	typically 3 mA (EN 61131-2, type 1/3)	
Current consumption power contacts	typ. 2 mA + load	
Current consumption via E-bus	typ. 90 mA	
Electrical isolation	500 V (E-bus/field voltage)	
Bit width in the process image	8 input bits	
Configuration	no address setting, configuration via TwinCAT System Manager	
Weight	approx. 55 g	

Technical Data	AKT2G-DNH-008-000	
Permissible ambient temperature range during	-25°C +60°C	
operation	(extended temperature range, aligned in horizontal installa tion position)	
	-25°C +45°C	
	(all other installation positions)	
Permissible ambient temperature range during storage	-40°C +85°C	
Permissible relative humidity	95%, no condensation	
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)	
Mounting	on 35 mm mounting rail conforms to EN 60715	
Vibration/shock resistance	according to EN 60068-2-6/EN 60068-2-27,	
	see also Installation instructions for terminals with increased mechanical load capacity	
EMC resistance burst/ESD	conforms to EN 61000-6-2 / EN 61000-6-4	
Protection class	IP20	
Installation position	see note Constraints regarding installation position and operating temperature range	
Approval	CE, cULus, ATEX	

14.6.9 AKT2G-DT-008-000

8-channel digital output terminal 24 V DC, 0.5 A, 1-wire system

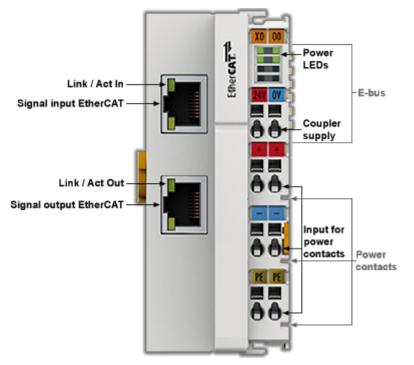


Technical Data	AKT2G-DT-008-000	
Number of outputs	8	
Non-reactive outputs	yes (see Interference-Free EtherCAT Terminals)	
Load type	ohmic, inductive, lamp load	
Nominal output voltage	24 VDC (-15% / +20%)	
Switching times	TON: 60 μs typ.; TOFF: 300 μs typ.	
Output current per channel	maximum 0.5 A (short-circuit proof)	
Switch-off energy (inductive)	max. 150 mJ/channel	
Current consumption from load voltage (power contacts)	typ. 15 mA	
Supply voltage for electronic	via the E-Bus	
Current consumption via E-bus	typ. 110 mA	
Electrical isolation	500 V (E-bus/field voltage)	
Bit width in the process image	8 output bits	
Configuration	no address setting, configuration via TwinCAT System Manager	
Weight	approx. 55 g	
Permissible ambient temperature range	Aligned in horizontal installation position:	
during operation	-25°C +60°C (extended temperature range)	
	All other installation positions: -25°C +45°C	

Technical Data	AKT2G-DT-008-000
Permissible ambient temperature range during storage	-40°C +85°C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Mounting	on 35 mm mounting rail conforms to EN 60715
Vibration/shock resistance	according to EN 60068-2-6/EN 60068-2-27,
	see also Installation instructions for terminals with increased mechanical load capacity
EMC resistance burst/ESD	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	see note Constraints regarding installation position and operating temperature range
Approval	CE, cULus, ATEX

14.6.10 AKT2G-ECT-000-000

EtherCAT Coupler for E-bus terminals



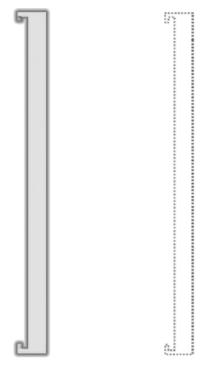
Characteristic	AKT2G-ECT-000-000	
Protection class	IP20	
Higher level network technology	100 MBit FastEthernet (100BASE-TX)	
Higher level network - max. connection length	100 m	
Higher level network connection technology	RJ45	
higher-level network protocol	EtherCAT Device Protocol	
supports HotConnect with address setting on the device	yes, Fast-Hot-Connect	

Technical Data	AKT2G-ECT-000-000	
Task in the EtherCAT system	Coupling of EtherCAT Terminals (ECT-xxx) to 100BASE-TX EtherCAT networks	
Number of EtherCAT Terminals	up to 65535 in the overall system	
Number of peripheral signals	max. 4.2 GB addressable IO points	
Data transfer medium	Ethernet 100BASE-TX (at least Ethernet CAT5 cable)	
Cable length between 2 Bus Couplers	max. 100 m (100BASE-TX)	
Protocol / Baud rate	EtherCAT Device Protocol / 100 MBaud	
HotConnect	no	
Delay	1 μs typ.	

Technical Data	AKT2G-ECT-000-000	
Bus connection	2 x RJ45	
Power supply	24 VDC (-15%/+20%)	
Current consumption	70 mA + (∑ E-bus current/4)	
E-bus power supply (5 V)	max. 2 A (-25 °C +55 °C)	
	max. 1 A (> +55 °C)	
Power contacts	max. 24 VDC, max. 10 A	
Electrical isolation	500 V (power contact/supply voltage/EtherCAT)	
Dimensions (W x H x D)	approx. 44 mm x 100 mm x 68 mm	
Weight	approx. 105 g	
Permissible ambient temperature range during operation	-25°C +60°C (extended temperature range)	
Permissible ambient temperature range during storage	-40°C + 85°C	
Permissible relative air humidity	95%, no condensation	
Mounting	on 35 mm mounting rail according to EN 60715	
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also Installation instructions] for enhanced mechanical load capacity	
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4	
Protection class	IP20	
Installation position	variable	
Approval	CE, ATEX, cULus, IECEx	

14.6.11 AKT2G-EM-000-000

Bus end cover for E-bus terminals, cover for power and E-bus contacts, grey

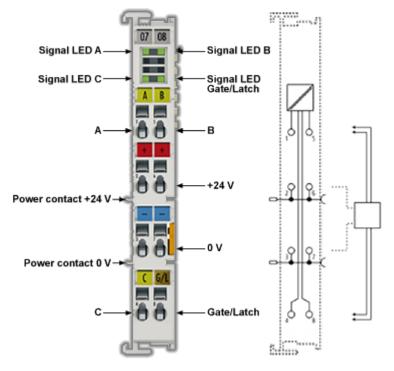


Each EtherCAT terminal block must be terminated at the right hand end with a AKT2G-EM-000-000 bus end cap due to mechanical and electrical protection.

Technical Data	AKT2G-EM-000-000	
Electrical isolation	-	
Bit width in the process image	0	
Configuration	no address or configuration settings	
Diagnosis	-	
PE contact	no	
Renewed infeed	-	
Connection facility to additional power contact	-	
Side by side mounting on Bus Terminals with power contact	yes	
Side by side mounting on Bus Terminals without power contact	yes	
Electrical connection to mounting rail	no	
Weight	approx. 8 g	
Permissible ambient temperature	-25°C +60°C (extended temperature range)	
Permissible ambient temperature range (during storage)	-40°C +85°C	
Permissible relative humidity	95%, no condensation	

Technical Data	AKT2G-EM-000-000	
Dimensions (W x H x D)	approx. 8 mm x 100 mm x 34 mm (width aligned: 5 mm)	
Mounting	aligned to the last terminal in the terminal block	
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27	
EMC resistance burst/ESD	conforms to EN 61000-6-2/EN 61000-6-4	
Protection class	IP 20	
Installation position variable		
Approval CE, ATEX, cULus		

14.6.12 AKT2G-ENC-180-000



14.6.12.1 Incremental Encoder Interface AKT2G-ENC-180

The AKT2G-ENC-180 EtherCAT Terminal is an interface with 24 V inputs for the direct connection of incremental encoders. A 32 bit counter with a quadrature decoder and a 32 bit latch for the zero pulse can be read, set or enabled.

The measurement of period and frequency is possible. The gate input allows the locking of the counter, selectively with a high or low level. The latch input is similarly configurable and evaluates high or low levels.

The AKT2G-ENC-180 supports distributed clocks, i.e. the input data can be synchronously acquired with other data that are similarly connected, distributed to distributed clock terminals. The universal system accuracy is around < 100 ns.

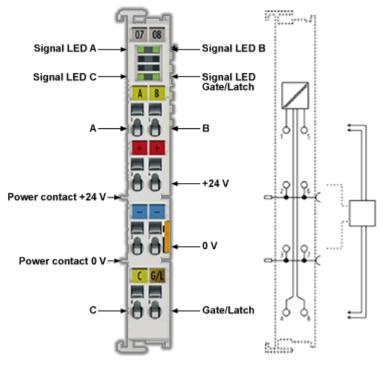
With a moving axis, the micro-increment functionality offers 256 times higher axis position resolution than physically provided by the encoder.

The AKT2G-ENC-180 can also be used as a single-channel 32/16 bit counter on channel A, in which case the signal level on channel B defines the count direction.

Technical Data	AKT2G-ENC-180-000	
Sensor inputs		
Encoder connection	A, B, C, gate/latch input, 24 V	
Encoder operating voltage	24 V	
Signal voltage "0" (inputs A, B, C, gate/latch)	0 V 5 V (EN 61131-2, type 1)	
Signal voltage "1" (inputs A, B, C, gate/latch)	15 V 30 V (EN 61131-2, type 1)	
Counter	1 x 32/16-bit binary, switchable	
Limit frequency	max. 400,000 increments/s with 4-fold evaluation), corresponds to 100 kHz	
Quadrature decoder	4-fold evaluation	
Timestamp resolution	1 ns	
Timestamp accuracy	100 ns	
Commands	Read, set, latch, gate function	
Power supply for electronic	via the E-Bus	
Distributed Clocks	yes	
Supply voltage	24 V _{DC} (-15 %/+20 %)	
Current consumption from the E-bus	typ. 130 mA	
Current consumption from the power contacts	0.1 A (excluding sensor load current)	
Electrical isolation	500 V (E-bus/field voltage)	
Supports NoCoeStorage function	yes	
Weight	approx. 50 g	
Permissible ambient temperature range during operation	-25 °C +60 °C (extended temperature range)	
Permissible ambient temperature range during storage	-40 °C +85 °C	
Permissible relative humidity	95%, no condensation	
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)	
Mounting	on 35 mm mounting rail conforms to EN 60715	
Vibration/shock resistance	according to EN 60068-2-6/EN 60068-2-27, see also Installation instructions for terminals with increased mechanical load capacity	
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4	
Protection class	IP20	
Installation position	variable	
Approval	CE, ATEX, cULus	

14.6.12.3 AKT2G-ENC-180-000 LEDs and Pin Assignment

802



AKT2G-ENC-180-000 LEDs

LED	Color	Meaning
А, В, С	green	flashes when pulses are present at the inputs
Gate, Latch	green	lights up when a signal is present at the gate/latch input
AKT2G-ENC-180-	000 Pir	n Assignment
Terminal Point	No.	Comment
А	1	Encoder input A
+24 V	2	+24 V (internally connected to terminal point 6 and positive power contact)
0 V	3	0 V (internally connected to terminal point 7 and negative power contact)
С	4	Encoder input C
С	5	Encoder input B
+24 V	6	+24 V (internally connected to terminal point 2 and positive power contact)
0 V	7	0 V (internally connected to terminal point 3 and negative power contact)
Gate / Latch 24 \	/ 8	Gate / Latch input

14.6.12.4 Commissioning AKT2G-ENC-180-000

14.6.12.4.1 Basic function principles

The terminal acquires the 90° phase-shifted digital output signal of an incremental encoder on channels A and B. The zero pulse is acquired on channel C. These signals are converted into a position value with quadruple evaluation with the aid of the quadrature encoder and the 32-bit counter. The latch and reset functions enable the exact referencing and saving of the counter value, irrespective of the speed.

Incremental encoders divide a 360° rotation of the encoder axis into individual steps (increments) and mark a full revolution by means of a special mark (zero pulse).

The phase angle between the signals on channels A and B sets the counting direction.

Up: signal on channel A leads signal on channel B by 90°

Down: signal on channel A lags signal on channel B by 90°

In case of single evaluation, the positive edges on channel A are counted.

In case of quadruple evaluation, the positive and negative edges on channel A and channel B are counted.

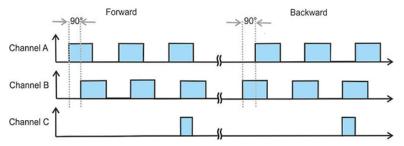


Figure 9-14: Quadrature decoder

Whereas absolute value encoders deliver an absolute and unambiguous position value over the complete travel path directly after switching on, it is necessary with incremental encoders to perform a reference run homing after switching on in order to be able to determine an unambiguous position.

Referencing can be carried out, for example, with the aid of reference cams or using the zero pulse of the encoder.

14.6.12.4.2 Operating modes and settings

14.6.12.4.2.1 General

The AKT2G-ENC-180 incremental encoder interface terminal enables connection of incremental encoders with the Bus Coupler or the PLC. A 32-bit counter with quadrature decoder as well as a 32-bit latch can be read, set or activated (switchable to 16-bit). In addition to the encoder inputs A, B and C, a gate/latch input (24 V) is also available on the AKT2G-ENC-180 for latching/saving the counter value. The gate/latch input is parameterizable via the CoE directory. no function, or disabling the counter at HIGH or LOW signal level.

The AKT2G-ENC-180 can also be used as a single-channel 32/16 bit counter on channel A, in which case the signal level on channel B defines the count direction. The changeover to this mode takes place via the CoE directory.

The Counter Value input value represents a 32-bit "position counter". At the period input the period between two positive edges of channel A is measured with a resolution of 100 ns (default setting, decimal value x 100 ns). Depending on the setting (index 0x8000:14, index 0x8000:16), the period length may be up to 1.6 s or 3.2 s.

14.6.12.4.2.2 Operating modes

Permissible operating modes for the AKT2G-ENC-180

The following modes are available for the AKT2G-ENC-180. They apply both for the encoder analysis and counter terminal mode.

The combinations of DC, PDO and CoE settings listed below are permissible per mode. Other settings can lead to irregular process data and error messages in the Safety PLC System Manager Logger window.

Mode	DC	Pain PDO	Optional PDO 1	Optional PDO 2	Features CoE
1	FreeRun	Predefined PDO Ass 0x1A00 + 0x1600 + 0	ignment: Standard 32- 0x1A02	-bit (MDP 511):	CoE combinations 0x8000:nn
		0x1A00 Inputs: 16 Bit Status, 32 Bit Counter Value, 32 Bit Latch Value +	0x1A02 32 Bit Period or 0x1A03 32 Bit Frequency		
		0x1600 Outputs: 16 Bit Control, 32 Bit Set Counter Value			
2	FreeRun	Predefined PDO Ass 0x1A01 + 0x1601 + 0	ignment: Standard 16 0x1A02	-bit (MDP 511):	CoE combinations 0x8000:nn
		0x1A01 Inputs: 16 Bit Status, 16 Bit Counter Value, 16 Bit Latch Value + 0x1601 Outputs: 16 Bit Control 16 Bit Set Counter Value	0x1A02 32 Bit Period or 0x1A03 32 Bit Frequency		
3	DC/DCi	Predefined PDO Ass Standard 32 Bit with 0x1A00 + 0x1600 + 0	64 Bit Timestamp (MD)P 511):	CoE combinations 0x8000:nn
		0x1A00 Inputs: 16 Bit Status, 32 Bit Counter Value, 32 Bit Latch Value + 0x1600 Outputs: 16 Bit Control, 32 Bit Set Counter Value	0x1A02 32 Bit Period or 0x1A03 32 Bit Frequency	0x1A04 64 Bit Timestamp or 0x1A05 32 Bit Timestamp (compact)	

Mode	DC	Pain PDO	Optional PDO 1	Optional PDO 2	Features CoE
4	DC/DCi	Predefined PDO Ass Standard 16 Bit with 0x1A01 + 0x1601 + 0	32 Bit Timestamp (MD	CoE combinations 0x8000:nn	
		0x1A01 Inputs: 16 Bit Status, 16 Bit Counter Value, 16 Bit Latch Value + 0x1601 Outputs: 16 Bit Control, 16 Bit Set Counter Value	0x1A02 32 Bit Period or 0x1A03 32 Bit Frequency	0x1A04 64 Bit Timestamp or 0x1A05 32 Bit Timestamp (compact)	

14.6.12.4.2.3 Explanatory notes for parameters and modes

14.6.12.5 Frequency

- The timeframe for the frequency calculation as well as the resolution can be parameterized in the CoE objects Frequency window (index: 0x80n0:11), Frequency scaling (index: 0x80n0:13), Frequency resolution (index: 0x80n0:15) and Frequency wait time (index: 0x80n0:17). The positive edges of track A are counted in the specified timeframe (see Frequency modes) and the next subsequent edge including the time until it arrives is counted. The waiting time for the edge can be set in the CoE object Frequency Wait Time (index: 0x80n0:17) (unit: ms) and is set as standard to 1.6 seconds. This is also the maximum value.
- The frequency is always specified as a positive number, irrespective of the sense of rotation.
- The size of the timeframe is 10 ms (default), but at the least the basic unit Frequency window base (index: 0x80n0:0F).
- This calculation is carried out in the terminal in free-running mode without reference to the distributed clocks system. It is therefore independent of the DC mode.
- AKT2G-ENC-180: No frequency measurement is possible if the counter is blocked by the gate. In this case the period can be measured regardless.
- AKT2G-ENC-180: A C or external reset restarts the frequency measurement. The last frequency value remains unchanged until a new frequency value is determined.
- The object Frequency window base (index: 0x80n0:0F) is used for switching the basic unit for the Frequency window between 1 µs and 1 ms, in order to adjust the time window for the measurement. The following maximum measuring windows are therefore possible:

Basic unit	Max. timeframe
1 µs	65.5 ms
1 ms	65 s

- on expiry of the measuring window Frequency window (index: 0x80n0:11), the subsequent positive edge on track A is awaited, but at the longest for 1.6 s or the time from Frequency wait time (index: 0x80n0:17).
- The frequency is measured with different accuracies depending on the selected basic unit Frequency window base (index 0x80n0:0F) and the window size.

14.6.12.6 Frequency mode A

The measurement is automatically performed in frequency mode A if the window size is smaller than or equal to 600 ms.

- Basic unit 1 µs: all window sizes
- Basic unit 1 ms: up to 600 ms window size

14.6.12.7 Measurement sequence

- The measurement starts with a positive edge at track A. The current counter value and time (resolution: 100 ns) are stored.
- On expiry of the measuring window Frequency window (index: 0x80n0:11), the subsequent rising edge on track A is awaited, but at the longest for 1.6 s or the time from Frequency wait time (index: 0x80n0:17).
- The frequency is calculated from the edge difference and the actual elapsed time.

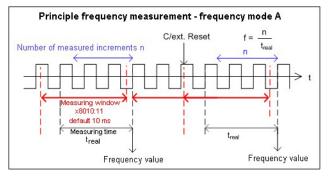


Figure 9-15: Frequency measurement principle - frequency mode A

14.6.12.8 Frequency mode B

The measurement is automatically performed in frequency mode B if the window size is greater than 600 ms.

• Basic unit 1 ms: from 601 ms window size

Measurement sequence

- At the start of the measurement the time and the current position are stored with a resolution of 100 ns, irrespective of the current signal position.
- After the measurement the current position is stored irrespective of the current signal position.
- The frequency is calculated from the number of increments and the actual elapsed time.
- The frequency measurement therefore takes place with reduced accuracy.
- The larger the measuring window in relation to the basic unit, the more precise the frequency calculation.

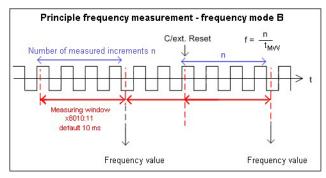


Figure 9-16: Frequency measurement principle - frequency mode B

14.6.12.9 Period calculation

- This calculation is carried out free-running in the slave without reference to the distributed clocks system. It is therefore independent of the DC mode.
- During each cycle the interval between two positive edges of input A is counted.
- Depending on the setting, periods of up to 1.6 s or 3.2 s in length are measurable.
- If no edge change occurs for approx. 1.6 s, any period specification is cancelled.

14.6.12.10 Gate function

The counter can be latched with the aid of the gate function. The Gate polarity object (index: 0x8000:04) provides three different options:

0: The gate function is inactive.

1: The counter value is latched by a HIGH level at the gate input. The counter value does not change as long as the HIGH level is applied. Signals on channels A and B have no effect on the counter value.

2: The counter value is locked by a LOW level at the gate input. The counter value does not change as long as the LOW level is applied. Signals on channels A and B have no effect on the counter value.

In the case of a simultaneous activation of the gate function (latch on HIGH level / latch on LOW level) and Enable extern reset (reset on positive edge / reset on negative edge), the counter value is first reset to zero. Latching subsequently takes place.

14.6.12.11 Latch

Activating the latch C input ("C") and latching the counter value

- The counter value is saved in Latch value (index: 0x6000:12) upon the first latch pulse (positive edge on input "C") after setting the bit (TRUE) in Enable latch C (index: 0x7000:01) (takes priority over Enable latch extern on positive / negative edge index: 0x7000:02 / 0x7000:04). If the bit is set, the subsequent pulses on the other inputs have no effect on the latch value in Latch value (index: 0x6000:12).
- After re-activation of Enable latch C (index: 0x7000:01), the next counter value at the latch input can be written only if the value of the Latch C valid bit (index 0x6000:01 FALSE) has been reset.

Activation of the external latch input and saving ("latching") of the counter value (index 0x7000:02, 0x7000:04)

- The counter value at the latch input Latch value (index 0x6000:12) will be saved upon the first external latch pulse with a positive edge if the bit (TRUE) is set in Enable extern latch on positive edge (index: 0x7000: 02). The subsequent pulses have no influence on the latch value in Latch value (index: 0x6000:12).
- The counter value at the latch input Latch value (index: 0x6000:12) will be saved upon the first external latch pulse with a negative edge if the bit (TRUE) is set in Enable extern latch on negative edge (index: 0x7000: 04). The subsequent pulses have no influence on the latch value in Latch value (index: 0x6000:12).
- After re-activation, a new counter value on the latch input can be written only if the value of the Latch extern valid bit (index: 0x6000:02) has been reset.

14.6.12.12 Reset

The counter can be reset via Enable C reset (index: 0x8000:01) or via Enable extern reset (index: 0x8000:02). Extern reset polarity (index: 0x8000:10) defines whether the reset takes place on a positive or negative edge at the external latch input.

"Enable C reset" (index: 0x8000:01)

• For activation the bit in Enable C reset (index: 0x8000:01) is set to TRUE.

The counter value is reset to zero if the zero pulse is present on channel C.

"Enable extern reset" (index: 0x8000:02),

- For activation the bit in Enable extern reset (index: 0x8000:02) is set to TRUE.
- "Extern reset polarity" (index: 0x8000:10)
 - Bit not set: the counter is set to zero with a negative edge at the external latch input.
 - Bit set: the counter is set to zero with a positive edge at the external latch input.

The simultaneous activation of the functions Enable C reset (index: 0x8000:01) und Enable extern reset (index: 0x8000:02) is not possible.

14.6.12.13 Up/down counter

• The operating mode (encoder or up/down counter) is selected via the CoE object Enable up/down counter (index: 0x80n0:03).

On the CoE - Online tab, click on the row of the index to be parameterized, enter the corresponding value in the SetValue dialog and confirm with OK.

- 0: the up/down counter is not active.
- 1: the up/down counter is active. In the case of the AKT2G-ENC-180 the counter value can be locked via the object Gate polarity (index:

0x8000:04) (s. Gate function).

- The counting direction (up/down) is specified via the signal level at channel B. An additional option for reversing the direction of rotation is to set the Reversion of rotation bit (index: 0x80n0:0E).
- Connection:

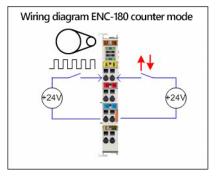


Figure 9-17: Counter connection principle

14.6.12.14 Micro-increments

- Works with and without distributed clocks, but in the AKT2G-ENC-180 this is only meaningful in conjunction with one of the DC modes
- By setting the counter value only the integer component can be modified.
- The principle:

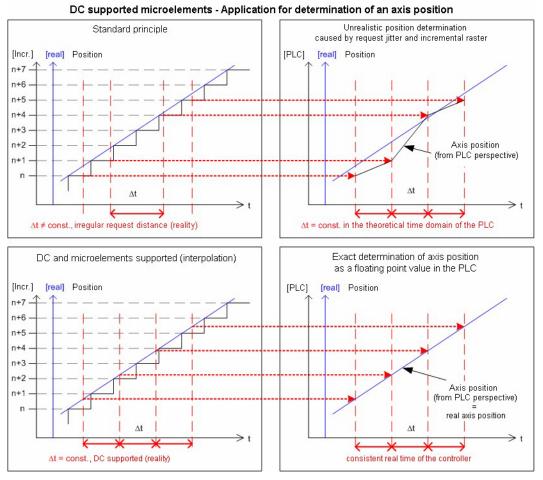


Figure 9-18: Principle of frequency measurement

The highly constant query cycles (accuracy: 100 ns) of the distributed clocks system permits the AKT2G-ENC-180 to interpolate interpolated axis positions between the counted encoder increments above a certain speed. The interpolation resolution is 8 bit, corresponding to 256 values. A standard encoder with 1,024 bars with 4-way evaluation and micro-increments thus becomes a high-resolution axis encoder with 4096 * 256 = 1,048,567 bars.

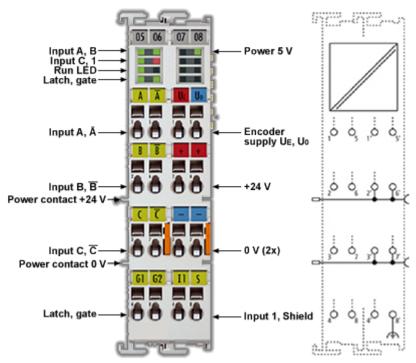
If the speed falls below the minimum speed, this is displayed by the object Extrapolation stall (index: 0x60n0:08) in the process data.

14.6.12.15 Digital filter

The AKT2G-ENC-180 has a digital filter on encoder channels A and B that can be switched off (object Disable Filter, index: 0x80n0:08). This acts as a diffuse low-pass filter at about 100 KHz (equals 400,000 increments/second with 4-fold evaluation), i.e. the permissible limit frequency.

In each application it is advisable to check whether it would be advantageous to deactivate the filter - the detection of fast axis movements may be improved as a result.

14.6.13 AKT2G-ENC-190-000



14.6.13.1 Interface terminal for incremental encoder

The AKT2G-ENC-190-000 EtherCAT Terminal is an interface for direct connection of incremental encoders with differential inputs (RS422). A 16-bit counter (in normal operating mode) or a switchable 16/32-bit counter (in enhanced operating mode) with a quadrature decoder and a 16-bit latch (in normal operating mode) or 32-bit latch (in enhanced operating mode) for the zero pulse can be read, set or enabled. Incremental encoders with alarm output can be connected at the negative switching status input of the interface. The measurement of period and frequency is possible. The gate input allows the locking of the counter, alternatively with a high or low level. The latch input is similarly configurable and evaluates high or low levels. The AKT2G-ENC-190-000 can also be used as bidirectional counter on channel A; channel B specifies the count direction.

14.6.13.2 Technology

The AKT2G-ENC-190-000 incremental encoder interface terminal enables connection of incremental encoders with

A/B/C track to the Bus Coupler and the PLC. A 16-bit counter (in normal operating mode) or a switchable 16/32-bit counter (in enhanced operating mode) with a quadrature decoder and a 16-bit latch (in normal operating mode) or 32-bit latch (in enhanced operating mode) can be read, set or enabled. Differential signals based on RS422 are provided as encoder connection. From hardware 09 [" 204] single-ended 5 V signals are possible for the AKT2G-ENC-190-000 based on pull-up resistors.

In addition to the encoder inputs A, B and C, an additional latch input G1 (24 V) and a gate input G2 (24 V) for locking the counter during operation are available.

The terminal is supplied as a 4-fold quadrature decoder with complementary analysis of the sensor signals A, B, C. If the incremental encoder has an alarm output it can be connected to the INPUT 1 status input of the AKT2G-ENC-190-000. The AKT2G-ENC-190-000 can optionally be operated as a bidirectional counter terminal on channel A.

14.6.13.2.1 AKT2G-ENC-190-000 input impedance

Figure 9-19: AKT2G-ENC-190-000

The signal source must be able to operate the input impedance of the AKT2G-ENC-190-000 (typically 220 Ω , subject to modification) with adequate voltage levels according to RS422.

14.6.13.2.2 Gate/latch input

For gate and latch inputs (24 V) a max. input frequency of 1 MHz is permitted. Subject to modification.

14.6.13.2.3 Level on interface

In differential mode the AKT2G-ENC-190-000 expects the signal levels after RS422. The data are transferred without ground reference as voltage difference between two cables (signal A and inverted signal /A). The terminal analyses signal levels in the range -200 mV < Vid < +200 mV as valid signals. The differential signal must be in the common mode range (<+13.2 V and >-10 V, with respect to GND) (cf. diagram). Signal levels outside this range can lead to destruction.

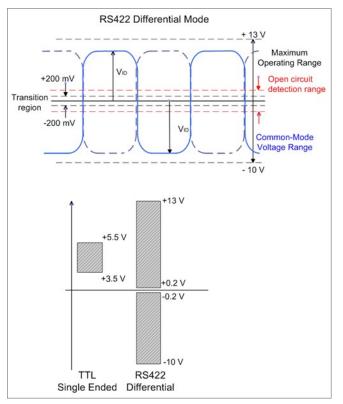


Figure 9-20: Level interface

In differential mode only the voltage difference is evaluated, so that common-mode interference on the transmission link does not lead to corruption of the wanted signal, since any interference affects both cables simultaneously.

If the AKT2G-ENC-190-000 is only operated in single-ended mode, a nominal level voltage between 3.5 V and 5.5 V is expected.

14.6.13.3 Technical Data

Technical Data	AKT2G-ENC-190-000
Sensor connection	A, \neg A, B, \neg B, C, \neg C (RS422 differential inputs) also single-ended connection (5 V ±20%) possible
Additional inputs	gate, latch (24 VDC, both max. 1 MHz permitted),
	status input (max. 5 VDC, potential-free, switching to negative potential)
Encoder operating voltage / Encoder supply	5 VDC (generated from the 24 V DC power contacts)
Sensor output current	0.5 A
Counter	16 bit, 16/32 bit switchable
Zero pulse latch	16 bit, 16/32 bit switchable
Limit frequency	1 MHz (equals 4 million increments with 4-fold evaluation)
Quadrature decoder	4-fold evaluation
Distributed Clocks	in enhanced operating mode
Broken wire detection to sensor	in enhanced operating mode
Commands	read, set, enable
Cycle time	min. 100 μs
Current consumption via Ebus	typ. 130 mA
Current consumption from the power contacts	0.1 A (excluding sensor load current)
Electrical isolation	500 V (E-bus/field voltage)
Bit width in process image	up to 6 bytes outputs, 22 bytes inputs, depends on parametrization
MTBF (+55°C)	-
Weight	approx. 100 g
Permissible ambient temperature range during operation	-25°C +60°C (extended temperature range)
Permissible ambient temperature range during storage	-40°C+85°C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 27 mm x 10 mm x 70 mm (width aligned: 24 mm)
Mounting	on 35 mm mounting rail conforms to EN 60715
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions for enhanced mechanical load capacity
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE ATEX cULus

14.6.13.4 AKT2G-ENC-190-000 LEDs and Connection

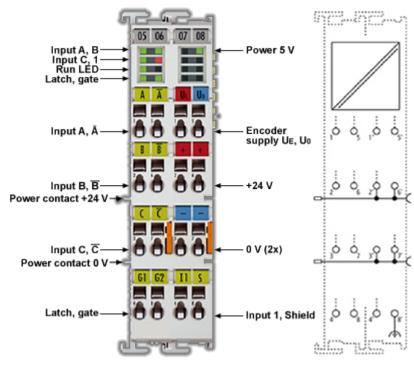


Figure 9-21: AKT2G-ENC-190-000

14.6.13.4.1 Connection

NOTE

Encoder supply via the terminal

The encoder supply voltage (5 V), can be taken from the terminal point 1'.

Terminal Point	No.	Comment
А	1	Encoder input A
В	2	Encoder input B
С	3	Encoder input C
Latch 24 V	4	Latch input
¬A	5	Encoder input A
¬B	6	Encoder input B
¬С	7	Encoder input C
Gate 24 V	8	Gate input
Ue = +5 V	1'	+5 V encoder supply
+24 V	2'	+24 V (internally connected to terminal point 6' and positive power contact)
0 V	3'	0 V (internally connected to terminal point 7' and negative power contact)
Input 1	4'	Status input 1 Alarm input from rotary encoder. Internally connected to 5 V via pull-up. Switching to negative potential, i.e. connection to GND leads to error bit and LED display. If externally supplied (not recommended) 5 V max. against GND is permitted.
Uo = 0 V	5'	0 V encoder supply
+24 V	6'	+24 V (internally connected to terminal point 2' and positive power contact)
0 V	7'	0 V (internally connected to terminal point 3' and negative power contact)
Shield	8'	Screen

14.6.13.4.2 LEDs

LED	Color	Meaning				
LED	Color	Meaning				
INPUT A, B, C	green	indicates TR	dicates TRUE level			
INPUT 1	red	is lit, if INPL	IT 1 is connected to GND			
		[INPUT 1 is (default)]	NPUT 1 is connected to an internal 5 V HIGH level though internal pull-up default)]			
LATCH	green	is lit, if a sigr	nal (+24 V) is connected to the latch input			
GATE	green	is lit, if a sigr	nal (+24 V) is connected to the gate input			
RUN	green	This LED indicates the terminal's operating state:				
		off	State of the EtherCAT State Machine: INIT = initialization of the terminal or BOOTSTRAP = function for firmware updates of the terminal			
		flashing	State of the EtherCAT State Machine: PREOP = function for mailbox communication and different standard-settings set			
		single flash	State of the EtherCAT State Machine: SAFEOP = verification of the Sync Manager channels and the distributed clocks.			
			Outputs remain in safe state			
		on	State of the EtherCAT State Machine: OP = normal operating state; mailbox and process data communication is possible			
POWER 5 V	green	Operating v	oltage display for incremental encoder power supply			

14.6.13.5 Commissioning AKT2G-ENC-190-000

14.6.13.5.1 Normal Operation Mode

14.6.13.5.1.1 Process data and modes - Normal Operation Mode

In AKT2G-ENC-190-000 "normal operation mode" the following modes are available:

Mode	Distributed Clock	Main PDO	Comment	Optional PDO 1	Comment	Features CoE	Comment
1	Operational	0x1A00 + 0x1600	16 bit Value/Latch Byte- Alignment	0x1A02	Frequency: 32 bit Period: 16 bit Window: 16 bit CoE object 0x8001:02	0x8000:01 + 0x8001:02	Register reload + Reload Value
2	Operational	11	п	п	п	0x8000:02	Index Reset
3	Operational	п	II	11	II	0x8000:03, :04, :05	FWD Cnt + pos/ neg Gate
4	Operational	0x1A01 + 0x1601	16 bit Value/Latch Word Alignment	0x1A02	Frequency: 32 bit Period: 16 bit Window: 16 bit CoE object 0x8001:02	0x8000:01 + 0x8001:02	Register reload + Reload Value
5	Operational	"	н	"	н	0x8000:02	Index Reset
6	Operational	П	II	II	Π	0x8000:03, :04, :05	FWD Cnt + pos/ neg Gate

818

NOTE

Frequency

- The timeframe for the frequency calculation is set to 10 ms (see Fig. "Process data" tab); in addition a variably configurable measuring window is available (parameterization via object 0x8001:01, output frequency value in object 0x6000:06).
- Only the increment edges in the specified time window are counted.
- If no edge change occurs for approx. 1.6 s, any frequency specification is cancelled.
- This calculation is carried out in the slave without reference to the distributed clocks system. It is therefore independent of the DC mode.
- No frequency measurement is possible if the counter is blocked by the gate. In this case the period can be measured regardless.
- If an encoder signal is only present at input A/A and the frequency/period is to be measured, the terminal must be set to "Enable FWD count" in CoE 0x8000:03.
- A C or external reset restarts the frequency measurement. The last frequency value remains unchanged until a new frequency value is determined.

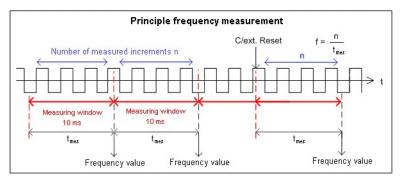


Figure 9-22: Frequency measurement principle in normal operation mode

Period calculation

- This calculation is carried out in the slave without reference to the distributed clocks system. It is therefore independent of the DC mode.
- In each cycle the interval between 2 positive edges of input A is counted with a resolution of 100 ns.
- If no edge change occurs for approx. 1.6 s, any period specification is canceled.
- NOTE

Frequency and period measurement

From the explanatory notes above it is apparent that the frequency measurement can measure the current axis status (velocity) significantly more accurately than the period measurement. Frequency measurement is therefore preferable, if possible.

Register Reload

• If Register Reload is enabled in CoE object 0x8000:01, the counter value is set to zero in the event of overflow over the value in CoE object 0x8001:02 and to the value in CoE object 0x8001:02 in the event of underflow below 0.

Index Reset

If Index Reset is enabled in CoE object 0x8000:02, input C resets the counter to 0.
 NOTE

"Register Reload" and "Index Reset"

"Register Reload" and "Index Reset" cannot be operated simultaneously.

FWD Cnt

- If FwdCnt is activated in CoE object 0x8000:03, the AKT2G-ENC-190-000 operates as counter on channel A. Channel B indicates the counting direction: B=TRUE forward, B=FALSE backward. The counter can be locked via the gate input (24 V).
 - CoE object 0x8000:04 (TRUE): Locking of the counter at the gate input with positive edge (0 V -> + 24 V).
 - CoE object 0x8000:05 (TRUE): Locking of the counter at the gate input with negative edge (+ 24 V -> + 0 V).

14.6.13.5.1.2 Object Description and Parameterization - Normal Operation Mode

14.6.13.6 Restore object

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore the default settings	UINT8	RO	0x01 (1 _{dec})
1011:01	SubIndex 001	If this object is set to "0x64616F6C" in the set value dialog, all backup objects are reset to their delivery state. Note: Some FW versions also accept the following input: "0x6C6F6164".	UINT32	RW	0x00000000 (0dec)

14.6.13.7 Configuration data

Index 8000 Non-Volatile Settings 0

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:0	Non- Volatile Settings 0	Maximum subindex	UINT8	RO	0x05 (5 _{dec})
8000:01	Enable register reload	The counter counts up to the "Counter reload value", or the "Counter reload value" (0x8001:02) is loaded in the event of an underflow Example 360° encoder with set bit:	BOOLEAN	RW	0x00 (0 _{dec})
		Moves in positive direction via Counter reload value: Reset counter value to 0.			
		Moves in negative direction less than 0: Reset counter value to Counter reload value			
8000:02	Enable index reset	Activates input "C" for resetting the counter.	BOOLEAN	RW	0x00
		Example 360° encoder with set bit:			(0 _{dec})
	reset	Moves in positive direction (signal at input "C"): Reset counter value to 0			
		Moves in negative direction (signal at input "C"): underflow with FFFF, FFFE etc.)			
8000:03		FALSE	BOOLEAN	RW	0x00
	FWD count	The terminal operates in quadrature decoder mode			(0 _{dec})
	count	TRUE			
		The terminal operates as counter, count direction to input B			
8000:04	Enable pos. gate	Gate input responds to positive edge and locks the counter	BOOLEAN	RW	0x01 (1 _{dec})
8000:05	Enable neg. gate	Gate input responds to negative edge and locks the counter	BOOLEAN	RW	0x00 (0 _{dec})

Index 8001 Non-Volatile Settings 1

Index (hex)	Name	Meaning	Data type	Flags	Default
8001:0	Non- Volatile Settings 1	Maximum subindex	UINT8	RO	0x02 (2 _{dec})
8001:01	Frequency window	The value specifies the size of the time window for the "Window" variable. resolution: 16 μ s; e.g. default value: 16 μ s x 100 _{dec} = 1.6 ms	UINT16	RW	0x0064 (100 _{dec})
8001:02	Counter reload value	If "Enable register reload" = TRUE, the counter counts up to this value and is loaded with this value in the event of an underflow	UINT16	RW	0xFFFF (65535 _{dec})

14.6.13.8 Input data

Index 6000 Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	Inputs	Length of this object	UINT8	RO	0x06 (6dec)
6000:01	Status	Status byte	UINT8	RO	0x00 (0 _{dec})
6000:02	Value	meter reading	UINT16	RO	0x0000 (0dec)
6000:03	Latch	Latch value	UINT16	RO	0x0000 (0 _{dec})
6000:04	Frequency	Frequency value (resolution: 0.01 Hz / digit) [fixed 10 ms measuring window]	UINT32	RO	0x00000000 (0 _{dec})
6000:05	Period	Period (resolution 500 ns / digit)	UINT16	RO	0x0000 (0 _{dec})
6000:06	Window	Measured value of the variable timeframe ("Frequency window" (0x8001:01)	UINT16	RO	0x0000 (0 _{dec})

14.6.13.9 Output data

Index 7000 Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	Outputs	Length of this object	UINT8	RO	0x02 (2 _{dec})
7000:01	Ctrl	Control byte	UINT8	RO	0x00 (0dec)
7000:02	Value	The counter value to be set via CNT_SET (CB.02).	UINT16	RO	0x0000 (0 _{dec})

14.6.13.9.0.1 Control and status byte

14.6.13.9.1 Control byte

The control byte (CB) is located in the output process image, and is transmitted from the controller to the terminal.

Bit	CB.7	CB.6	CB.5	CB.4	CB.3	CB.2	CB.1	CB.0
Name	-	-	-	-	EN_LATCH_EXTN	CNT_SET	EN_LATCH_EXTP	EN_LATC

Legend

Bit	Name	Descr	Description							
CB.3	EN_ LATCH_ EXTN	1bin	With a valid EN_LATCH_EXTN bit the counter value is stored in latch input (index 0x6000:03) when the first external latch pulse with falling edge is encountered. Subsequent pulses have no influence on the latch value. Please note: A new counter value can only be written to the latch input when the Latch Valid bit (LAT_EXT_VAL) in the status byte has a low signal level.							
CB.2	CNT_ SET	rise	The counter is set with rising edge of CNT_SET to the value specified via the process data (index 0x7000:02).							
CB.1	EN_ LATCH_ EXTP	1bin	With a valid EN_LATCH_EXTP bit the counter value is stored in latch input (index 0x6000:03) when the first external latch pulse with rising edge is encountered. Subsequent pulses have no influence on the latch value. Please note: A new counter value can only be written to the latch input when the Latch Valid bit (LAT_EXT_VAL) in the status byte has a low signal level.							
СВ.0	EN_ LATC	1bin	The null point latch (C input) is activated. The counter value is saved when the first external latch pulse after a valid EN_LATC bit encountered (this has priority over EN_LAT_EXTN / EN_LAT_EXTP). If the bit is set subsequent pulses have no influence on the latch value. Please note: A new counter value can only be written to the latch input when the Latch Valid bit (LATC_VAL) in the status byte has a low signal level (the LATC_VAL bit (SB.0) is only cleared by the terminal when the C pulse = LOW).							

14.6.13.9.2 Status byte

The status byte (SB) is located in the input process image, and is transmitted from terminal to the controller.

Bit	SB.7	SB.6	SB.5	SB.4	SB.3	SB.2	SB.1	SB.0
Name	-	-	STATUS_ INPUT	OVERFLOW	UNDERFLOW	CNTSET_ ACC	LAT_EXT_ VAL	LATC_ VAL

14.6.13.9.3 Legend

Bit	Name	Descri	Description					
SB.5	STATUS_ INPUT	0 bin /1 _{bin}	Indicates the status of INPUT 1					
SB.4	OVERFLOW	1 _{bin}	This bit is set if the 16-bit counter overflows (65535 to 0). It is reset when the counter exceeds one third of its measuring range (21845 to 21846) or immediately an underflow occurs.					
SB.3	UNDERFLOW	1 _{bin}	This bit is set if the 16-bit counter underflows (0 to 65535). It is reset when the counter drops below two thirds of its measuring range (43690 to 43689) or immediately an overflow occurs.					
SB.2	CNTSET_ACC	1 _{bin}	The data for setting the counter (index 0x7000:02) is accepted from the terminal.					

Bit	Name	Descr	iption
SB.1	LAT_EXT_VAL	1 _{bin}	An external latch pulse has occurred. The data with index 0x6000:03 match the latched value when the bit is set. To reactivate the latch input EN_LAT_EXTN or EN_LATCH_EXTP (CB.3 or CB.1) has first to be cleared and then to be set once more.
SB.0	LATC_VAL	1 _{bin}	A zero point latch has occurred. The data with index 0x6000:03 match the latched value when the bit is set. In order to reactivate the latch input, it is necessary for EN_LATC (CB.0) first be cleared and then to be set once more.

14.6.13.9.3.1 Single-Ended Connection for TTL Encoder

In addition to encoders with differential RS422 interface, single-ended encoders with TTL interface are also supported. Please note the following:

- Operating mode selection encoder "0x8000:03 Enable FWD count = FALSE".
- The differential inputs (/A, /B, /C) must remain open and must not be connected to ground

14.6.13.9.4 Enhanced Operation Mode

14.6.13.9.4.1 Process data and modes - Enhanced operation mode

In AKT2G-ENC-190-000 "enhanced operation mode" the following modes are available:

Mode	DC	Main PDO	Comment	Optional PDO 1	Comment	Optional PDO 2	Comment	Features CoE	Comment
7	FreeRun	0x1A04 + 0x1603	Count/Latch in 32 bit	0x1A05 or 0x1A06	Frequency (32 bit) or Period (32bit)			0x80n0:nn	CoE combinations 0x80n0:nn
8	п	0x1A03 + 0x1602	compact: Count/Latch in 16 bit	п	"			п	n
9	DC/DCi	0x1A04 + 0x1603	Count/Latch in 32 bit	II	u	0x1A07 or 0x1A08	Timestamp 64 bit Timestamp 32 bit (compact)	II	n
10	II	0x1A03 + 0x1602	compact: Count/Latch in 16 bit	"	II	"	п	Π	п

14.6.13.10 Frequency

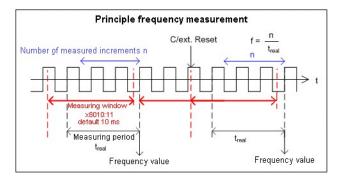
- The time window for the frequency calculation and the resolution can be parameterized in CoE objects 0x80n0:11 [" 141], 0x80n0:13 [" 141], 0x80n0:15 [" 141], 0x80n0:17 [" 141].
- The positive edges of track A are counted within the specified timeframe and the next edge including the time up to it are counted. The waiting time can be set in CoE object 0x80n0:17 "Frequency Wait Time" (unit: ms). The default value is 1.6 sec. This is also the maximum value.
- The time window is 10 ms (default), min. 1 µs. With the default setting it is possible to measure frequencies up to approx. 800 kHz. At higher frequencies a smaller value must be selected for the timeframe.
- The time is measured with a resolution of 100 ns.
- This calculation is carried out in the slave without reference to the distributed clocks system. It is therefore independent of the DC mode.
- No frequency measurement is possible if the counter is blocked by the gate. In this case the period can be measured regardless.
- If an encoder signal only is only present at input A/A and the frequency/period is to be measured, the terminal must be set to "Up/Down Counter" in CoE 0x8010:03 [" 141].
- A C or external reset restarts the frequency measurement. The last frequency value remains unchanged until a new frequency value is determined.

14.6.13.11 Frequency measurement

• Basic unit 1 µs: all window sizes

14.6.13.12 Measurement sequence

- The measurement starts with a positive edge at track A. The current counter value and time (resolution: 100 ns) are stored.
- After the measuring window time has elapsed (index 0x80n0:11 [" 141]), the system waits for the following rising edge at track A or a maximum of 1.6 sec or the time from 0x80n0:17 [" 141]
- The frequency is calculated from the edge difference and the actual elapsed time.



14.6.13.13 Period calculation

- This calculation is carried out in the slave without reference to the distributed clocks system. It is therefore independent of the DC mode.
- In each cycle the interval between 2 positive edges of input A is counted with a resolution of 100 ns.
- If no edge change occurs for approx. 1.6 s, any period specification is cancelled.

NOTE

Frequency and period measurement

From the explanatory notes above it is apparent that the frequency measurement can measure the current axis status (velocity) significantly more accurately than the period measurement. Frequency measurement is therefore preferable, if possible.

14.6.13.14 Latch

Activation of latch C input ("C") and saving ("latching") of the counter value (index 0x70n0:01 [" 145])

- The counter value is saved at the first external latch pulse (positive edge at input "C") after the bit has been set ("TRUE") in index 0x70n0:01 [" 145] (has priority before 0x70n0:02 [" 145] / 0x70n0:04 [" 145]). The subsequent pulses at the other inputs have no influence on the latch value in index 0x60n0:12 [" 144] if the bit is set.
- Note for "Latch C valid" bit: A new counter value at the latch input can only be written once the value of the "Latch C valid" bit (index 0x60n0:01 [* 144]) is "FALSE".

Activation of the external latch input ("gate/latch") and latching of the counter value (index 0x70n0:02 [" 145], 0x70n0:04 [" 145])

- The counter value at the latch input (Index 0x70n0:02 [" 145]) will be saved upon the first external latch pulse with a rising edge if the bit ("TRUE") is set in index 0x60n0:12 [" 144]. The subsequent pulses have no influence on the latch value in index 0x60n0:12 [" 144].
- The counter value at the latch input (Index 0x60n0:12 [" 144]) will be saved upon the first external latch pulse with a falling edge if the bit ("TRUE") is set in index 0x70n0:04 [" 145]. The subsequent pulses have no influence on the latch value in index 0x60n0:12 [" 144].
- Note for "Latch extern valid" bit: A new counter value at the latch input can only be written once the value of the "Latch extern valid" bit (index 0x60n0:02 [* 144]) is "FALSE".

14.6.13.15 Reset

- Counter reset (index 0x80n0:01 [" 141], 0x80n0:02 [" 141], 0x80n0:10 [" 141]): For a counter reset via input C set the bit in index 0x80n0:01 [" 141], for a reset via the external latch input set the bit in index 0x80n0:02 [" 141].
- The functions "Enable C reset" (0x80n0:01 [" 141]) and "Enable extern reset" (0x80n0:02 [" 141]) cannot be activated simultaneously.
- Note for "Extern reset polarity", index 0x80n0:10 [" 141]: The edge for setting the counter to zero can be selected via index 0x80n0:10 [" 141].

Bit not set: counter is set to zero with falling edge.

Bit set: counter is set to zero with rising edge.

14.6.13.16 Up/down counter

- The mode (encoder or up/down counter) is set via the CoE objects (profile-specific objects, tab CoE Online, index 0x80n0:03 [" 141] "Enable up/down counter"). Click on the corresponding row of the index to be parameterized, enter 1 in the SetValue dialog and confirm with OK.
- Set the gate polarity accordingly via object 0x80n0:04 [" 141].
- An additional option for reversing the rotation direction is available by setting the bit in index 0x80n0:0E [" 141].

14.6.13.17 Overflow/underflow

- Overflow/underflow control is inactive in combination with an activated reset function (C/external).
- The underflow bit (0x60n0:04 [" 144]) is set if an underflow ...00 → ...FF occurs. It is reset if 2/3 of the counter range are underrun.
- The overflow bit (0x60n0:05 [" 144]) is set if an overflow FF...→ 00... occurs. It is reset if 1/3 of the counter range is exceeded.

14.6.13.18 Open circuit detection

- A separate open circuit detection can be activated for each of the channels A, B and C (index 0x80n0:0B [" 141], 0x80n0:0C [" 141], 0x80n0:0D [" 141]).
- Open circuit detection is activated for channels A and B by default.
- A differential voltage of typically -1.5 V >Vid > +1.5 V is detected as an open circuit.
- If an open circuit is detected, it is indicated as process data open circuit = TRUE. The bit in object 0x60n0:07 [" 144] is set. An open circuit is indicated separately in indices 0xA0n0:01 [" 145] (track A), 0xA0n0:02 [" 145] (track B) and 0xA0n0:03 [" 145](track C).
- TxPDO state also becomes TRUE if an open circuit is detected, since invalid data have to be assumed.

NOTE

Open circuit detection vs. single-ended lines (TTL interface)

The open circuit detection does principally not work with single-ended lines (TTL interface).

14.6.13.19 Micro-increments

- Works with and without distributed clocks, but this is only meaningful in conjunction with one of the DC modes
- By setting the counter value only the integer component can be modified.

• The principle:

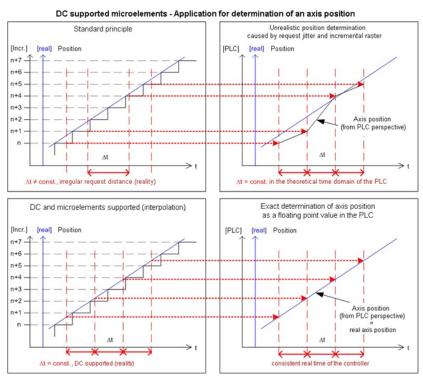


Figure 9-23: Frequency measurement principle in enhanced operation mode

The highly constant query cycles (accuracy: 100 ns) of the distributed clocks systems enable the AKT2G-ENC-190 to interpolate axis positions between the counted encoder increments from a certain speed. The interpolation resolution is 8 bit, corresponding to 256 values. A standard encoder with 1,024 bars with 4way evaluation and micro-increments thus becomes a high-resolution axis encoder with 4096 * 256 = 1,048,567 bars.

Underrunning of the minimum velocity is indicated by the object 0x60n0:08 [" 144] (extrapolation stall) in the process data.

14.6.13.20 Configuration Data

14.6.13.20.1	Index 8010 ENC Settings
--------------	-------------------------

Index (hex)	Name	Meaning	Data type	Flags	Default
8010:0	ENC Settings	Maximum subindex	UINT8	RO	0x17 (32dec)
8010:01	Enable C reset	The counter is reset via the C input.	BOOLEAN	RW	0x00 (0dec)
8010:02	Enable extern reset	A counter reset is triggered via the external latch input (24 V)	BOOLEAN	RW	0x00 (0dec)
8010:03	Enable up/down counter	Enablement of the up/down counter in place of the encoder with the bit set. Increments are counted at input A. Input B specifies the counting direction.	BOOLEAN	RW	0x00 (0dec)
8010:04		0: Disable gate	BIT2	RW	0x01
	polarity	1: Enable pos. gate (gate locks with "HIGH" level)			(1dec)
		2: Enable neg. gate (gate locks with "LOW" level)			
8010:08	Disable filter	0: Activates the input filter (inputs A, /A, B, /B, C, /C only)	BOOLEAN	RW	0x01 (1dec)
		1: Deactivates the input filter			
		If a filter is activated a signal edge must be present for at least 2.4 µs in order to be counted as an increment.			
8010:0A	micro	If DC mode is activated, the AKT2G-ENC-190 interpolates micro-increments between the integer encoder increments. The lower 8 bits of the counter value are used in each case for the display. A 32-bit counter thus becomes a 24+8-bit counter, a 16-bit counter becomes an 8+8-bit counter.	BOOLEAN	RW	0x00 (0dec)
8010:0B	Open circuit detection A	An open circuit on track A is indicated in index 0x6010:07 and as process data. Diagnosis is only possible if the corresponding input is wired differentially. A differential voltage < 3.5 V (typical, subject to change) is detected as a broken wire.	BOOLEAN	RW	0x01 (1dec)
8010:0C	Open circuit detection B	An open circuit on track B is indicated in index 0x6010:07 and as process data. Diagnosis is only possible if the corresponding input is wired differentially. A differential voltage < 3.5 V (typical, subject to change) is detected as a broken wire.	BOOLEAN	RW	0x01 (1dec)

Index (hex)	Name	Meaning	Data type	Flags	Default
8010:0D	circuit	An open circuit on track C is indicated in index 0x6010:07 and as process data. Diagnosis is only possible if the corresponding input is wired differentially. A differential voltage < 3.5 V (typical, subject to change) is detected as a broken wire.	BOOLEAN	RW	0x00 (0dec)
8010:0E	Reversion of rotation	Activates reversion of rotation	BOOLEAN	RW	0x00 (0dec)
8010:10	Extern reset polarity	0: Fall (the counter is set to zero with a falling edge)1: Rise (the counter is set to zero with a rising edge)	BIT1	RW	0x01 (1dec)
8010:11	Frequency window	This is the minimum time over which the frequency is determined. Default 10 ms [resolution: 1 μs]	UINT16	RW	0x2710 (10000dec)
		The number of pulses in the time window + the following is measured. The maximum waiting time is specified in the "Frequency Wait Time" parameter. The number of pulses is divided by the actual time window size. The determined frequency is output in index 0x6010:13 and as a process data.			
		The frequency calculation is carried out locally without distributed clocks function.			
8010:13	Frequency scaling	Scaling of the frequency measurement (must be divided by this value to obtain the unit in Hz): 100: "0.01 Hz"	UINT16	RW	0x0064 (100dec)
8010:14		Resolution of the period in the process data:	UINT16	RW	0x0064
	scaling	100: "100 ns" period value is a multiple of 100 ns 500: "500 ns" period value is a multiple of 500 ns			(100dec)
8010:15	Frequency resolution	Resolution of the frequency measurement: 100: "0.01 Hz"	UINT16	RW	0x0064
		100. 0.01112			(100dec)

Index (hex)	Name	Meaning	Data type	Flags	Default
8010:16	Period resolution	Internal resolution of the period measurement:	UINT16	RW	0x01F4 (500dec)
		100: "100 ns" period value is a multiple of 100 ns			(555666)
		The period is calculated internally with a resolution of 100 ns. The max. measurable period can then be approx. 1.6 seconds.			
		500: "500 ns" period value is a multiple of 500 ns Internally the period is calculated with 500 ns resolution. The maximum measurable period is approx.			
		32.7 ms. The resolution of process data continues to be the value according to index 0x8010:14 (e.g. 100 ns [default]).			
8010:17	Frequency Wait Time	Waiting time [ms] for frequency measurement	UINT16	RW	0x0640 (1600dec)
		Once the time specified in the frequency window has elapsed, the system waits for the next positive edge from track A. This enables the update speed for the Frequency process data to be optimized, depending on the expected frequencies. At least double the period of the minimum frequency to be measured should be entered here. t >= $2*(1 / fmin)$			()

14.6.13.20.2 Ir	dex 6010 ENC Inputs
-----------------	---------------------

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:0	ENC Inputs	Maximum subindex	UINT8	RO	0x16 (22dec)
6010:01	Latch C valid	The counter value was locked with the "C" input. The data with index 0x6010:12 match the latched value when the bit is set. To reactivate the latch input, index 0x7010:01 must be cancelled and then reset.	BOOLEAN	RO	0x00 (0dec)
6010:02	Latch extern valid	The counter value was locked via the external latch. The data with index 0x6010:12 match the latched value when the bit is set. To reactivate the latch input, index 0x7000:02 or object index 0x7000:04 must be cancelled and then reset.	BOOLEAN	RO	0x00 (0dec)
6010:03	Set counter done	The counter was set.	BOOLEAN	RO	0x00 (0dec)

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:04		Counter underflow.	BOOLEAN	RO	0x00 (0dec)
	underflow	Overflow/underflow control is inactive in combination with a reset function (C/external).			
6010:05		Counter overflow.	BOOLEAN	RO	0x00 (0dec)
	overflow	Overflow/underflow control is inactive in combination with a reset function (C/external).			
6010:06	Status of input status	State of the status input (alarm "input 1")	BOOLEAN	RO	0x00 (0dec)
6010:07	Open circuit	Indicates an open circuit.	BOOLEAN	RO	0x00 (0dec)
		Configuration via index 0x8010:0A, 0x8010:0B, 0x8010:0C			
6010:08	Extrapolation stall	The extrapolated part of the counter is invalid	BOOLEAN	RO	0x00 (0dec)
6010:09	Status of input A	Status of input A	BOOLEAN	RO	0x00 (0dec)
6010:0A	Status of input B	Status of input B	BOOLEAN	RO	0x00 (0dec)
6010:0B	Status of input C	Status of input C	BOOLEAN	RO	0x00 (0dec)
6010:0C	Status of input gate	The state of the gate input	BOOLEAN	RO	0x00 (0dec)
6010:0D	Status of extern latch	Status of the extern latch input	BOOLEAN	RO	0x00 (0dec)
6010:0E	Sync Error	The Sync error bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle.	BOOLEAN	RO	0x00 (0dec)
		This means a SYNC signal was triggered in the AKT2G-ENC-190, although no new process data were available (0=OK, 1=NOK).			
6010:0F	TxPDO State	Validity of the data of the associated TxPDO (0 = valid, 1 = invalid).	BOOLEAN	RO	0x00 (0dec)
6010:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0dec)
6010:11	Counter value	Counter value	UINT32	RO	0x00000000 (0dec)
6010:12	Latch value	Latch value	UINT32	RO	0x00000000 (0dec)
6010:13	Frequency value	The frequency (setting of the scaling and resolution in index 0x8010:13 and 0x8010:15)	UINT32	RO	0x00000000 (0dec)

832

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:14	Period value	The period (setting of the scaling and resolution in index 0x8010:14 and 0x8010:16)	UINT32	RO	0x00000000 (0dec)
6010:16	Timestamp	Timestamp of the last counter change	UINT64	RO	

14.6.13.20.3 Index 7010 ENC Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7010:0	ENC Outputs	Maximum subindex	UINT8	RO	0x11 (17dec)
7010:01	Enable latch C	Activate latching via input "C".	BOOLEAN	RO	0x00 (0dec)
7010:02	Enable latch extern on positive edge	Activate external latch with positive edge.	BOOLEAN	RO	0x00 (0dec)
7010:03	Set counter	Set counter	BOOLEAN	RO	0x00 (0dec)
7010:04	Enable latch extern on negative edge	Activate external latch with negative edge.	BOOLEAN	RO	0x00 (0dec)
7010:11	Set counter value	The counter value to be set via "Set counter" (index 0x7010:03).	UINT32	RO	0x00000000 (0dec)

14.6.13.21 Information / Diagnostic Data (Channel Specific)

14.6.13.21.1 Index A010 ENC Diag data

Index (hex)	Name	Meaning	Data type	Flags	Default
A010:0	ENC Diag data	Maximum subindex	UINT8	RO	0x03 (3dec)
A010:01	Open circuit A	Open circuit on track A	BOOLEAN	RO	0x00 (0dec)
A010:02	Open circuit B	Open circuit on track B	BOOLEAN	RO	0x00 (0dec)
A010:03	Open circuit C	Open circuit on track C	BOOLEAN	RO	0x00 (0dec)

14.6.13.21.1.1 Single-ended connection for TTL encoder

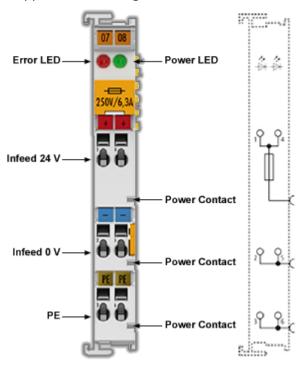
In addition to encoders with differential RS422 interface, single-ended encoders with TTL interface are also supported. Please note the following:

- Operating mode selection encoder "0x80n0:03 [" 141] Enable up/down counter = FALSE".
- The differential inputs (/A, /B, /C) must remain open and must not be connected to ground
- The option to reverse the direction of rotation is given by setting the bit in index 0x80n0:0E [" 141] "reversion of rotation =TRUE".
- The open circuit detection does not work in principle conditionally with single-ended connection

14.6.14 AKT2G-PSF-024-000

Power supply terminal with fuse, 24 V DC

The AKT2G-PSF-024 feed terminal can be positioned at any location between the input and output terminals for establishing a further potential group or for supplying the terminals following on the right in applications with high current load. The E-Bus is looped through.



Technical Data	AKT2G-PSF-024-000
Nominal voltage	24 VDC
Power contact current load	max. 10 A
Electrical isolation	500 V (E-bus/field potential)
Integrated fine-wire fuse	yes; 6.3 A
Current consumption from E-Bus	-
Bit width in the process image	-
Configuration	no address or configuration settings
Power LED	yes
Diagnosis (fuse)	yes, Error LED
Electrical connection to mounting rail	no
PE contact	yes
Renewed infeed	yes
Connection facility to additional power contact	1
Side by side mounting on Bus Terminals with power contact	yes

Technical Data	AKT2G-PSF-024-000
Side by side mounting on Bus Terminals without power contact	yes
Weight	approx. 55 g
Permissible ambient temperature range (during operation)	0°C +55°C
Permissible ambient temperature range (during storage)	-25°C +85°C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Mounting	on 35 mm mounting rail conforms to EN 60715
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC resistance burst/ESD	conforms to EN 61000-6-2/EN 61000-6-4
Protection class	IP 20
Installation position	variable, see Positioning of passive Terminals
Approval	CE, ATEX, cULus

Hazard to individuals and devices!

When designing a Bus Terminal block with different potentials on the power contacts (e.g. 230 V AC and 24 V DC), please note that it is mandatory to use potential separation terminals! Bring the bus system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

Terminal Point		Description					
Indication	Number						
+24 V	1	Supply input + 24 V (connected internally with terminal 4 and positive power contact)					
0 V	2	0V for supply input (connected internally with terminal 5 and negative power contact)					
PE	3	PE (connected internally with terminal 6 and PE power contact)					
+24 V 4		Supply input + 24 V (connected internally with terminal 1 and positive power contact)					
0 V 5		0 V for supply input (connected internally with terminal 2 and negative power contact)					
PE	6	PE (connected internally with terminal 3 and PE power contact)					

14.6.14.1 Connections

14.6.14.2 LEDs

LED	Color	Meaning				
Power LED	green	off No input voltage at supply input				
		on 24 VDC at supply input				
Error LED	red	off Fuse OK				
		on	Fuse error			

14.6.15 AKT2G-SDI-004-000

14.6.15.1 Safety terminal with 4 digital fail-safe inputs

The AKT2G-SDI-004 is a digital input terminal with floating contacts for 24 V DC. The Bus Terminal has 4 fail-safe inputs.

With two-channel connection, the AKT2G-SDI-004 meets the requirements of IEC 61508:2010 SIL 3, DIN EN ISO 13849-1:2015 (Cat 4, PL e), UL508, UL1998 and UL991.

The safety terminal has the typical design of an EtherCAT terminal.

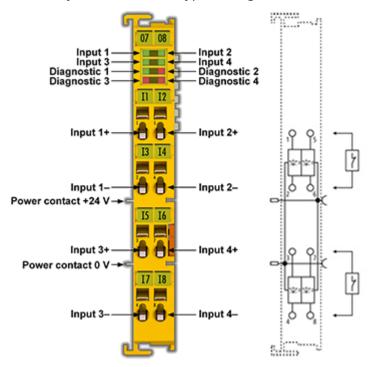


Figure 9-24: AKT2G-SDI-004 – safety terminal with 4 digital fail-safe inputs

★ TIP

Be sure to review

- Safety Instructions
- Safety Operation
- Safety Terminal Reaction Times
- Safety I/O Maintenance

14.6.15.2 Intended use

Caution - Risk of injury!

Safety components may only be used for the purposes described below!

The safety terminals expand the application range of the E-Bus terminal system with functions that enable them to be used for machine safety applications. The safety terminals are designed for machine safety functions and directly associated industrial automation tasks. They are therefore only approved for applications with a defined fail-safe state. This safe state is the wattless state. Fail-safety according to the relevant standards is required.

The safety terminals enable connection of:

- 24 VDC sensors (AKT2G-SDI-004) such as emergency off pushbutton switches, pull cord switches, position switches, two-hand switches, safety mats, light curtains, light barriers, laser scanner, etc.
- 24 VDC actuators (AKT2G-SDO-004) such as contactors, protection door switches with tumbler, signal lamps, servo drives, etc.

NOTE

Test pulses

When selecting actuators please ensure that the AKT2G-SDO-004 test pulses do not lead to actuator switching or diagnostic message from the AKT2G-SDO-004.

The following safety components have been developed for these tasks:

- The AKT2G-SDI-004 is an EtherCAT Terminal with 4 digital fail-safe inputs.
- The AKT2G-SDO-004 is an EtherCAT Terminal with 4 digital fail-safe outputs.

These safety components are suitable for operation on the

• Kollmorgen AKT2G-ECT-000-000 series Bus Couplers

Power supply from SELV/PELV power supply unit!

The safety components must be supplied with 24 VDC by an SELV/PELV power supply unit with an output voltage limit Umax of 36 VDC. Failure to observe this can result in a loss of safety.

Follow the machinery directive!

The safety components may only be used in machines as defined in the machinery directive.

Ensure traceability!

The buyer has to ensure the traceability of the device via the serial number.

14.6.15.3 Technical Data

Product designation	AKT2G-SDI-004			
Number of inputs	4			
Status display	4 (one green LED per input)			
Reaction time (read input/write to E-	typically: 4 ms,			
bus)	maximum: see error reaction time			
Error reaction time	≤ watchdog time			
Cable length between sensor and terminal	unshielded max. 100 m (0.75 or 1 mm ²) shielded max. 100 m (0.75 or 1 mm ²)			
Output current of the clock outputs	typically 10 mA, max. 15 mA			
Input process image	6 bytes			
Output process image	6 bytes			
AKT2G-SDI-004 supply voltage (PELV)	24 VDC (–15% / +20%)			
Signal voltage "0" inputs	-3 V to 5 V (EN 61131-2, type 3) see section Characteristic curve of the inputs			
Signal voltage "1" inputs	11 V to 30 V (EN 61131-2, type 3) see section Characteristic curve of the inputs			
Current consumption of the modular	4 channels occupied: typically 12 mA			
electronics at 24 V (without current consumption of sensors)	0 channels occupied: typically 1.4 mA			
Current consumption via E-bus	4 channels occupied: approx. 200 mA			
Power dissipation of the terminal	typically 1 W			
Electrical isolation (between the channels)	no			
Electrical isolation (between the channels and the E-bus)	yes			
Insulation voltage (between the channels and the E-bus, under common operating conditions)	insulation tested with 500 VDC			
Dimensions (W x H x D)	12mm x 100mm x 68mm			
Weight	approx. 50 g			
Permissible ambient temperature (operation) up to SW 05	0 °C to +55 °C (see note in Installation position and minimum distances)			
Permissible ambient temperature (operation) from SW 06 (week 02/2014)	-25°C to +55 °C (see note in Installation position and minimum distances)			
Permissible ambient temperature (transport/storage)	-40°C to +70°C			
Permissible air humidity	5% to 95%, non-condensing			

Product designation	AKT2G-SDI-004		
Permissible air pressure	750 hPa to 1100 hPa		
(operation/storage/transport)	(this corresponds to a height of approx690 m to 2450 m over sea level assuming an international standard atmosphere)		
Climate category according to EN	3К3		
60721-3-3	(the deviation from 3K3 is possible only with optimal environmental conditions and also applies only to the technical data which are specified differently in this documentation)		
Permissible level of contamination	level of contamination 2		
according to EN 60664-1	(comply with the section Safety I/O Maintenance		
Impermissible operating conditions	Safety terminals must not be used under the following operating conditions:		
	• under the influence of ionizing radiation (that exceeds the level of the natural environmental radiation)		
	in corrosive environments		
	 in an environment that leads to unacceptable soiling of the Bus Terminal 		
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4		
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27		
Shocks	15 g with pulse duration 11 ms in all three axes		
Protection class	IP20		
Permitted operating environment	In the control cabinet or terminal box, with minimum protection class IP54 according to IEC 60529		
Permissible installation position	see section Installation position and minimum distances		
Approvals	CE, cULus		

14.6.15.4 Safety Parameters

Key Figures	AKT2G-SDI-004
Lifetime [a]	20
Prooftest Interval [a]	not required ¹
PFHD	1.11E-09
%SIL3	1.11%
PFD	8.29E-05
%SIL3	8.29 %
MTTFd	high
DC	high
Performance level	PLe
Category	4
HFT	1
Element classification ²	Туре В

1. Special proof tests are not required during the entire service life of the AKT2G-SDI-004 EtherCAT terminal.

2. Classification according to IEC 61508-2:2010 (chapter 7.4.4.1.2 and 7.4.4.1.3)

The AKT2G-SDI-004 EtherCAT Terminal can be used for safety-related applications within the meaning of IEC 61508:2010 up to SIL3 and EN ISO 13849-1 up to PL e (Cat4).

For the calculation or estimation of the MTTFd value from the PFHD value, further information can be found in ISO 13849-1:2015 Table K.1.

14.6.15.5 Characteristic curve of the inputs

The characteristic curve of the inputs is similar to type 3 according to EN 61131-2.

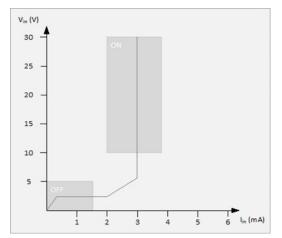


Figure 9-25: Characteristic curve of the inputs



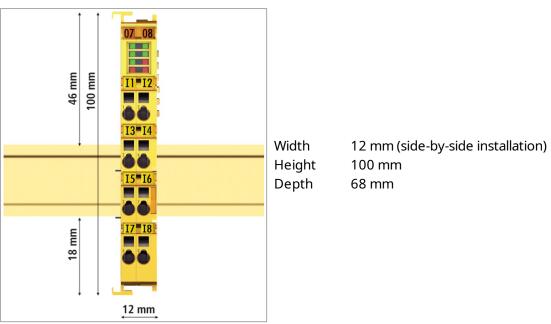
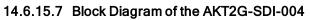


Figure 9-26: Dimensions of the AKT2G-SDI-004.



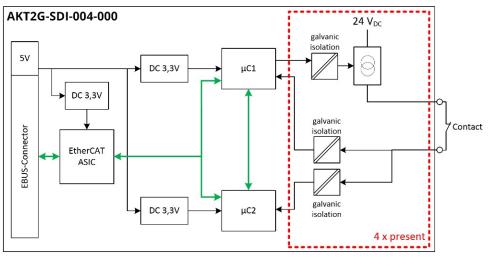


Figure 9-27: Block diagram of the AKT2G-SDI-004

The block diagram shows the basic configuration of a channel in the AKT2G-SDI-004. The part with a red border is present four times in the terminal.

14.6.15.8 AKT2G-SDI-004 Pin Assignment

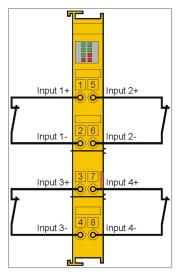


Figure 9-28: AKT2G-SDI-004 pin assignment

Terminal point	Input	Signal
1	1	Input 1+
2	-	Input 1-
3	3	Input 3+
4	-	Input 3-
5	2	Input 2+
6	-	Input 2-
7	4	Input 4+
8	1	Input 4-

NOTE

Configurable inputs

The inputs 1 to 4 can be occupied as you want with normally closed contacts or normally open contacts. The corresponding analysis is carried out in the safety PLC.

14.6.15.9 Signal Cables

14.6.15.9.1 Permitted cable length

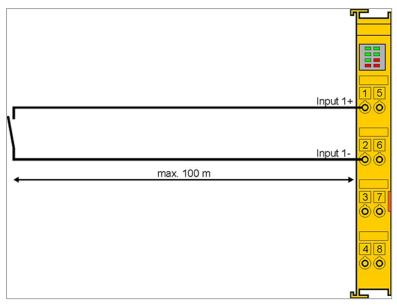


Figure 9-29: Permitted cable length

When connecting a single switching contact via its own continuous cabling (or via a non-metallic sheathed cable), the maximum permitted cable length is 100 m.

The use of contact points, connectors or additional switching contacts in the cabling reduces the maximum propagation.

14.6.15.9.2 Cable routing

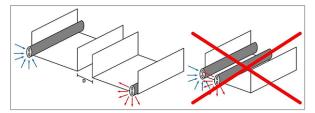


Figure 9-30: Cable routing

NOTICE

Route the signal cable separately

The signal cable must be routed separately from potential sources of interference, such as motor supply cables, 230 VAC power cables etc.!

Interference caused by cables routed in parallel can influence the signal form of the test pulses and thus cause diagnostic messages (e.g. sensor errors or OpenLoad errors).

- D: Distance between the cable ducts should be as large as possible
- blue arrows: signal line
- red arrows: potential source of interference

The common routing of signals together with other clocked signals in a common cable also reduces the maximum propagation, since crosstalk of the signals can occur over long cable lengths and cause diagnostic messages.

The test pulses can be switched off (sensor test parameter) if the connection of a common cable is unavoidable. However, this then leads to a reduction in the degree of diagnostic cover when calculating the performance level.

14.6.15.9.3 Test pulses

The typical length of a test pulse (switching from 24 V to 0 V and back to 24 V) is 350 µs and takes place approx. 250 times per second.

The test pulses at the outputs Input 1+ to Input 4+ are generated separately for each channel in order to be able to detect cross-circuits between the individual channels of a terminal and also between channels of different terminals. In order to generate test pulses as shown in the diagram, the sensor test active safety parameter must be set to true for the respective channels. The test cycle for all four channels is typically 4 ms. The times between the test pulses of different channels vary, thus allowing better diagnostic detection.

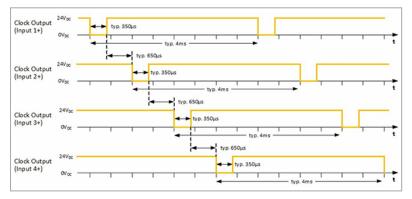


Figure 9-31: Typical course of test pulses of the inputs

If self-testing sensors are to be used on the safe inputs, please refer to chapter Configuration for light barriers, light grids, light curtains etc.

14.6.15.10 Address Settings on Safety Terminals with 65535 Possible Addresses

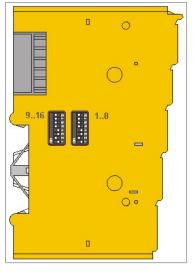


Figure 9-32: Address settings on safety terminals with 65535 possible addresses

Set the safety address for the terminal using the two dip switches (with 8 setting options) on the lefthand side of the AKT2G-SDI-004 safety terminal. Safety addresses between 1 and 65535 are available.

DIP switches										Address						
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
ON	OFF	OFF	OFF	OFF	OFF	OFF	1									
OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	2								
ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	3								
OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	4							
ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	5							
OFF	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	6							
ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	7							
OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	8						
ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	65535

846

Safety address

Each safety address may only be used once within a network! The address 0 is not a valid safety address!

14.6.15.11 Safety Terminal Address and Parameters

PrmName	Meaning	Values
FSoE_Address	DIP switch address	1 to 65535
Operating Mode	Digital / standstill monitoring 1 and 2	Digital / standstill 1 and 2
Sensor test channel 1 active	The clock signal for connection Input1+ is checked at connection Input1	true / false

PrmName	Meaning	Values
Sensor test channel 2 active	The clock signal for connection Input2+ is checked at connection Input2	true / false
Sensor test channel 3 active	The clock signal for connection Input3+ is checked at connection Input3	true / false
Sensor test channel 4 active	The clock signal for connection Input4+ is checked at connection Input4	true / false
Logic channel 1 and 2	Logic of channels 1 and 2	 single logic asynchronous repetition OSSD (sensor test must be switched off) any pulse repetition OSSD (sensor test must be switched off) short cut is no module fault
Logic channel 3 and 4	Logic of channels 3 and 4	 single logic asynchronous repetition OSSD (sensor test must be switched off) any pulse repetition OSSD (sensor test must be switched off) short cut is no module fault
Store Code	This parameter is required for the safety Restore Mode	0x0000
Project CRC	This parameter is required for the safety Restore Mode	0x0000

NOTE

Test pulses

If the parameters **Current Measurement active** or **Testing of outputs active** are set to TRUE, the terminal generates test pulses at the outputs. To avoid generating test pulses at the channel outputs, testing of outputs active and Current measurement active must be set to FALSE.

Please note that deactivating **Current measurement active** and/or **Testing of outputs active** may reduce the achievable performance level. A calculation example for the performance level can be found in the safety PLC application manual.

There are no known applications for which it would make sense to set **Testing of outputs active** to FALSE, while **Current measurement active** is set to TRUE.

14.6.15.11.1 AKT2G-SDI-004 configuration for light barriers, light grids, light curtains etc.

The AKT2G-SDI-004 also supports direct connection of contact-free protective devices with two selftesting outputs such as light barriers, light grids, light curtains, laser scanners, etc.

Sensors with self-testing outputs!

Only sensors with self-testing outputs and a maximum sensor self-test duration of 350 µs may be connected to the AKT2G-SDI-004 (see illustration below).

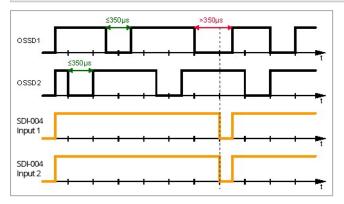


Figure 9-33: Maximum permissible sensor self-test duration of 350 µs

Parameter

To connect these sensors please set the following parameters for the AKT2G-SDI-004 in the safety PLC software:

Connect the two sensor signals either to channels 1 and 2 or channels 3 and 4 and *activate asynchronous repetition OSSD* or *any pulse repetition* for the two inputs used under parameter *Logic for channel x and y*. The difference between these settings is that with *any pulse repetition* simultaneous tests of the OSSD signals up to 350 µs are allowed.

For the two inputs used set the sensor test for the AKT2G-SDI-004 to False.

14.6.15.11.2 Configuration of the AKT2G-SDI-004 for safety switching mats

The AKT2G-SDI-004 also supports direct connection of safety switching mats.

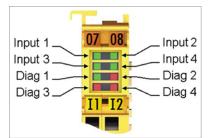
Parameter

To connect these switching mats please set the following parameters for the AKT2G-SDI-004 in the safety PLC software:

Connect the two sensor signals either to channels 1 and 2 or channels 3 and 4 and activate *short cut channel x/y is no module fault* for the two inputs used under parameter *Logic for channel x and y*.

14.6.15.12 AKT2G-SDI-004 Diagostic LEDs

The LEDs Diag 1 to Diag 4 display diagnostic information for the AKT2G-SDI-004.



14.6.15.12.1 Diag 1 (green)

The **Diag 1** LED indicates the state of the safety network.

Flashing Code	Meaning
LED illuminated continuously	normal operation:
	Safety communication OK
rapid flickering, alternating with 1 flash pulse	Error in S parameter (safety parameter)
rapid flickering, alternating with 2 flash pulses	Error in I parameter (Individual parameter)
rapid flickering, alternating with 3 flash pulses	Waiting for S and I parameter
rapid flickering, alternating with 4 flash pulses	S- and I-parameter correct: waiting for first host message
rapid flickering, alternating with 5 flash pulses	Watchdog error
rapid flickering, alternating with 6 flash pulses	CRC error
rapid flickering, alternating with 7 flash pulses	Sequence number error
rapid flickering, alternating with 8 flash pulses	Communication error in the safety protocol

14.6.15.12.2 Diag 2 (red)

The **Diag 2** LED illuminates red if the terminal detects an external supply or cross-circuit. The LED extinguishes once the error is rectified.

14.6.15.12.3 Diag 3 (red) and Diag 4 (red)

If the **Diag 3** LED is lit, the **Diag 4** LED indicates internal terminal errors.

14.6.15.12.3.1 Flashing Codes

In the case of such an error, the **Diag 4** LED on the AKT2G-SDI-004 displays flashing codes that describe the error in more detail.

A flashing code consists of four sequences, which are interrupted in each case by a short break. After the four sequences there is a long break, following which the flashing code is displayed again.

Count the individual sequences of the flashing code.

The errors indicated by the following flashing codes are reversible. After successful troubleshooting the terminal can be restarted.

Diag 3 LED	Diag 4 LED	Meaning	Remedy
	Flashing Code	Meaning	Remedy
lit	6-1-1-1	max. internal temperature exceeded	Ensure that the permissible ambient temperature is adhered to.
	7-1-1-1	internal temperature below min. value	
	2-1-2-1	max. supply voltage µC1 exceeded	Check the supply voltage.
	3-1-2-1	max. supply voltage µC2 exceeded	
	4-1-2-1	voltage fell below min. supply voltage μ C1	
	5-1-2-1	voltage fell below min. supply voltage μ C2	
	8-1-1-1	Temperature difference between the measuring points exceeded	Check the installation position and the ambient temperature.

If another flashing code is displayed, this means that there is an internal terminal error that has stopped the terminal. In this case the terminal must be checked by Kollmorgen.

NOTE

Note the flashing codes and return the terminal

Note the flashing code displayed and include this information with the terminal when you return it.

14.6.15.13 AKT2G-SDI-004 Diagostic Objects

Do not change CoE objects!

Do not make any modifications to the CoE objects in the safety components! Any modifications of the CoE objects will permanently set the safety components to the Fail-Stop state.

Index FA80_{hex}: Internal temperature values

The CoE object FA80_{hex} indicates the current internal temperature values of the AKT2G-SDI-004.

Index	Name	Meaning	Flags	Default
FA80:01	Temperature 1	Temperature measurement 1	RO	0 _{bin}
FA80:02	Temperature 2	Temperature measurement 2	RO	0 _{bin}

Index 800E_{hex}: diagnostic information

The CoE object 800E_{hex} displays further diagnostic information.

Index	Name	Meaning			Flags	Default
800E:0	Diag	The follow informatio	ving sub-indices contai n.	n detailed diagnostic	RO	
800E:0A	Sensor test error	rror Bit Error during the sensor test				
		0	1 _{bin}	Error at input 1		0 _{bin}
		1	1 _{bin}	Error at input 2		0 _{bin}
		2	1 _{bin}	Error at input 3		0 _{bin}
		3	1 _{bin}	Error at input 4		0 _{bin}
800E:0B	Error during two channel evaluation	Bit	two channels, i.e. the	Error during the contiguous evaluation of two channels, i.e. the two channels contradict each other.		
		0 1 _{bin} Error in the first input pair			0 _{bin}	
		1	1 _{bin}	Error in the second input pair		0 _{bin}
800E:0C	Error in the safety mat	Bits	Error in the input pair	,	RO	
	operation mode: input pair disagree	1, 0	11 _{bin}	Error in the first input pair		00 _{bin}
		3, 2	11 _{bin}	Error in the second input pair		00 _{bin}

Index	Name	Meaning			Flags	Default
800E:0D	Error in the safety mat operation mode: external supply	Bit	Error in the test pulses in the safety mat operating mode; i.e. the terminal has letected an external supply.		RO	
		0	1 _{bin}	Error at input 1		0 _{bin}
		1	1 _{bin}	Error at input 2		0 _{bin}
		2	1 _{bin}	Error at input 3		0 _{bin}
		3	1 _{bin}	Error at input 4		0 _{bin}

NOTE

Differing diagnostic messages possible

Due to the variable order or execution of the test series, diagnostic messages differing from those given in the table above are possible.

14.6.15.14 Certificates

The AKT2G-SDI-004-000 has been tested to the following directives and standards.

- 2006/42/EC
- EN 61508-1:2010 (up to SIL 3)
- EN 61508-2:2010 (up to SIL 3)
- EN 61508-3:2010 (up to SIL 3)
- EN 62061:2005/A2:2015 (up to SILCL 3)
- EN ISO 13849-1:2015 (Cat. 4, PL e)
- EN 81-20:2014
- EN 81-22:2014
- EN 81-50:2014
- EN 13243:2015

Note on approval accroding to EN 81-20, EN 81-22, and EN 81-50

- The Safety components may only be used in machines that have been designed and installed in accordance with the requirements of the EN 60204-1 standard.
- Provide a surge filter for the supply voltage of the Safety components against overvoltages. (Reduction to overvoltage category II)
- EN 81 requires that in the case of devices with internal temperature monitoring, a stop must be reached in the event of an overtemperature. In this case, passengers must be able to disembark (see EN 81-20 chapter 5.10.4.3, for example). To ensure this, application measures are necessary. The internal terminal temperature of the Safety components can be read out by the user. There is a direct switch-off at the maximum permissible temperature of the respective Safety component (see Installation position and minimum distances).
- The user must select a temperature threshold below the maximum temperature such that a stop can be reached in all cases before the maximum temperature is reached. Information on the optimum terminal configuration can be found under Notes on the arrangement of Safety components and under Example configuration for temperature measurement.
- For the use of the Safety components according to EN 81-22 and EN 81-50, the conditions described in the manuals for achieving category 4 according to EN ISO 13849-1:2015 must be observed.
- The use of Safety components is limited to indoor applications.

- Basic protection against direct contact must be provided, either by fulfilling protection class IP2X or by installing the Safety components in a control cabinet which corresponds at least to protection class IP54 according to EN 60529.
- The ambient conditions regarding temperature, humidity, heat dissipation, EMC and vibrations, as specified in the operating instructions under technical data, must be observed.
- The operating conditions in potentially explosive atmospheres (ATEX) are specified in the operating instructions.
- The safe state (triggering) of the application must be the de-energized state. The safe state of the Safety components is always the de-energized, switched-off state, and this cannot be changed.
- The service life specified in the operating instructions must be observed.
- If the Safety component is operated outside the permissible temperature range, it changes to "Global Shutdown" state.
- The Safety components must be installed in a control cabinet with protection class IP54 according to EN 60529, so that the requirement for contamination level 3 according to EN 60664-1 can be reduced to level 2.
- The Safety components must be supplied by a SELV/PELV power supply unit with a maximum voltage of Umax ≤ 36 VDC.

14.6.16 AKT2G-SDO-004-000

14.6.16.1 Safety terminal with 4 digital fail-safe outputs

The AKT2G-SDO-004 is a safe output terminal with digital outputs for connecting actuators (contactors, relays, etc.) with a maximum current 0.5 A (24 VDC). The Bus Terminal has 4 fail-safe outputs.

The AKT2G-SDO-004 meets the requirements of IEC 61508:2010 SIL 3, DIN EN ISO 13849-1:2015 (Cat 4, PL e), UL508, UL1998 and UL991.

The safety terminal has the typical design of an EtherCAT terminal.

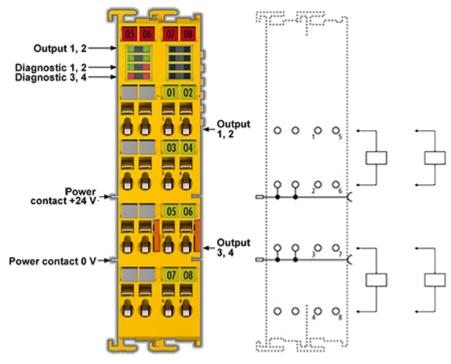


Figure 9-34: AKT2G-SDO-004-000 safety terminal with 4 digital fail-safe outputs

★ TIP

Be sure to review

- Safety Instructions
- Safety Operation
- Safety Terminal Reaction Times
- Safety I/O Maintenance

Description of Safety Parameters

Parameter Name	Description	Values
Standard outputs	In addition the safe output can be switched off from the PLC program. The safe outputs are logically ANDed with the standard digital outputs.	True / False
Current measurement active	Current measurement for the outputs are activated	True / False
Testing of outputs active	Test pulses for the outputs are activated	True / False

Parameter Name	Description	Values
Error acknowledge active	True: Terminal errors lead to a reset of the FSoE connection (error code 14 (0x0E)). This error code is shown in the diagnostic data for the connection until it is acknowledged. False (Default): Terminal errors can only be reset by switching the power supply off and back on again.	True / False

For example, using the BBH SafePLC2 software, the parameters can be configured from the AKT2G-SDO-004-000 Properties:

al.	Device			
	Logical Device Address	٠	1	-
	Cycle Time		8 ms	
4	FSoE			
	Slave Address		1	-
	Watchdog		200	÷
	Connection ID:		1	*
	RT Ethernet Bus		ECAT	Ŧ
	FC Connection			
	Standard Outputs Active			
	Current Measurement Active			
	Testing Of Outputs Active			
	Error Acknowldgement Active			
1	Miscellaneous			

14.6.16.2 Intended use

Caution - Risk of injury!

Safety components may only be used for the purposes described below!

The safety terminals expand the application range of the E-Bus terminal system with functions that enable them to be used for machine safety applications. The safety terminals are designed for machine safety functions and directly associated industrial automation tasks. They are therefore only approved for applications with a defined fail-safe state. This safe state is the wattless state. Fail-safety according to the relevant standards is required.

The safety terminals enable connection of:

- 24 VDC sensors (AKT2G-SDI-004) such as emergency off pushbutton switches, pull cord switches, position switches, two-hand switches, safety mats, light curtains, light barriers, laser scanner, etc.
- 24 VDC actuators (AKT2G-SDO-004 such as contactors, protection door switches with tumbler, signal lamps, servo drives, etc.

NOTE

Test pulses

When selecting actuators please ensure that the AKT2G-SDO-004 test pulses do not lead to actuator switching or diagnostic message from the AKT2G-SDO-004.

The following safety components have been developed for these tasks:

- The AKT2G-SDI-004 is an EtherCAT Terminal with 4 digital fail-safe inputs.
- The AKT2G-SDO-004 is an EtherCAT Terminal with 4 digital fail-safe outputs.

These safety components are suitable for operation on the

• Kollmorgen AKT2G-ECT-000-000 series Bus Couplers

Power supply from SELV/PELV power supply unit!

The safety components must be supplied with 24 VDC by an SELV/PELV power supply unit with an output voltage limit Umax of 36 VDC. Failure to observe this can result in a loss of safety.

Follow the machinery directive!

The safety components may only be used in machines as defined in the machinery directive.

Ensure traceability!

The buyer has to ensure the traceability of the device via the serial number.

14.6.16.3 Technical Data

Product designation	AKT2G-SDO-004-000
Number of outputs	4
Status display	4 (one green LED per output)
Error reaction time	≤ watchdog times
Output current per channel	max. 500 mA, min. 20 mA with parameter current measurement active set
Actuators	When selecting actuators please ensure that the AKT2G-SDO-004- 000 test pulses do not lead to actuator switching.
Cable length between actuator and terminal	unshielded max. 100 m shielded max. 100 m
Wire cross section	min. 0.75 mm ²
Input process image	6 bytes
Output process image	6 bytes
AKT2G-SDO-004-000 supply voltage (PELV)	24 VDC (-15%/+20%)
Current consumption via E-bus	approx. 221 mA
Power dissipation of the terminal	typically 2 W
Electrical isolation (between the channels)	no
Electrical isolation (between the channels and the E-bus)	yes
Insulation voltage (between the channels and the E-bus, under common operating conditions)	insulation tested with 500 V_{DC}
Dimensions (W x H x D)	24mm x 100mm x 68mm
Weight	approx. 100 g
Permissible ambient temperature (operation) up to SW 03	0°C to +55°C (see note in Installation position and minimum distances)
Permissible ambient temperature (operation) from SW 04 (week 02/2014)	-25°C to +55 °C (see note in Installation position and minimum distances)
Permissible ambient temperature (transport/storage)	-40°C to +70°C
Permissible air humidity	5% to 95%, non-condensing
Permissible air pressure	750 hPa to 1100 hPa
(operation/storage/transport)	(this corresponds to a height of approx690 m to 2450 m over sea level assuming an international standard atmosphere)

Product designation	AKT2G-SDO-004-000
Climate category according to EN	3К3
60721-3-3	(the deviation from 3K3 is possible only with optimal environmental conditions and also applies only to the technical data which are specified differently in this documentation)
Permissible level of contamination	level of contamination 2
according to EN 60664-1	(comply with the section Safety I/O Maintenance
Impermissible operating conditions	safety terminals must not be used under the following operating conditions:
	 under the influence of ionizing radiation (that exceeds the level of the natural environmental radiation) in corrosive environments in an environment that leads to unacceptable soiling of the Bus Terminal
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
Shocks	15 g with pulse duration 11 ms in all three axes
Protection class	IP20
Permitted operating environment	In the control cabinet or terminal box, with minimum protection class IP54 according to IEC 60529
Permissible installation position	see section Installation position and minimum distances
Approvals	CE, cULus

14.6.16.4 Safety Parameters

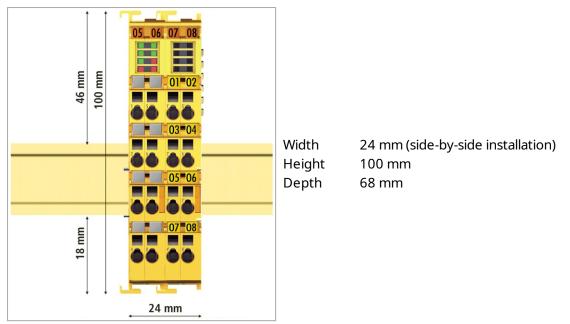
Key Figures	AKT2G-SDO-004-000
Lifetime [a]	20
Prooftest Interval [a]	not required ¹
PFH _D	1.25E-09
%SIL3	1,25 %
PFD	8.45E-05
%SIL3	8,45 %
MTTF _d	high
DC	high
Performance level	PLe
Category	4
HFT	1
Element classification ²	Туре В

1. Special proof tests are not required during the entire service life of the AKT2G-SDO-004 EtherCAT terminal.

2. Classification according to IEC 61508-2:2010 (chapter 7.4.4.1.2 and 7.4.4.1.3)

The AKT2G-SDO-004 EtherCAT Terminal can be used for safety-related applications within the meaning of IEC 61508:2010 up to SIL3 and EN ISO 13849-1 up to PL e (Cat4).

For the calculation or estimation of the MTTF_{d} value from the PFH_{D} value, further information can be found in ISO 13849-1:2015 Table K.1.



14.6.16.5 Dimensions

Figure 9-35: Dimensions of the AKT2G-SDO-004.

14.6.16.6 Block Diagram of the AKT2G-SDO-004

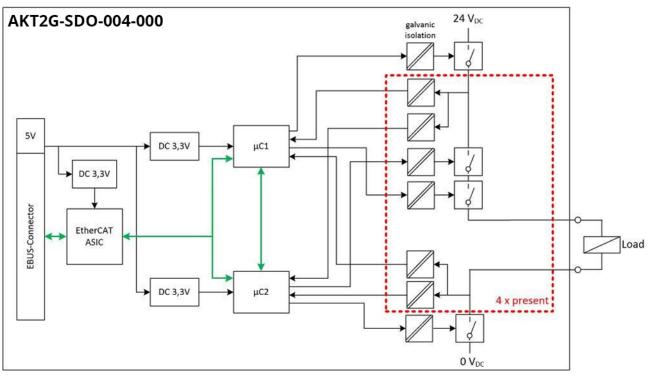
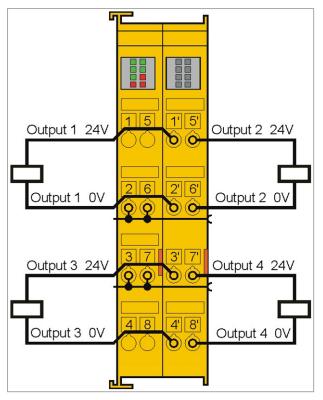


Figure 9-36: Block diagram of the AKT2G-SDO-004

The block diagram shows the basic configuration of a channel in the AKT2G-SDO-004. The part with a red border is present four times in the terminal. The high-side and low-side switches only exist once for all channels. This means that each channel has a total of four stop paths.



14.6.16.7 AKT2G-SDO-004 Pin Assignment

Figure 9-37: AKT2G-SDO-004 pin assignment

Terminal point	Output	Signal
1	-	not used, no function
2		positive power contact
3	-	negative power contact
4		not used, no function
5	-	not used, no function
6		positive power contact
7	-	negative power contact
8		not used, no function
1'	1	Output 1+
2'		Output 1-
3'	3	Output 3+
4'		Output 3-
5'	2	Output 2+
6'		Output 2-
7'	4	Output 4+
8'		Output 4-

860

NOTE

Test Pulses

When selecting actuators please ensure that the AKT2G-SDO-004 test pulses do not lead to actuator switching or diagnostic message from the AKT2G-SDO-004.

14.6.16.8 Signal Cables

14.6.16.8.1 Permitted cable length

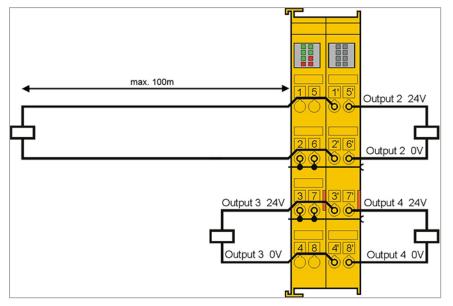


Figure 9-38: Permitted cable length

When connecting a single switching contact via its own continuous cabling (or via a non-metallic sheathed cable), the maximum permitted cable length is 100 m.

The use of contact points, connectors or additional switching contacts in the cabling reduces the maximum propagation.

14.6.16.8.2 Cable routing

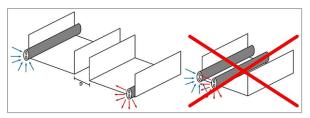


Figure 9-39: Cable routing

NOTICE

Route the signal cable separately

The signal cable must be routed separately from potential sources of interference, such as motor supply cables, 230 VAC power cables etc.!

Interference caused by cables routed in parallel can influence the signal form of the test pulses and thus cause diagnostic messages (e.g. sensor errors or OpenLoad errors).

- D: Distance between the cable ducts should be as large as possible
- blue arrows: signal line
- red arrows: potential source of interference

The common routing of signals together with other clocked signals in a common cable also reduces the maximum propagation, since crosstalk of the signals can occur over long cable lengths and cause diagnostic messages.

The test pulses can be switched off (sensor test parameter) if the connection of a common cable is unavoidable. However, this then leads to a reduction in the degree of diagnostic cover when calculating the performance level.

14.6.16.8.3 Test Pulses

The typical length of test pulse (switching from 24 V to 0 V and back to 24 V) is 300 μ s to 800 μ s. Testing usually takes place 5 to 7 times per second.

NOTE

862

Test pulses for the outputs

The following diagram shows a typical test pulse curve for the four outputs of an AKT2G-SDO-004. The parameters Current measurement active and Testing of outputs active are enabled.

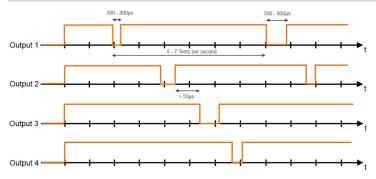


Figure 9-40: Typical course of test pulses of the outputs.

14.6.16.9 Address Settings on Safety Terminals with 1023 Possible Addresses

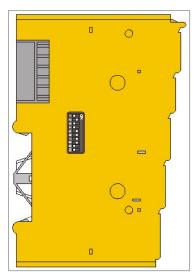


Figure 9-41: Address settings on safety terminals with 1023 possible addresses

The safety address of the terminal is set via the 10-way DIP switch on the left-hand side of the safety terminal. Safety addresses between 1 and 1023 are available.

DIP switch							Address			
1	2	3	4	5	6	7	8	9	10	
ON	OFF	OFF	OFF	1						
OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	4
ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	5
OFF	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	6
ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	7
ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	1023

Safety address

Each safety address may only be used once within a network! The address 0 is not a valid safety address!

14.6.16.10 Safety Terminal Address and Parameters

PrmName	Meaning	Values
FSoE_ Address	DIP switch address	1 to 1023
Standard outputs active	In addition the output can be switched off from the standard PLC. The safe output is linked with the standard logic signal AND.	true / false

PrmName	Meaning	Values
Current measurement active	Current measurement for the outputs is activated	true / false
Testing of outputs active	Test pulses for the outputs are activated	true / false
Error acknowledge active	True: Terminal errors lead to a reset of the safety connection (error code 14 (0x0E)). This error code is shown in the diagnostic data for the connection until the user acknowledges it via ErrAck.	true / false
	False (Default): Terminal errors can only be reset by switching the power supply off and back on again.	
Store Code	This parameter is required for the safety Restore Mode	0x0000
Project CRC	This parameter is required for the safety Restore Mode	0x0000

NOTE

Test pulses

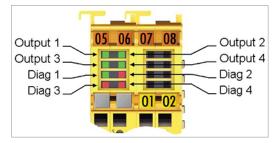
If the parameters **Current Measurement active** or **Testing of outputs active** are set to TRUE, the terminal generates test pulses at the outputs. To avoid generating test pulses at the channel outputs, testing of outputs active and Current measurement active must be set to FALSE.

Please note that deactivating **Current measurement active** and/or **Testing of outputs active** may reduce the achievable performance level. A calculation example for the performance level can be found in the safety PLC application manual.

There are no known applications for which it would make sense to set **Testing of outputs active** to FALSE, while **Current measurement active** is set to TRUE.

14.6.16.11 AKT2G-SDO-004 Diagostic LEDs

The LEDs Diag 1 to Diag 4 display diagnostic information for the AKT2G-SDO-004.



14.6.16.11.1 Diag 1 (green)

The **Diag 1** LED indicates the state of the safety network.

Flashing Code	Meaning
LED illuminated continuously	Diagnostic flashing code in preparation

14.6.16.11.2 Diag 2 (red)

The **Diag 2** LED indicates the state of the digital outputs.

Flashing Code	Meaning
rapid flickering, alternating with 1 flash pulse	Error at output 1
rapid flickering, alternating with 2 flash pulses	Error at output 2
rapid flickering, alternating with 3 flash pulses	Error at output 3
rapid flickering, alternating with 4 flash pulses	Error at output 4
rapid flickering, alternating with 5 flash pulses	Field voltage too low
rapid flickering, alternating with 6 flash pulses	Field voltage too high
rapid flickering, alternating with 7 flash pulses	Internal terminal temperature too low
rapid flickering, alternating with 8 flash pulses	Internal terminal temperature too high
rapid flickering, alternating with 9 flash pulses	Temperature difference error
rapid flickering, alternating with 10 flash pulses	Error in output circuit

14.6.16.11.3 Diag 3 (red) and Diag 4 (red)

If the **Diag 3** LED is lit, the **Diag 4** LED indicates internal terminal errors.

These errors lead to shutdown of the terminal. The terminal must be checked by Kollmorgen.

14.6.16.11.3.1 Flashing Codes

In the case of such an error, the **Diag 4** LED on the AKT2G-SDO-004 displays flashing codes that describe the error in more detail.

A flashing code consists of four sequences, which are interrupted in each case by a short break. After the four sequences there is a long break, following which the flashing code is displayed again.

Count the individual sequences of the flashing code.

NOTE

Note the flashing codes and return the terminal

Note the flashing code displayed and include this information with the terminal when you return it.

14.6.16.12 Possible Causes of Diagnostic Messages - AKT2G-SDO-004

Diagnostics	Possible cause	Remedial actions
Diag 2 LED Flash code 1 to 4 or	If parameters "Testing of outputs active" and/or "Current measurement active" are enabled:	
10	Faulty test pulses.	Eliminate cross-circuit or external supply.
	Cause: external supply or cross- circuit.	
	Faulty test pulses.	Isolate lines and lay in separate non-metallic
	Cause: parallel routed cables with high capacitive	sheathed cable. Create a distance between the non-metallic
	coupling and dynamized signals, possibly also in common cables	sheathed cables.
	Cause: Current is below the limit of 20	Select actuator accordingly.
	mA or above the limit of 500 mA.	Current > 20mA and
		< 500mA
	Regardless of whether the parameters "Testing of outputs active" and/or "Current measurement active" are enabled:	
	The output voltage lies below the	Eliminate short-circuit.
	permissible voltage range (24V - 15%/+20%).	Design power supply accordingly.
	A possible cause is a short-circuit at the output or e.g. a voltage drop at the instant of switching.	Check supply lines for voltage drop.
	EMC faults	Take suitable EMC measures
	Internal defect	Replace terminal
Diag 2 LED Flash code 5	Voltage at the power contacts not switched on.	Switch on voltage at the power contacts and reset the error display through PowerOn Reset of the terminal
	Voltage at the power contacts was switched on after the terminal supply	Switch on voltage at the power contacts before or at the same time as the terminal supply and reset the error display through PowerOn Reset of the terminal
	Voltage on the power contacts too low.	Increase the voltage at the power contacts and reset the error display through PowerOn Reset of the terminal
	EMC faults	Take suitable EMC measures
	Internal defect	Replace terminal

866

Diagnostics	Possible cause	Remedial actions	
Diag 2 LED Flash code 6	Field voltage too high. Voltage on the power contacts too high.	Reduce the voltage at the power contacts and reset the error display through PowerOn Reset of the terminal	
	Voltage briefly too high due to external influences, such as switching contactors off.	Use an R/C or diode-based protective circuit on the actuators	
	EMC faults	Take suitable EMC measures	
	Internal defect	Replace terminal	
Diag 2 LED	Terminal temperature too low	Comply with the specified temperature range	
Flash code 7	EMC faults	Take suitable EMC measures	
	Internal defect	Replace terminal	
Diag 2 LED	Terminal temperature too high	Comply with the specified temperature range	
Flash code 8	EMC faults	Take suitable EMC measures	
	Internal defect	Replace terminal	
Diag 2 LED Flash code 9	Terminal temperature difference too large:	Replace terminal	
	one of the 3 internal measuring points is faulty		
	Terminal temperature difference too large:	Check the installation position of the terminal and modify it according to the specifications in section	
	An internal measuring point shows an elevated temperature due to inadequate convection.	Mechanical installation, if required	
	EMC faults	Take suitable EMC measures	
	Internal defect	Replace terminal	

14.6.16.13 Certificates

The AKT2G-SDO-004-000 has been tested to the following directives and standards.

- 2006/42/EC
- EN 61508-1:2010 (up to SIL 3)
- EN 61508-2:2010 (up to SIL 3)
- EN 61508-3:2010 (up to SIL 3)
- EN 62061:2005/A2:2015 (up to SILCL 3)
- EN ISO 13849-1:2015 (Cat. 4, PL e)
- EN 81-20:2014
- EN 81-22:2014
- EN 81-50:2014
- EN 13243:2015

Note on approval accroding to EN 81-20, EN 81-22, and EN 81-50

• The Safety components may only be used in machines that have been designed and installed in accordance with the requirements of the EN 60204-1 standard.

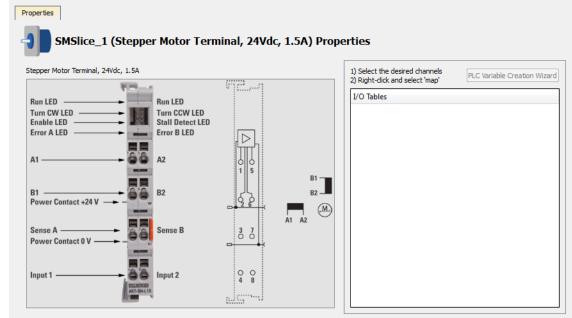
- Provide a surge filter for the supply voltage of the Safety components against overvoltages. (Reduction to overvoltage category II)
- EN 81 requires that in the case of devices with internal temperature monitoring, a stop must be reached in the event of an overtemperature. In this case, passengers must be able to disembark (see EN 81-20 chapter 5.10.4.3, for example). To ensure this, application measures are necessary. The internal terminal temperature of the Safety components can be read out by the user. There is a direct switch-off at the maximum permissible temperature of the respective Safety component (see Installation position and minimum distances).
- The user must select a temperature threshold below the maximum temperature such that a stop can be reached in all cases before the maximum temperature is reached. Information on the optimum terminal configuration can be found under Notes on the arrangement of Safety components and under Example configuration for temperature measurement.
- For the use of the Safety components according to EN 81-22 and EN 81-50, the conditions described in the manuals for achieving category 4 according to EN ISO 13849-1:2015 must be observed.
- The use of Safety components is limited to indoor applications.
- Basic protection against direct contact must be provided, either by fulfilling protection class IP2X or by installing the Safety components in a control cabinet which corresponds at least to protection class IP54 according to EN 60529.
- The ambient conditions regarding temperature, humidity, heat dissipation, EMC and vibrations, as specified in the operating instructions under technical data, must be observed.
- The operating conditions in potentially explosive atmospheres (ATEX) are specified in the operating instructions.
- The safe state (triggering) of the application must be the de-energized state. The safe state of the Safety components is always the de-energized, switched-off state, and this cannot be changed.
- The service life specified in the operating instructions must be observed.
- If the Safety component is operated outside the permissible temperature range, it changes to "Global Shutdown" state.
- The Safety components must be installed in a control cabinet with protection class IP54 according to EN 60529, so that the requirement for contamination level 3 according to EN 60664-1 can be reduced to level 2.
- The Safety components must be supplied by a SELV/PELV power supply unit with a maximum voltage of Umax ≤ 36 VDC.

14.6.17 About AKT-SM-Lxx Stepper Slices

The KAS provides basic support for Kollmorgen stepper slices (AKT-SM-L15-000 & AKT-SM-L50-000). This includes the ability to scan, discover, and initialize the EtherCAT network. Configuration and motion is commanded via the application code or UDFBs.

The stepper slice is visible in the project tree and its properties may be viewed but there are several limitations.

- Axes may not be mapped to the slice
- PLC variables may not be mapped to the PDO objects.



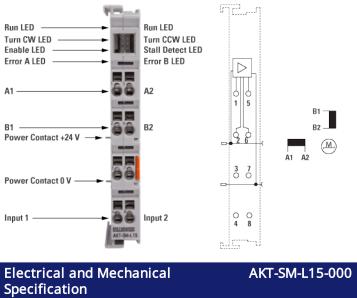
• The Properties view does not support mapping I/O to the AKT-SM-Lxx.

14.6.18 AKT-SM-L15-000 | Stepper motor terminal, 24 V DC, 1.5 A

14.6.18.1 General Specification

The Bus Terminal is intended for the direct connection of different small stepper motors. The slimline PWM output stages for two motor coils are located in the Bus Terminal together with two inputs for limit switches. The terminal can be adjusted to the motor and the application by changing just a few parameters. 64-fold micro-stepping ensures particularly quiet and precise motor operation. In many applications, integrated zero-speed monitoring makes an encoder system or limit switch unnecessary.

14.6.18.2 Electrical and Mechanical Specification



Electrical and Mechanical Specification	AKT-SM-L15-000
Number of outputs	1 Stepper Motor, 2 phases
Number of inputs	2

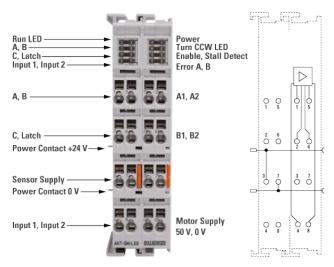
Electrical and Mechanical Specification	AKT-SM-L15-000
Output current	2 x 1 A, 2 x 1.5 A peak current, overload- and short-circuit- proof
Maximum step frequency	125,000 steps/s
Step pattern	full step, half step, up to 64-fold micro stepping
Current controller frequency	approx. 25 kHz
Diagnostics LED	error phase A and B, loss of step/stagnation, power, enable
Resolution	approx. 5,000 positions in typ. applications (per revolution)
Power supply	824 V DC (for output stage over power contacts)
Electrical isolation	500 V _{rms} (Standard Bus/signal voltage)
Current consump. from Standard Bus	typ. 60 mA
Bit width in the process image	input: n x 2 x 16 bit data, 2 x 16 bit control/status
Weight	50 g
Operating/storage temperature	0+55 °C/-25+85 °C
Relative humidity	95 %, no condensation
Vibration / shock resistance	conforms to EN 60068-2-6/EN 60068-2-27/29
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protect. class / installation pos.	IP 20/variable
Pluggable wiring	For all Bus terminals

14.6.19 AKT-SM-L50-000 | Stepper motor terminal, 50 V DC, 5 A, with incremental encoder

14.6.19.1 General Specification

The Bus Terminal is intended for stepper motors with medium performance range. The PWM output stages cover a wide range of voltages and currents. Together with two inputs for limit switches, they are located in the Bus Terminal. The stepper motor terminal can be adjusted to the motor and the application by changing just a few parameters. 64-fold micro stepping ensures particularly quiet and precise motor operation. Together with a stepper motor, the AKT-SM-L50-000 represents an inexpensive small servo axis.





870

Electrical and Mechanical Specification	AKT-SM-L50-000
Number of outputs	1 Stepper Motor, 2 phases
Number of inputs	2 limit position, 4 for an encoder system
Supply voltage	850 V DC
Output current	2 x 3.5 A, 2 x 5 A peak current
Maximum step frequency	125,000 steps/s
Step pattern	full step, half step, up to 64-fold micro stepping
Current controller frequency	approx. 25 kHz
Diagnostics LED	error phase A and B, loss of step/stagnation, power, enable
Resolution	approx. 5,000 positions in typ. applications (per revolution)
Power supply	via the Standard Bus
Current consump. from Standard Bus	typ. 100 mA
Electrical isolation	500 V _{rms} (Standard Bus/signal voltage)
Encoder signal	524 V, 5 mA single-ended
Pulse frequency	max. 400,000 increments/s (with 4-fold evaluation)
Bit width in the process image	input/output: 2 x 16 bit data + 1 x 8 bit control/status
Weight approx.	100 g
Operating/storage temperature	0+55 °C/-25+85 °C
Relative humidity	95 %, no condensation
Vibration / shock resistance	conforms to EN 60068-2-6/EN 60068-2-27/29
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protect. class / installation pos.	IP 20/variable
Pluggable wiring	For all Bus terminals

14.7 Drives

This section details the following drives:

AKD part number	Description
AKD-B00106	120/240 VAC 1.5A Drive
AKD-B00306	120/240 VAC 3A Drive
AKD-B00606	120/240 VAC 6A Drive
AKD-B01206	120/240 VAC 12A Drive
AKD-B02406	120/240 VAC 24A Drive
AKD-B04806	120/240 VAC 48A Drive
AKD-B00107	240/480 VAC 1.5A Drive
AKD-B00307	240/480 VAC 3A Drive
AKD-B00607	240/480 VAC 6A Drive
AKD-B01207	240/480 VAC 12A Drive
AKD-B02407	240/480 VAC 24A Drive
AKD-N00307	3A Distributed Servo Drive
AKD-N00607	6A Distributed Servo Drive

Table 9-5: List of AKD Drives

Related Documents

For further information on drives, refer to the following manuals:

Drives Guide	Description
AKD PDMM User Manual	Describes software installation, setup, and operation for the AKD PDMM drive. Includes basic topics and examples to help guide you in setting up and using the various features in the drive.
AKD PDMM Fault Card	Describes AKD PDMM (including AKD) faults, warnings, error messages, and alarms. Provides cause and remedy instructions to help determine the specifics of the failure and to correct the underlying problem.
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.
AKD2G Installation Manual	Describes the AKD2G series of digital drives and includes mechanical, electrical, software and functional safety options.
AKD, AKD2G, S700 (in NA) Accessories Manual	This manual describes the accessories for Kollmorgen digital drive systems and servo drive motors.
AKD EtherCAT Manual	Describes the installation, setup, range of functions, and software protocol for the EtherCAT AKD product series,
AKD2G EtherCAT Manual	AKD2G EtherCAT and CANopen Communications Manual describes the installation, setup, range of functions, and software protocol for the AKD2G product series.
S300 Reference Documentation	Kollmorgen website that gives access to all S300 manuals.
S700 Reference Documentation	Kollmorgen website that gives access to all S700 manuals.

Table 9-6: List of Drive Manuals

NOTE

The AKD manuals are located under: C:\Program Files\Kollmorgen\AKD WorkBench 1.0.x.y\WebHelp (x.y must be replaced with the version number) (this location differs if you chose another location when installing AKD).

15 Troubleshooting

In addition to the topics contained in this section, the following topics may prove useful in troubleshooting.

- Error Management this topic provides a programmatic procedure for when a non-fatal error occurs in the machine and motion needs to be halted quickly.
- Practical Application: Using Trace Time To Measure CPU Load this topic provides a procedure for calculating CPU usage.

15.1 FAQs

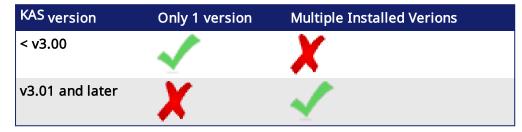
15.1.1 Basic

Why does the Installer not Start when I insert the CD?

Your Autorun feature may be deactivated. Open an Explorer window to see the autorun.exe file and 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.



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. With the release of v3.02 multiple instances of the KAS IDE are permitted. That includes running one older (<v3.02) instance and any number of newer instances. This includes multiple instances of the same version. The only limitation is available RAM.



★ TIP

The KAS IDE supports simultaneously running multiple instances. Double-clicking a shortcut or project file, or selecting the application from the Start menu will open another instance, Double-clicking on a project file more than once will open 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?

Right-clicking on a project file allows you to choose which of your installed versions of the KAS IDE to use for that instance.

	1			
		Launch with KAS IDE		_
AKD2G -axis		Open with KAS IDE version $\qquad\qquad>$	3.00.0.74983	
	2	Edit with Notepad++	3.01.0.75514	
ļ		Scan with Windows Defender		

How do I choose the default version of the KAS IDE to use?

The default version of the KAS IDE is that which is most recently installed. This can be changed to a preferred version by changing an Environment Variable.

From the list of System variables, edit KAS_INSTALL_DIR. Enter the path to your preferred version.

System Variable	
able name: KAS_INSTALL_	DIR
able value: C:\Program Fi	iles (x86)\Kollmorgen\Kollmorgen Automation Suite 3.01.0.75514
Browse Directory Brows	se File OK Cancel
	New Edit Delete
System variables	
System variables Variable	Value
-	Value ^ Destination=file
Variable	
Variable asl.log	Destination=file C:\WINDOWS\system32\cmd.exe
Variable asl.log ComSpec	Destination=file C:\WINDOWS\system32\cmd.exe
Variable asl.log ComSpec devmgr_show_nonpresent_d	Destination=file C:\WINDOWS\system32\cmd.exe . 1
Variable asl.log ComSpec devmgr_show_nonpresent_d KAS_INSTALL_DIR	Destination=file C:\WINDOWS\system32\cmd.exe . 1 C:\Program Files (x86)\Kollmorgen\Kollmorgen Automation Suite 3
Variable asl.log ComSpec devmgr_show_nonpresent_d KAS_INSTALL_DIR NUMBER_OF_PROCESSORS	Destination=file C:\WINDOWS\system32\cmd.exe . 1 C:\Program Files (x86)\Kollmorgen\Kollmorgen Automation Suite 3 4
Variable asl.log ComSpec devmgr_show_nonpresent_d KAS_INSTALL_DIR NUMBER_OF_PROCESSORS OS	Destination=file C:\WINDOWS\system32\cmd.exe . 1 C:\Program Files (x86)\Kollmorgen\Kollmorgen Automation Suite 3 4 Windows_NT

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, such as FBD blocks.

The KAS IDE will attempt to recreate what you have copied, but there are some limitations. For example, it may not automatically associate variables with a function. For example, if you copy and paste only a function block from an FFLD program, the assigned variables will not be copied and pasted, but if you select the entire network the KAS IDE will automatically create the variables for you.

Axis IFaultMonitor DK ×	Constant Two State Vende Durk Image: State Type Image: State Ty	=
Copying a block	Copying an entire network recreates assigned variables in the Dictionary	

NOTE

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:

- Close your current project without saving
- 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
- Delete all the remaining files and folders
- You can now create your new project

Where can I get the latest User Manuals?

Everything in our manuals may be found in the help. If you need a printable document or want stand-alone manuals, the documentation is available as PDFs on KDN. https://www.kollmorgen.com/en-us/developer-network/kas-downloads/

Learning Kollmorgen Automation Suite

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.

Iser Options	×
AKD GUI Scope Settings AKD GUI Scope Settings when scope page is opened ACD Drive parameter will be loaded with the default settings.	
AutoSave Recovery Files Settings AutoSave Recovery Files Period (min): 5 🔄 🖉 Enabled	Apply Cancel

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 will be prompted to recover a found file. When there is more than one file the recovery process will start with the latest date from the newest KAS IDE version. 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 will be prompted to recover the next oldest file. Choosing to not recover the file will delete it, and you will be prompted to open the next oldest file until none are left.

15.1.2 Setup

How can I download new Firmware to my AKD Drive?

How are fieldbuses connected to the KAS Runtime?

As depicted in figures found in Different Implementations, 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?

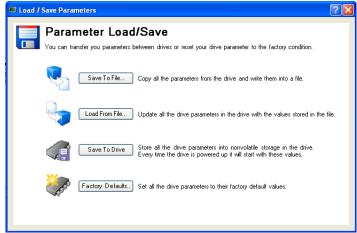
As described in Communication and Fieldbus, KAS can communicate with outside world through Ethernet, CANopen, DeviceNet.

Why does Online Configuration Mode not work after I reload the drive's factory default parameters?

Description

This issue occurs when you perform the following

- Connect to the controller and download your application
- In the project explorer, open the EtherCAT properties
- Click theOnline Configuration Mode
- In the project explorer, right-click on the AKD_1 and select Load/Save Parameter...
- Then select the Factory Defaults... command to reset the drive to its default parameters



Clicking the Online Configuration Mode leads to the following 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 (for more details, ENI File Tab)

How can I fix security issues?

If you encounter any security issues during execution of Kollmorgen Automation Suite, refer to your IT department to set your proxy properly.

Firewall

You may have to define your firewall settings to allow accessing the IP addresses used by KAS (for instance, IP address of the target system, or localhost IP address for the KAS Runtime Simulator: 127.0.0.1).

Port numbers

Port numbers have to be set properly in your firewall settings to avoid any trouble during communication, such as 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.

15.1.3 Computer / Controller

How many axes can the KAS IDE manage in 1 ms?

This number is mainly dependent upon the application and your PC's computing power. An average number would be 20 axes/ms

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 the following CPU loads:

- CycleJitter (microseconds)
- Motion execution time (microseconds)
- PLC execution time (microseconds)
- Real Time Margin (microseconds)

To view the load, do as follows:

- Open the Softscope
- Plug four probes to any kind of data (see procedure here)
- In the Control Panel, click the TraceTimes button

I cannot log into the controller's webpage. What do I do?

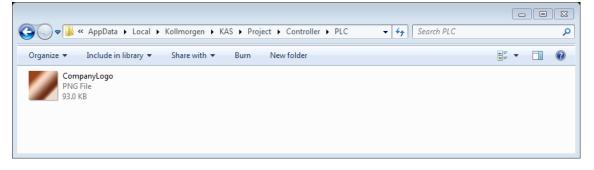
If you are unable to log into the controller's webpage using a valid user ID and password, the controller's file system may be corrupt. To recover the system you will need to use the push buttons on the controller (see Booting from the Recovery Image) to recover the system.

How can I check the if there is enough NVRAM space to store my Retain Variables?

For explanation, How can I check the NVRAM space is enough to store my retain variables?

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 IDEwill attempt 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.



15.1.4 Performance

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. Press the Refresh Tree button in KAS IDE WorkBench view.

General Properties	I/O	Configuration	PDO			
200	0	Disable & Clear F	aults			
no_name (Online)						

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?

The status bar may display a warning if communication between the KAS IDE, controller, and drive is too slow.

(1) Inactive SW HW CS STO 🚽	<
1. Communcation with drive is too slow. Check Local log for details.	<
A.AA.AAAAA	_

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.

Information and Logs								
Lo	cal I	log	Controller log	Find and R	eplace E	Browse Variabl	e Breakpoints Compiler Output	
Filt	er			*				
			Time		Source	Level	Message	
3	4		4/8/2020 11:33:46	AM (593)	IDE	WARNING	Drive #0: communication too slow for IDE-Workbench. Avg response time is 160ms.	
4	Δ	<u> </u>	4/8/2020 11:34:01	AM (786)	IDE	WARNING	Drive #0: communication too slow for IDE-Workbench. Avg response time is 157ms.	
5	Δ	4	4/8/2020 11:34:32	AM (073)	IDE	WARNING	Drive #0: communication too slow for IDE-Workbench. Avg response time is 162ms.	
4.4	i.	_~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	A	~~~~~	w	han all and a second and a second	

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.

_												_			
🔊 Resource	Monitor												-	- 🗆	\times
File Monitor	r Help														
Overview Cl	PU Memory	Disk	Network												
🗹 Image	~	PID		Send (B/sec)	Receive (B/se	t) Total (B,	/sec)				^			View	s 🔽 ^
kas_ide.exe	e	13936		1,201	5,069	9 6,	,269				¥	Netv	vork	1	Mbps -
Network Ac	tivity		927 Kbps	Network I/O		0 %	Network U	tilization	1	(•	-4	vila-Ad	t Al Al	h _A zt,
Filtered by kas	s_ide.exe														
Image		PID	Address	Send (B/	sec) Receive (E	3/sec) Tota	l (B/sec)								
kas_ide.exe		13936	5 192.168.0.102	1,2	201 5	i,069	6,269								~~
													econds		0 -
TCP Connec	ctions											TCP	Connectio	ns	100 -
Filtered by kas	s_ide.exe														
Image	Local Address	Local Por	t Remote Addr	. Remote Port	Packet Loss (%)	Latency (ms)	Send (B/	sec)	Receive (B/sec)	Total (B/sec)	^				
kas_ide.exe	192.168.0.105	50058	192.168.0.102	9900	0	116		585	4,652	5,337					
kas_ide.exe	192.168.0.105	50332	192.168.0.102	4003	0	123		183	326	509					
kas_ide.exe	192.168.0.105	50055	192.168.0.102	9900	0	109		52	15	67	~			. L. I. d	0 - 0
Listening Po	orts	1			· · · · ·	~~~~~	<u></u>	~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		2				1000/

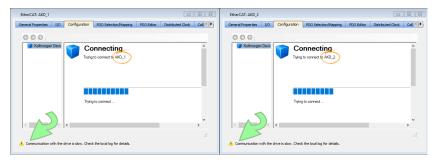
Communication delays between the Controller and Drive can occur while an application is running on the controller. Degraded drive communication can be caused by excessive ASCII SDO commands from the application code.

• Check your application code for calls to DriveParamRead/Write(...), DriveParamStrRead(...), or ECATRead/WriteSdo(...), and minimize the frequency of these calls.

• Use Online Configuration Mode with KAS IDE WorkBench to avoid ASCII SDO channel competition between your application and KAS IDE WorkBench

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.



To resolve this situation:

- 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:

- Open one WorkBench window
- · Wait for communication and initialization to complete successfully
- Open each additional WorkBench window one at a time, allowing enough time for each to successfully complete initialization.

15.1.5 EtherCAT

Is EtherCAT limited to Master/Slave applications?

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 in two ways:

- 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, too.

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 will be limited by the maximum EtherCAT frame size, which is 1500 bytes. Please see the Frame Size in the EtherCAT Master Settings Tab for more details.

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. So the EtherCAT master uses a polling mode to ensure the SDO command is

880

completed. Note that in operational mode, this polling is performed every 50 cycles 1. As a consequence, you can expect the acknowledgement of your SDO command usually before less than 100 ms. So, a good practice is to set the update rate for SDO communication in your PLC application each 25 cycles.

EtherCAT Motion Bus Concepts for more details.

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 upon 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 the following 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 \mus: 17 > 14
```

There are three possible solutions.

- Increase the Cycle Time see EtherCAT Master Settings Tab
- Deselect unused PDOs see PDO Selection/Mapping Tab
- Delete unused objects see Delete a CoE Object

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 approx. 10 milliseconds. KAS Simulator cannot execute the PLC programs faster because Windows is not able to handle timing less than 10ms.

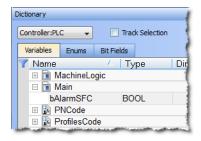
15.1.6 Programming

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, you have to:

• In the Dictionary, declare a Boolean PLC variable linked to the related SFC program



• Add the instruction in the Actions tab related to the SFC step, with first parameter set to **A** (for Alarm) as shown below

¹To avoid overloading the controller, this rate is set according to the communication load, as well as the duration the AKD takes to process commands

					Main*
First Level	Actions	P1	N	PO	Notes
bAlarr	nSFC (A,	t#3:	∍);)	

Figure 10-1: 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?

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.

For more details, refer to Fast Inputs with Pipe Network Motion

How do I implement feedback?

There are two kinds of feedback:

15.1.6.0.0.1 Primary feedback

With a S300 drive you can use a resolver for primary feedback.

15.1.6.0.0.2 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. To configure the block use the MLSmpConPNAxis, MLSmpConPLCAxis, or MLSmpConECAT functions.

How do I implement Torque Feed-forward?

Current drives that support torque feed-forward are: S300 and AKD drives.

To use torque feed-forward, you have to rely on a CONVERTER Pipe Network block. To configure the block use the MLCNVConnectEx function. 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)

For more details, refer to the three following links:

- Torque Feed-forward
- Guidelines for Choosing feed-forward Control in Industrial Applications

3A

- 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

For example, with an AKD where:

Rated current

IL.KBUSFF1.0Additive 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?

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?

This item is explained here

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 will make capturing a correct pipe position difficult. To remove the delays, you should use a sampler block to capture the actual position of an axis. The following shows how this works.



The sampler block above 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, you need to add the difference between the axis command position and the master pipe value.

A hardware limit flag is being thrown by a PipeNetwork axis at the start of my program.

If a hardware limit flag is being triggered at the program startup by a PipeNetwork 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.

15.2 Compiler Errors

The following errors may be seen when compiling a project.

"EtherCAT: ERROR: Failed to retrieve Vendor...", or "EtherCAT: ERROR: Failed to retrieve EtherCAT device..."

These errors indicate either the ESI file for the EtherCAT device is missing or 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
```

```
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
```

To correct the error, identify which EtherCAT devices or modules are listed as "Unknown" in the Project View and import an appropriate ESI file. For more details, see: Unknown – Missing ESI File.

"EtherCAT: ERROR: In AKD_1, PDO object index 0x6040, subIndex 0 is redundant", or "Failed to export ENI file, check the logs for more details.", or "EtherCAT: Warning : In AKD_1, PDO object index 0x3470, subIndex 4 is redundant."

These errors are indicative of redundant PDO entries, see "Redundant PDO Entries" on page 346.

Compile Optimized Code FAILED

The following compiler output indicates a failure to compile the optimized ("C") code:

Please send the following items to Kollmorgen technical support to help us resolve it with you.

- 1. the compilation log (copied from the compiler output)
- 2. the files in the following directory (the path can be copied and pasted into the Windows Explorer address bar):

%LOCALAPPDATA%\Kollmorgen\KAS\Project\Controller\PLC\CC\

Kollmorgen technical support contact information can be found in Global Support Contacts, or at Kollmorgen.com. Please call or e-mail the appropriate support for your region.

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. The problem can be resolved by modifying the file permissions to allow writing to the file. Below are examples of error messages for specific files.

```
Controller:PLC:> ------ Compile Optimized Code -----
Controller:PLC:> ERROR: cannot remove old output file
"C:\Users\my.name\AppData\Local\Kollmorgen\KAS\Project\Controller\PLC\C
C\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 ------
```

Figure 10-2: The file cclog.txt is not writeable.

```
Controller:----- Compile PLC -----
...
Controller:PLC:main
Controller:PLC:c:\users\my.name\appdata\local\kollmorgen\kas\project\co
ntroller\plc\cc\main.c:
Cant write output file
Controller:PLC:appli
Controller:PLC:Error(s) detected
Controller:----- PLC failed -----
```

Figure 10-3: The file main.c is not writeable.

15.3 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 205). This will decrease the PLCProgExecTime.

15.4 EtherCAT Diagnostics & Errors

15.4.1 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:

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 will 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 CRC (Cyclic Redundancy Check) 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.

- **Application Started:** When the application is started, the KAS controller resets the Link Lost Counter, Rx Error Counter, Rx Invalid Frame Counter, Forwarded Rx Error Counter, and EtherCAT Processing Unit Error Counter registers of all the nodes.
- **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 alarm.
- **Application Stopped:**When the application is stopped, the controller will read 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 will have non-zero values when a cable disconnection or an EtherCAT frame corruption occurs, The following example indicates that 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
```

15.4.2 EtherCAT Error Messages

886

This section covers the following error messages linked to the EtherCAT motion bus that are displayed in the Information and Logs window:

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.

These messages can arise due to the following causes seen in the following table.

Error	Case Description	Results
Wrong / Missing Device	 Case Description 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.
Link Loss / Device Fault	This kind of error can appear anytime in the EtherCAT communication, typically when a cable is disconnected or 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. 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. In addition, all still-reachable axes have to be 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).
Frame Loss	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.

Error	Case Description	Results
Working Counter	When EtherCAT master (KAS controller) sends out an EtherCAT datagram 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.	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 error and shuts down the EtherCAT communication.
	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. 	

15.4.2.1 Other Messages Linked to EtherCAT

For more information see "Communication and Fieldbus" on page 47

The following 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>

15.4.3 EtherCAT Communication Diagnosis Steps

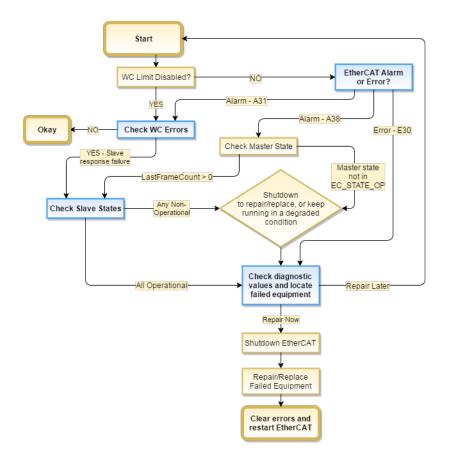
This topic covers the steps to diagnose EtherCAT communication errors, and provides examples for how an application program can help a technician detect the issue and repair it.

15.4.3.1 Diagnosing EtherCAT Communication Errors

The flow chart below explains the steps involved in diagnosing the EtherCAT errors.

🗙 TIP

The blue fields in the flow chart are clickable. Each points to sample code that can be used in a program to help with diagnosis.



15.4.3.2 Code Examples for Diagnosing EtherCAT Communication Errors

Checking 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 then the return value will be greater than 0.

```
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;
```

Checking for existing EtherCAT Alarms and Errors:

The GetCtrlErrors function can be 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.
```

Checking the Device (slave) States:

The device (slave) state can be read by the application using the function block ECATDeviceStatus.

```
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
END_IF;
```

Checking the Connections for Errors:

This sample code uses **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.

15.5 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 EtherCAT Coupler Technical Manual.

15.5.1 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. See below for more information.

The ETHERCAT Bus Coupler has respectively a green and yellow LED at the RJ45 plug sockets, which indicate the state of the fieldbus (Figure 4.1). The RUN and ERROR LEDs (upper middle) indicate the state of the EtherCAT State Machine.

On the upper right hand side of the Bus Couplers are two more green LEDs that indicate the supply voltage. The left hand LED indicates the presence of the 24 V supply for the Bus Coupler. The right hand LED indicates the presence of the supply to the power contacts.

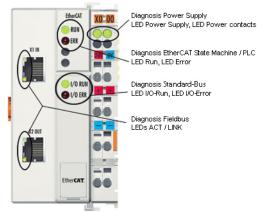


Figure 10-4: ETHERCAT Coupler Diagnostic LEDs

15.5.2 EtherCAT LED Power Supply Diagnosis

LED		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

15.5.3 EtherCAT LED Off Power Supply Diagnosis

LEDs	
Left Green LED Off	Bus coupler has no power
Right Green LED Off	No 24 VDC power connected to the power contacts

15.5.4 LEDs for EtherCAT State Machine/PLC Diagnosis

LEDs		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

15.5.5 LEDs for EtherCAT Connection Diagnosis

LEDs		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

15.5.6 LEDs for EtherCAT Data Diagnosis

LEDs			Display	Sta	itus	Description
I/O-Run	Green		Off			EtherCAT Bus inactive
			On	_		EtherCAT Bus active
LED Red; I/O Error	Error Code Argument	Desc	ription	Reme	dy	
Persistent, continuous blinking		EMC	problems	pea • Imp • If a by a	iks blement EN EtherCAT	supply for overvoltage or under voltage //C measures Bus error is present, it can be localized f the coupler (by switching it off and)

892

LED Red; I/O Error	Error Code Argument	Description	Remedy
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; halve the number of Bus Terminals attached and check whether the error is still present with the remaining Bus Terminals. 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, and if the error occurs again then 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
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

15.6 Troubleshooting 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 (the PCMM or AKD PDMM).

Are there any errors?

The first thing to do is to check the ErrorID output of FSoEParamsInit. There are some common failure cases that can be identified from the ErrorID values and the Controller log messages.

ECERR_ DEVICE_	The Safety parameters can only be sent after the _{EtherCAT} network state is in op- mode. To verify _{EtherCAT} is ready:
NOTREADY	 Verify the motion engine is running. Use <u>MLMotionStatus</u> to verify the status is MLSTATUS_RUNNING before calling FSoEParamsInit. Check the The <u>EtherCAT</u> state directly. Use <u>ECATMasterStatus</u> to verify the state is EC_STATE_OP.
ECERR_ DEVICE_	This error is generated if the argument specified for <mark>FSoEParamsInit</mark> 's FSoEMasterAddress is not a FSOE Master.
FSOE_ MASTER_ INVALID	 Check to ensure that the EtherCAT address specified for the FSoE master is correct. Check the supported FSoE Masters list.
ECERR_	This can occur if the FSoE Master is not responding or is not configured properly.
DEVICE_ TIMEOUT	 Check the FSoE Master diagnostic LEDs, Error Codes/Messages, log messages, and any other diagnostic tools available from the FSoE Master vendor. For 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.

- 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 will be 6 Normal Operation and FSoE Sate will be 54 Process Data.



• If the CRC does not match the Safety State will be 3 -Configuration required.



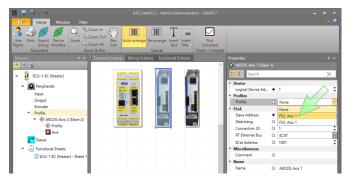
AKD2G module(s) are incorrectly set to use "Workbench Parameters"

The AKD2G has two options for configuring safety parameters which are determined by selecting the modules associated with the AKD2G in the KAS IDE Project Tree.

If the module(s) are set for "Workbench Parameters" then the FSoE Master cannot set the AKD2G's safety parameters. The "WorkBench Parameter" modules are set by default when the AKD2G is scanned and discovered. The modules must be changed to accept the parameter from the Safety Maser if you want to use FSoEParamsInit. The picture below shows the appropriate modules for using FSoEParamsInit with AKD2G.

> Subprograms		🗢 Eti	herCAT N	Module Selector		×
Motion		Select	a module	to add:		
> 🔬 PLCopen	PLCopen	E	AKD2	!G		
Controller Onboard I/O				Name	Туре	Ident
🛥 EtherCAT				AKD2G Dual Avia	FS2, Axis 2 - WorkBench Parameters	0x220000(2228224)
😪 AKD2G_1	AKD2G Dual Axis		7			
> BBB Device_1	SCU			AKD2G Dual Axis	FS2, Axis 2	0x210000(2162688)
References		17	-	AKD2G Dual Axis	FS2. Axis 1 - WorkBench Parameters	0x1120000(17956864)
옯 Fieldbus			1			
Control Panel	Panel			AKD2G Dual Axis	FS2, Axis 1	0x1110000(17891328)
FaultReportPanel	Panel	11-				
KVBProject	KVB 2					
SafePLC2						
plorer Dictionary Libraries						Ok Cancel

Additionally, the AKD2G profile in the SafePLC2 project should be set to "FS*x* Axis *Y*" in order to send the parameters to the AKD2G.



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 will only initialize safety parameters if the FSoE master determines there are parameters to set in the safety device(s). Please check the FSoE master configuration using the vendor's software. For 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
- AKD2G Safety Parametrization Using FSoE with SCU-1-EC and PCMM/AKD PDMM
- FSoEParamsInit

15.7 Troubleshooting KAS IDE and SafePLC2

Following are several 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 which are missing ESI files will be identified in the KAS IDE's EtherCAT **Devices** tab.

Devic	es Master E	NI File ESI Files F	SoE / PDO Connections	
	_			
	EtherC	AT Devices		
-				
-				
_ Sh	ow only drives			
#	Name	Vendor	Product	Мо
1	AKD2G_1	Kollmorgen	AKD2G Dual Axis FS2 EtherCAT Drive (CoE)	AKE
2	Device 1	BBH Products GmbH	scu	N/A
	Coupler 1	Kollmorgen	AKT2G-ECT-000-000 EtherCAT Gateway	N//
Ļ	DI_Slice_1	Kollmorgen	AKT2G-DN-008-000 8Ch. Dig. Input 24V, 3ms	N//
5	DI_Slice_2	Kollmorgen	AKT2G-DNH-008-000 8Ch. Dig. Input 24V, 10µs	N//
5	DO_Slice_1	Kollmorgen	AKT2G-DT-008-000 8-channel digital output terminal 24 V DC, 0.5 A	N//
7	Al_Slice_1	Kollmorgen	AKT2G-AN-430-000 4Ch. Ana. Input +/-10V Diff., +/-20mA SingleEnded, 16 Bit	N/A
8	AO_Slice_1	Kollmorgen	AKT2G-AT-410-000 4Ch. Ana. Output 0-10V, 12bit	N/A
)	DO_SafeSlice_1	Kollmorgen	AKT2G-SDO-004-000, 4 Ch. Safety Output 24V, 0.5A	N//
0	DL SafeSlice 1	Kollmorgen	AICT2C-SDL004-000_4 Ch. Safety Input 24V	N/A
1<	Device_2	0	Missing ESI for 0x7703052	N/A
2	Device_3	0	Missing Editor 0x0583052	N/A
			-	
< 🗌				

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, then the Local log messages will indicate a problem with device creation. For 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

To correct the problem: Use the KAS IDE's EtherCAT-> **ESI Files** tab to add the ESI files for the missing devices.

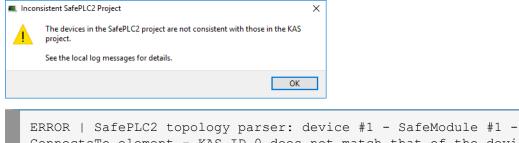
To avoid the problem: Add all Safety devices from the KAS IDE, including their ESI files, and then launch SafePLC2 from the KAS IDE project tree. SafePLC2 will automatically read the Safety device topology from the KAS IDE.

Topology mismatch with the imported SafePLC2 project

If a SafePLC2 project file is imported into a KAS project, the KAS IDE will check the incoming Safety project's EtherCAT topology. Importing the project will fail If there is a topology mismatch and the Local logs will identify the mismatched nodes.

For example:

896



```
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"
```

Another example:

```
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 will be retained if importing the SafePLC2 project fails.

To correct the problem: Import the SafePLC2 project to the matching .kas project that was originally exported to SafePLC2.

Duplicate FSoE Address

When compiling, the KAS IDE checks to ensure the FSoE address is unique. A compiler warning message is generated if the address is not unique. For example:

```
EtherCAT: Warning: FSoE address 101 is duplicated for the following devices: AKD2G_1, DI_SafeSlice_1, DO_SafeSlice_1
```

To correct the problem:

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 Working with SafePLC2 Projects.

FSoE / PDO Connections do not match the EtherCAT topology.

The compiler will generate 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. For example:

```
< compiler messages go here! >
```

To correct the problem: Use the KAS IDE's EtherCAT-> **Scan Devices** to discover and associate the physical Safety devices that are expected by the SafePLC2 project.

To avoid the problem: Scan and discover all physical EtherCAT Safety devices from the KAS IDE before launching SafePLC2 from the KAS IDE Project tree. SafePLC2 will automatically read the Safety device topology from the KAS IDE.

15.8 Connect Remotely

You can connect to a AKD PDMM or PCMM from an external network using VPN or other tunneling protocol. To do so you must open the following ports which are used by KAS to connect to a controller.

 KAS IDE: Controller Web Server; Download PLC application; Runtime front end communication. This is necessary for the connection to the runtime. HMI and other Modbus devices to communicate with Modbus TCP. Ethernet/IP communication KAS IDE: Oscilloscope variable monitoring. KAS IDE: Communication with the PLC Virtual Machine. This is necessary for the connect o the runtime. KAS IDE: Communication with the runtime engine; WorkBenchviews. This is necessary the connection to the runtime. Profinet communication 	
 2222 Ethernet/IP communication 4002 KAS IDE: Oscilloscope variable monitoring. 4003 KAS IDE: Communication with the PLC Virtual Machine. This is necessary for the connect to the runtime. 9900 KAS IDE: Communication with the runtime engine; WorkBenchviews. This is necessary the connection to the runtime. 34962 Profinet communication 	
 4002 KAS IDE: Oscilloscope variable monitoring. 4003 KAS IDE: Communication with the PLC Virtual Machine. This is necessary for the connect to the runtime. 9900 KAS IDE: Communication with the runtime engine; WorkBenchviews. This is necessary the connection to the runtime. 34962 Profinet communication 	
 4003 KAS IDE: Communication with the PLC Virtual Machine. This is necessary for the connector to the runtime. 9900 KAS IDE: Communication with the runtime engine; WorkBenchviews. This is necessary the connection to the runtime. 34962 Profinet communication 	
 to the runtime. 9900 KAS IDE: Communication with the runtime engine; WorkBenchviews. This is necessary the connection to the runtime. 34962 Profinet communication 	
the connection to the runtime. 34962 Profinet communication	ection
	y for
24064 Profinet communication	
44848 Ethernet/IP communication	

16 Annexes: Lists of Manuals, Content, and Support Information

This section provides lists of available manuals, tables, figures, how-to's, etcetera.

16.1 List of How-Tos

16.1.1 PLC Code How-Tos

- Declare an Array
- Control an SFC Child
- Draw SFC divergences
- Create SFC Parallel Branches
- Toggle a FBD Connection to make it Negative
- Change a Link in the Pipe Network
- Create a PLCopen Axis
- Read Output of a MC Function Block in ST
- Sort the Variables in the Dictionary
- Understand the Location Details in the Find and Replace window
- Start and Stop a PLC Application using HTTP Requests from the HMI

16.1.2 EtherCAT Fieldbus How-Tos

How To Create a Digitizing Axis

16.1.3 Advanced Motion How-Tos

- Use Fast Inputs with PLCOpen
- Use Fast Inputs with Pipe Network Motion
- Implement the Torque Feed-forward
- Working With A Digitizing Axis in PLCopen

16.1.4 Run the Application How-Tos

- Choose the Appropriate Level for Log Messages
- Plug a Probe in the Softscope
- Plug Motion Variables in the Softscope
- Plug PLC Variables in the Softscope
- How Do I Export the Oscilloscope Data?
- Set Breakpoints
- Activate Online Change
- Change Priority among Programs
- Specify the Duration of a Cycle

16.1.5 Hardware How-Tos

- Download a new Firmware to my AKD Drive
- Check the NVRAM space is enough to store my retain variables
- Download your Application on the HMI device (AKI)

16.2 List of KAS Manuals

Find below the available literature for the following components:

- KASIDE
- KVB

- HMI
- I/O
- AKD2G, AKD, & S300/S700
- PDMM / PCMM

16.2.1 KAS IDE

Document Title	PDF	Description
Release Notes Getting Started	PDF	The KAS version 3.05 Release Notes contain fixed limitations, known limitations, workarounds, and information on all hardware and software components that have been updated, changed or added in this release. Covers the main steps to get your KAS system up and running.
Getting Started	٢	What does it contain?
		 HW Installation (Connection and Wiring): Wiring & hardware details, connectors, system diagrams HW Configuration: Basic configuration and settings needed to start the HW components (HMI + Industrial PC + Fieldbus + I/O) SW Installation: KAS software setup
30 Minutes to Motion	PDF 📐	Covers the main topics to help you start 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. What does it contain?
		 Key Features Explore the Workspace Build a motion project Almost every task that you perform in KAS falls under one of the following basic steps (which may not always be completed in the following order):
		 Start Projects - Create a project from scratch, or modify an existing project. Add Components - Add elements to build your project necessary to control the motion part of your system. Build Output - Select a device and generate the application that you will deliver to users. Run Output - Make the output accessible to your end-users.
IDE User Manual	PDF	Contains the content to help you with KAS IDE, except the topics included in the Reference Manuals.
Reference Manual - PLC Library	PDF	Contains Technical References on PLC Programming Languages and Library.
Reference Manual - Motion Library		Contains Technical References on Motion Library for Pipe Network and PLCopen.

16.2.2 KVB

KVB Guide	Description
Kollmorgen	Quick Start that covers the most important points to install and use
Visualization Builder™	Kollmorgen Visualization Builder, in order to configure HMI Panels and PC
Quick Start Guide	operated control applications.

KVB Guide	Description
Kollmorgen Visualization Builder™ User Manual	Contains all the content to help you with Kollmorgen Visualization Builder.

16.2.3 HMI

HMI Part Number	Description	KVB	Manual
AKI2G-CDA-MOD-05T-000	Graphical Display 7" TFT LCD, Touchscreen	v2.20	Ø
AKI2G-CDA-MOD-07T-000	Graphical Display 5" TFT LCD, Touchscreen	v2.20	Ø
AKI2G-CDB-MOD-07T-000	Graphical Display 7" TFT LCD, Touchscreen	v2.20	Ø
AKI2G-CDB-MOD-12T-000	Graphical Display 12" TFT LCD, Touchscreen	v2.20	Ø
AKI-CDC-MOD-12T-000	Graphical Display 12.1" TFT LCD, Touchscreen	v2.0	Ø
AKI-CDC-MOD-15T-000	Graphical Display 15.4" TFT LCD, Touchscreen	v2.0	Ø
AKI-CDC-MOD-21T-000	Graphical Display 21.5" TFT LCD, Touchscreen	v2.0	Ø

16.2.4 I/O

AKT2G (EtherCAT) Terminals

I/O Terminal Part Number	I/O Terminal Description
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, parameterisable, -10/0+10 V, -20/0/+4+20 mA, 16 bit
AKT2G-AT-410-000	4-channel analog output terminal 010 V, 12 bit, 1-wire system
AKT2G-AT-425-000	4-channel analog output terminal -10 V+10 V, 12 bit, 4 x 2-wire system
AKT2G-DN-002-000	Up/down counter 24 V DC, 100 kHz, 32 bit counter depth
AKT2G-DN-008-000	8-channel digital input terminal 24 V DC, filter 3.0 ms, 1-wire system
AKT2G-DNH-008-000	8-channel digital input terminal 24 V DC, filter 10 μs, 1-wire system
AKT2G-DT-008-000	8-channel digital output terminal 24 V DC, 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, grey
AKT2G-ENC-180-000	1-channel incremental encoder interface, 32 bit
AKT2G-ENC-190-000	Incremental encoder interface with differential input, 16/32 bit
AKT2G-PSF-024-000	Power supply terminal with fuse, 24 V DC
AKT2G-SDI-004-000	4-channel digital input terminal, Safety, 24 V DC
AKT2G-SDO-004-000	4-channel digital output terminal, Safety, 24 V DC, 0.5 A

I/O Terminal Part Number	Description	Link	Replaced By
AKT-AN-200-000	2 Channel Thermocouple Input Module	Ø	AKT2G-AN-240-000
AKT-AN-400-000	4 Channel Thermocouple Input Module	Ø	AKT2G-AN-400-000
AKT-AN-410-000	4 Channel Analog Input Module, 0-10 VDC	Ø	AKT2G-AN-430-000
AKT-AN-420-000	4 Channel Analog Input Module, 0-20 mA	Ø	AKT2G-AN-430-000
AKT-AN-810-000	8 Channel Analog Input Module, 0-10 VDC	Ø	AKT2G-AN-430-000
AKT-AN-820-000	8 Channel Analog Input Module, 0-20 mA	Ø	AKT2G-AN-430-000
AKT-AT-220-000	2 Channel Analog Output Module, 0-20 mA	Ø	_
AKT-AT-410-000	4 Channel Analog Output Module, 0-10 VDC	Ø	_
AKT-AT-420-000	4 Channel Analog Output Module, 0-20 mA	Ø	_
AKT-AT-810-000	8 Channel Analog Output Module, 0-10 VDC	Ø	AKT2G-AT-410-000
AKT-AT-820-000	8 Channel Analog Output Module, 0-20 mA	Ø	_
AKT-DN-004-000	4 Channel Digital Input Module, 24 VDC 3ms	Ø	AKT2G-DN-008-000
AKT-DN-008-000	8 Channel Digital Input Module, 24 VDC 3ms	Ø	AKT2G-DN-008-000
AKT-DNH-004-000	4 Channel Digital Input Module, 24 VDC 0.2ms	Ø	AKT2G-DNH-008-000
AKT-DNH-008-000	8 Channel Digital Input Module, 24 VDC 0.2ms	Ø	AKT2G-DNH-008-000
AKT-DT-004-000	4 Channel Digital Output Module, 24 VDC 0.5A	Ø	AKT2G-DT-008-000
AKT-DT-008-000	8 Channel Digital Output Module, 24 VDC 0.5A	Ø	AKT2G-DT-008-000
AKT-DT-2RT-000	2 Channel Relay Output Module, 230 V AC 2.0A Rel.2NO PotFree	Ø	-
AKT-ECT-000-000	EtherCAT Bus Coupler	Ø	AKT2G-ECT-000-000
AKT-EM-000-000	Standard-Bus End Terminal	Ø	AKT2G-EM-000-000
AKT-IM-000-000	Isolation / Separation Terminal	Ø	_
AKT-PS-024-000	Power Supply, 24 VDC	Ø	AKT2G-PSF-024-000
AKT-PSF-024-000	Fused Power Supply with diagnostics, 24 VDC	Ø	AKT2G-PSF-024-000
AKT-SM-L15-000	Stepper Motor Terminal, 24 VDC, 1.5 A	Ø	_
AKT-SM-L50-000	Stepper Motor Terminal, 50 VDC, 5 A	Ø	

AKT (K-Bus) Terminals

IMPORTANT

902

24-volt power is not passed through AKT-AN-200-000 and AKT-AN-400-000 thermocouple modules. To get 24VDC power to devices that need it (such as an AKT_AT-410-000 Output module) there are two possible solutions.

- Place the module requiring 24VDC before the thermocouple module.
- Add a power feed module (AKT-PS-024-000 or AKT-PSF-024-000) after the thermocouple module.

16.2.5 AKD2G, AKD, & S300/S700

KAS IDE User Guide | 16 Annexes: Lists of Manuals, Content, and Support Information

Drives Guide	Description
AKD PDMM User Manual	Describes software installation, setup, and operation for the AKD PDMM drive. Includes basic topics and examples to help guide you in setting up and using the various features in the drive.
AKD PDMM Fault Card	Describes AKD PDMM (including AKD) faults, warnings, error messages, and alarms. Provides cause and remedy instructions to help determine the specifics of the failure and to correct the underlying problem.
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.
AKD2G Installation Manual	Describes the AKD2G series of digital drives and includes mechanical, electrical, software and functional safety options.
AKD, AKD2G, S700 (in NA) Accessories Manual	This manual describes the accessories for Kollmorgen digital drive systems and servo drive motors.
AKD EtherCAT Manual	Describes the installation, setup, range of functions, and software protocol for the EtherCAT AKD product series,
AKD2G EtherCAT Manual	AKD2G EtherCAT and CANopen Communications Manual describes the installation, setup, range of functions, and software protocol for the AKD2G product series.
S300 Reference Documentation	Kollmorgen website that gives access to all S300 manuals.
S700 Reference Documentation	Kollmorgen website that gives access to all S700 manuals.

16.2.6 PDMM / PCMM

Product Guide		Description
AKD PDMM Quick Start	Ø	Contains all information needed to safely install and setup an AKD drive
AKD and AKD PDMM Installation Manual	Ø	Covers the most important points to install the drive hardware and software Provides instructions for basic drive setup and connection to a network
PCMM Installation Guide	Ø	Covers the most installation and setup of the controller.
AKD PDMM User Guide	Ø	Contains information on AKD drive parameters, EWV (Embedded Workbench Views), and using the AKD PDMM in the KAS IDE.
AKD PDMM Fault Card	Ø	Includes technical data and dimensional drawings of accessories such as cables, brake resistors, and mains supplies

17 Acronyms & Initialisms

Term	Definition	Description
АКА	Also Known As	Provides an alias to a name
AKD	Advanced Kollmorgen Drive	Drives
AKI	Advanced Kollmorgen Interface	HMI
AKT	Advanced Kollmorgen Terminal	Remote Input/Output Terminals
ANSI	American National Standards Institute	ANSI is a private, nonprofit organization that oversees the development of voluntary consensus standards for products, services, processes, systems, and personnel
ASFB	Application Specific Function Block	Library that can be written to provide a specific application task
ASIC	Application-Specific Integrated Circuit	An ASIC is an integrated circuit (IC) customized for a particular use, rather than for general-purpose use. Modern ASICs often include entire 32-bit processors, memory blocks including ROM, RAM, EEPROM, Flash and other large building blocks
BISS	Bi-directional Serial Synchronous interface	An open-source communication protocol for feedback devices. With BiSS, all of the computation for interpolation in regard to position occurs on the ASIC directly in the encoder
САМ	Computer-Aided Manufacturing	CAM means the use of a wide range of computer-based software tools that assist engineers and CNC machinists in the manufacture or prototyping of product components
CAN	Controller Area Network	CAN is a broadcast, differential serial bus standard developed for connecting electronic control units. Each node is able to send and receive messages, but not simultaneously.
CF	Compact Flash	CF is a mass storage device format used in portable electronic devices
CIP	Common Industrial Protocol	The Common Industrial Protocol allows complete integration of control with information, multiple CIP Networks, and Internet technologies
CRC	Cyclic Redundancy Check	A CRC is a type of function that takes as input a data stream of any length and produces as output a value of a certain fixed size. The term CRC is often used to denote either the function or the function's output. A CRC can be used as a checksum to detect accidental alteration of data during transmission or storage
CSV	Comma-Separated Values	CSV file format is a file type that stores tabular data
DMA	Dynamic Memory Allocation	DMA is the allocation of memory storage for use in a computer program during the run-time of that program. It can be seen also as a way of distributing ownership of limited memory resources among many pieces of data and code

Term	Definition	Description
EDS	Electronic Data Sheet	The Electronic Data Sheet is a file format that defines the communication behavior and object dictionary for the devices following the CANopen standard CiA 306. EtherCAT ESI files amy contain references to EDS files.
ENI	EtherCAT Network Information	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.
ERP	Enterprise Resource Planning	ERP integrates (or attempts to integrate) all data and processes of an organization into a unified system
ESI	EtherCAT Slave Information	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.
FBD	Function Block Diagram	A function block diagram describes a function between input variables and output variables. A function is described as a set of elementary blocks
FFLD	Free Form Ladder Diagram	Free Form Ladder logic is a method of drawing electrical logic schematics. It is now a very popular graphical language for programming Programmable Logic Controllers (PLCs). It was originally invented to describe logic made from relays. The name is based on the observation that programs in this language resemble ladders, with two vertical "rails" and a series of horizontal "rungs" between them
FoE	File over EtherCAT	This very simple protocol, similar to TFTP, enables access to any data structure in the device. Standardized firmware upload to devices is therefore possible, irrespective of whether or not they support TCP/IP
FPGA	Field-Programmable Gate Array	FPGA is a semiconductor device that can be configured by the customer or designer after manufacturing; hence the name "field-programmable"
FSoE	FailSafe over EtherCAT	The protocol FSoE was specified for the transmission of safety relevant data. It is used to send input information of safety sensors (such as safety light curtains or emergency stop buttons) to a safety logic controller. Based on these inputs, this controller computes the commands for the safe outputs (such as contactors or safety relevant drives) and thus controls the safety functionality of the machine
GUI	Graphical User Interface	A GUI is a type of user interface which allows people to interact with a computer and computer-controlled devices

Term	Definition	Description
HMI	Human-Machine Interfaces	Also known as computer-human interfaces (CHI), and formerly known as man-machine interfaces, they are usually employed to communicate with PLCs and other computers, such as entering and monitoring temperatures or pressures for further automated control or emergency response
IC	Integrated Circuits	Miniaturized electronic circuits (consisting mainly of semiconductor devices, as well as passive components) that have been manufactured in the surface of a thin substrate of semiconductor material
IDE	Integrated Development Environment	An integrated development environment is a type of computer software that assists computer programmers in developing software. IDEs normally consist of a source code editor, a compiler and/or interpreter, build-automation tools, and a debugger
IDN	Identification Number	An IDN preceded by the prefix "P", specifies a product specific (manufacturer) IDN in short-hand notation. The actual IDN number for a product-specific IDN, can be obtained by adding 32768 to the short-hand numeric value. For convenience, the actual IDN number is given in parentheses following the short hand notation. For example, P2 is a manufacturer-specific IDN whose actual IDN number is 32770
IEC	International Electrotechnical Commission	IEC is a not-for-profit, non-governmental international standards organization that prepares and publishes International Standards for all electrical, electronic and related technologies
IEC 61131		IEC standard for Programmable logic controllers (PLCs)
IEC 61131-3		IEC 61131-3 is the third part of the open international standard IEC 61131. The current (second) edition was published in 2003. IEC 61131-3 currently defines five programming languages for programmable control systems It deals with programming languages and defines two graphical and two textual PLC programming language standards
IL	Instruction List	It is a low-level language and resembles assembly
IPC	Industrial PC	Industrial PC is the x86 PC-based computing platform for industrial applications. Industrial PC offers features different from the consumer PC on the reliability, compatibility, expansibility and long term supply. KAS IPC usually includes a touch-screen display as a combined input and output device.
IRQ	Interrupt Request	An interrupt request refers to the act of interrupting the bus lines used to signal an interrupt

Term	Definition	Description
JTAG	Joint Test Action Group	JTAG is used for accessing sub-blocks of integrated circuits, and is also useful as a mechanism for debugging embedded systems, providing a convenient "back door" into the system. When used as a debugging tool, an in-circuit emulator - which in turn uses JTAG as the transport mechanism - enables a programmer to access an on-chip debug module which is integrated into the CPU via the JTAG interface. The debug module enables the programmer to debug the software of an embedded system
KAS	Kollmorgen Automation Suite	Umbrella name for a software package including the KAS IDE and the KAS Runtime software
KAS IDE	Kollmorgen Automation Suite - Integrated Development Environment	The KAS IDE is the GUI View environment. It is a Windows integrated design environment (IDE) containing all the tools and editors (based on the different IEC 61131 languages) that users need during the entire life cycle of the machine
KAS Runtime	Kollmorgen Automation Suite - Runtime	The KAS Runtime is the engine that provides a soft PLC and a motion controller
KVB IDE	Kollmorgen HMI Development Environment	Kollmorgen Visualization Builder is an editor that allows the end-user to control the KAS Runtime
LD	Ladder Diagram	FFLD
LSB	Least Significant Bit	Sometimes abbreviated as LSB, the least significant bit is the lowest bit in a series of numbers in binary; the LSB is located at the far right of a string. For example, in the binary number: 10111001, the least significant bit is the far right "1".
MDI	Multiple Document Interface	Graphical computer applications with an MDI are those whose windows reside under a single parent window (usually with the exception of modal windows), as opposed to all windows being separate from each other (single document interface).
		Advantages:
		- With MDI, a single menu bar and/or toolbar is shared between all child windows, reducing clutter and increasing efficient use of screen space
		- An application's child windows can be hidden/shown/minimized/maximized as a whole
		- Features such as "Tile" and "Cascade" can be implemented for the child windows
ML	Motion Library	The Motion Library is the interface between the IEC61131-3 logical application and the motion engine. It gives access from IEC61131-3 to pipe and Pipe Blocks parameters and methods as well as to higher levels of functionalities such a homing, tensioning, dynamic correction, etc.
MSB	Most Significant Bit	Sometimes abbreviated as MSB, the most significant bit is the bit position in a binary number having the greatest value

Term	Definition	Description
MDP	Modular Device Profile	The EtherCAT Modular Device Profile defines the data structure organization for a device with subdivided substructures. The object dictionary structure, the PDO structure, and their corresponding indexes are defined by the MDP specification. This makes it possible for the EtherCAT master to support a variety of modular device types from different manufacturers. MDP supports devices with either physical plug-in modules or logical modules, enabling users to assemble scalable devices or select from a range of pre-built devices with various feature sets.
NAT	Network Address Translation	In computer networking, NAT is the process of modifying network address information in datagram (IP) packet headers while in transit across a traffic routing device for the purpose of remapping a given address space into another.
NIC	Network Interface Controller	A network interface controller (or card) is a hardware device that handles an interface to a computer network and allows a network-capable device to access that network
NVRAM	Non-Volatile Random Access Memory	NVRAM is the general name used to describe any type of random access memory which does not lose its information when power is turned off. This memory is in contrast to the most common forms of random access memory today, which both require continual power in order to maintain their data. NVRAM is a subgroup of the more general class of non-volatile memory types, the difference being that NVRAM devices offer random access, like hard disks. The best-known form of NVRAM memory today is flash memory
OEM	Original Equipment Manufacturer	A term that refers to containment-based re-branding, namely where one company uses a component of another company within its product, or sells the product of another company under its own brand. OEM refers to the company that originally manufactured the product
OPC	OLE for Process Control	OPC is the original name for an open standard to specify the communication of real-time plant data between control devices from different manufacturers
PCI	Peripheral Component Interconnect	The PCI specifies a computer bus for attaching peripheral devices to a computer motherboard
РСММ		Programmable controller which lets you control multiple EtherCAT slave drives and I/O. Essentially a AKD PDMM without an AKD.
PD	Programmable Drive	(Also known as Servo Amplifiers or Servo Drive) A Drive can be programmable, which means it has an open hardware and software architecture to make it ready for nearly all conceivable customer-specific modifications
PDMM	Programmable Drive Multi-axis Master	Programmable drive which lets you control multiple EtherCAT slave drives and I/O

Term	Definition	Description
PDO	Process Data Object	 PDO is a type of protocol frame used in some fieldbuses. A PDO contains one or more object dictionary entries, which define the application data transferred between devices. EtherCAT uses the same communication mechanisms (PDO and SDO) as CANopen. EtherCAT transfers the process data between the master and slave device cyclically. PDOs have several attributes to define their properties. PDOs can be exclusive, meaning that no additional PDOs may be assigned to a device if an exclusive PDO is assigned. If the PDO type is fixed (Fixed attribute = 1) then the
		PDO's content cannot be changed by users.
PID	Proportional-Integral- Derivative	 A PID controller is a generic control-loop feedback mechanism widely used in industrial control systems. An "error" occurs when an event or a disturbance triggers off a change in the process variable. A PID controller attempts to correct the error between a measured process variable and a desired setpoint by calculating and then outputting a corrective action that can adjust the process accordingly
PLC	Programmable Logic Controller	A Programmable Logic Controller, PLC, or Programmable Controller is a digital computer used for automation of industrial processes, such as control of machinery on factory assembly lines. Used to synchronize the flow of inputs from (physical) sensors and events with the flow of outputs to actuators and events
PNE	Pipe Network Engine	The Pipe Network concept is an innovative solution to solve axis synchronization problems. It is based on Pipe Blocks representing the whole mechanical system by analogy
POU	Programmable Organization Unit	An application is a list of programs. Programs are executed sequentially within the target cycle according to the order defined by the user and displayed in the Project View
Qwt	Qt Widgets	Qwt is a graphics extension to the Qt GUI application framework from Trolltech ASA
RTC	Real-Time Computing	RTC is the study of hardware and software systems which are subject to a "real-time constraint" (i.e., operational deadlines from event to system response)
RTOS	Real-Time Operating System	RTOS is a multitasking operating system intended for real- time applications
S300	Servostar 300 drive	Servo Drive
S700	Servostar 700 drive	Servo Drive
SCADA	Supervisory Control And Data Acquisition	SCADA systems are typically used to perform data collection and control at the supervisory level. Some SCADA systems only monitor without doing control, these systems are still referred to as SCADA systems

Term	Definition	Description
SDO	Service Data Object	The SDO protocol is used to read and write values across fieldbuses . The SDO data is defined by the object dictionary.
		EtherCAT uses the same communication mechanisms (PDO and SDO) as CANopen. SDO data is non-cyclic and is applicable for non-deterministic data transfers.
SFC	Sequential Function Chart	It can be used to program processes that can be split into steps. The main components of SFC are: - Steps with associated actions - Transitions with associated logic conditions - Directed links between steps and transitions
SPLC	Software version of a PLC	Usually working on PC-based hardware
ST	Structured Text	A high-level language which is block structured and syntactically resembles Pascal
TDI	Tabbed Document Interface	TDI allows multiple documents to be contained within a single window, using tabs to navigate between them
ТМР	Trapezoidal Motion Profile	This Pipe Block is a source block that frequently serves as a virtual master for a system composed of several pipes. Generally, a trapezoidal motion profile generator is used to generate a flow of values with a first derivative which produces a trapezoidal trajectory
UDFB	User-Defined Function Block	UDFB can be used as a sub-Function Block in another program of the application. It is described using FBD, LD, ST or IL language. Input/output parameters of a UDFB (as well as private variables) are declared in the variable editor as local variables of the UDFB
UDP	User Datagram Protocol	UDP is a network protocol used for the Internet. This protocol assumes that the Internet Protocol (IP) is used as the underlying protocol. This protocol provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism. The protocol is transaction oriented, and delivery and duplicate protection are not guaranteed.
USB	Universal Serial Bus	USB is a serial bus standard to interface devices
UTF8	Unicode Transformation Format (8-bit)	UTF-8 is a variable-length character encoding for Unicode. It is able to represent any character in the Unicode standard, yet the initial encoding of byte codes and character assignments for UTF-8 is backward-compatible with ASCII
UU	User Units	A coordinate value or length expressed in user units represents a coordinate value or length in the current user coordinate system. Thus, 10 user units represent a length of 10 units in the current user coordinate system.
XML	Extensible Markup Language	XML is a general-purpose markup language. It is classified as an extensible language because it allows its users to define their own tags
VDK	VisualDSP Kernel	Operating system supported by Blackfin microprocessors

Term	Definition	Description
VLAN	Virtual LAN	A VLAN is a group of hosts with a common set of requirements that communicate as if they were attached to the Broadcast domain, regardless of their physical location. A VLAN has the same attributes as a physical LAN, but it allows for end stations to be grouped together even if they are not located on the same network switch. Network reconfiguration can be performed using software instead of physically relocating devices
ХРе	Windows XP Embedded	XPe is a componentized version of the Professional edition of Windows XP. An original equipment manufacturer is free to choose only the components needed, thereby reducing operating system footprint and also reducing attack area as compared with XP Professional. Unlike Windows CE, Microsoft's operating system for portable devices and consumer electronics, XP Embedded provides the full Windows API, and support for the full range of applications and device drivers written for Microsoft Windows. The system requirements state that XPe can run on devices with at least 32MB Compact Flash, 32MB RAM and a P-200 microprocessor
WUI	Web User Interface	WUI is the set of means by which people interact with a particular machine, device, computer program or other complex tool via the Web

18 Glossary

Terms in this Glossary are provided for informational purposes only and can describe features not included in your particular license.

Term	Definition
Actuator	A mechanical device for moving or controlling a mechanism or system. An actuator typically is a mechanical device which transforms an input signal (usually an electrical signal) into motion
Bandwidth	In computer networking, bandwidth often refers to a data rate measured in bits/s, for example, network throughput. The reason for the connection of data rate with the term bandwidth is that the limit to the data rate of a physical communication link is related to its bandwidth in hertz
Cam profiling	The position of a slave axis is mathematically linked to the position of a master axis. A good example of this would be in a system where two rotating drums turn at a given ratio to each other. A more advanced case of electronic gearing is electronic camming. With electronic camming, a slave axis follows a profile which is a function of the master position. This profile need not be linear, but it must be a mathematical function
CANopen	CANopen is a communication protocol and device profile specification for embedded systems used in automation for fieldbuses working in real-time
Caret	The term caret is also sometimes used in graphical user interface terminology where it means a text insertion point indicator, frequently represented by a blinking vertical bar. In this context, it can be used interchangeably with the word cursor , although the latter term is often reserved for a mouse pointer
Casting	For Typecasting, Typecasting
СОМ	COM is the original name of the serial port interface. It does not only refer to physical ports, but also to virtual ports, such as ports created by bluetooth or USB-to-Serial adapters
Contactor	A contactor is an electrically controlled switch (relay) used for switching a power circuit. A contactor is activated by a control input which is a lower voltage/current than that which the contactor is switching. Unlike a circuit breaker, a contactor is not intended to interrupt a short-circuit current
Datagram	A datagram is a basic transfer unit in which the delivery arrival time and order are not guaranteed. A datagram consists of header and data areas. The source and destination addresses as well as a type field are found in the header of a datagram.
DeviceNet	DeviceNet is a communication protocol (based on Controller Area Network) used in the automation industry to interconnect control devices for data exchange. Typical applications are information exchange, safety devices, and large I/O control networks
Drive	In electrical engineering, a drive is an electronic device providing power to a motor or servo, and controlling it through the current and timing in its coils

Term	Definition
Driver	In computing and electronics, a driver is a software component allowing higher-level computer programs to interact with a computer hardware device. A driver typically communicates with the device through the computer bus or communications subsystem to which the hardware is connected
Endian	Big-endian and little-endian describe the order in which a sequence of bytes are stored in computer memory. Big-endian is an order in which the "big end" (most significant value in the sequence) is stored first (at the lowest storage address). Little- endian is an order in which the "little end" (least significant value in the sequence) is stored first For example the decimal integer 56789652 (0x03628a94 in hexadecimal) is stored as follows:
	 0x03 0x62 0x8a 0x94 on big-endian 0x94 0x8a 0x62 0x03 on little-endian
	KAS applications can be downloaded to big-endian or little-endian processor targets
Environment	Environment objects are global objects that exist before the execution of the script. Typically, they are global objects of the KAS IDE that can be accessed from the script
EtherCAT	"Ethernet for Control Automation Technology" EtherCAT is an open, high-performance Ethernet-based fieldbus system. The development goal of EtherCAT was to apply Ethernet to automation applications which require short data update times (also called cycle times) with low communication jitter (for synchronization purposes) and low hardware costs
Ethernet	Ethernet is a large, diverse family of frame-based computer networking technologies that operate at many speeds for local area networks (LANs)
EtherNet/IP	An open industrial application layer protocol for industrial automation applications. The EtherNet/IP application layer protocol is based on the CIP layer
Fast Inputs	The inputs are taken into account at each cycle depending on the system periodicity (for example each millisecond). Under certain circumstances it can be insufficient when more accuracy is needed, or if a quick response is required from the system. To fill the gap, a drive can have some Fast Input connections (generally one or two). When an event happens that triggers a Fast Input (e.g. when a sensor sends a rising edge), the detection of a signal occurs faster (which can be 1000 times more accurate than the system periodicity). Then the timestamp associated with this input can be provided to the controller to take corrective action
Feedback Device	A process whereby some proportion of the output signal of a system is passed (fed back) to the input. In automation, a device coupled to each motor to provide indication of the motor's shaft angle, for use in commutating the motor and controlling its speed and position

Term	Definition
feed-forward	This describes an element or pathway within a control system which passes a controlling signal from a source in the control system's external environment, often a command signal from an external operator, to a load elsewhere in its external environment
Fieldbus	A Fieldbus is an industrial network protocol used for distributed control (e.g. EtherCAT, CAN, Profinet). It is a way of connecting instruments in a plant design
Flash Memory	A Flash memory is a non-volatile computer storage chip that can be electrically erased and reprogrammed. In addition to being non- volatile, flash memory offers fast read access times, as fast as dynamic RAM, although not as fast as static RAM or ROM. Its mechanic shock resistance explain the popularity over hard disks in portable devices; so does its high durability, being able to withstand high pressure, temperature, immersion in water etc.
Frame	In networking dialect, a message is called a frame
Front-end	In software design, the front-end is the part of a software system that interacts directly with the user
Homing	The Homing procedure allows, based on a position measurement, to set a position offset to the motor in order to ensure it is physically at the home position. The homing offset is saved in the controller.
Interrupt	An interrupt is an asynchronous signal from hardware indicating the need for attention or a synchronous event in software indicating the need for a change in execution
Jerk	In physics, jerk is the rate of change of acceleration; more precisely, the derivative of acceleration with respect to time
Latch	The control word is used to activate the drive's latch status machine. The latch control word is processed independently of the EtherCAT bus cycle. The status word is used to return the drive's latch status
MAC address	A Media Access Control address (MAC address) is a quasi-unique identifier assigned to most network adapters or network interface cards (NICs) by the manufacturer for identification. If assigned by the manufacturer, a MAC address usually encodes the manufacturer's registered identification number
ModBus	ModBus is a serial communications protocol and is now the most commonly available means of connecting industrial electronic devices. ModBus is often used to connect a supervisory computer with a
	remote terminal unit in supervisory control and data acquisition (SCADA) systems.
	Versions of the ModBus protocol exist for serial port and Ethernet (it is widely used with TCP/IP over Ethernet)
Motion Bus	A Motion bus is an industrial network protocol used for real-time distributed control (e.g. EtherCAT).

Term	Definition
Motion control	Motion control is a sub-field of automation, in which the position and/or velocity of machines are controlled using some type of device such as a hydraulic pump, linear actuator, or an electric motor, generally a servo. Motion control is an important part of robotics and CNC machine tools; however, it is more complex than in the use of specialized machines, where the kinematics is usually simpler. The latter is often called General Motion Control (GMC). Motion control is widely used in the packaging, printing, textile and assembly industries
Motor	An actuator focused to a movement, converting electrical energy in a force or torque
Non-volatile	Information is stored in a specific memory to remain accessible even when the application has been powered off
Online Change	Applies to downloading PLC code changes while the application is running.
Online Configuration Mode	Applies to EtherCAT communication to the AKD drives in a special mode.
OpenGL	OpenGL (Open Graphics Library) is a standard specification defining a cross-language, cross-platform API for writing applications that produce 2D and 3D computer graphics. The Softscope uses this API to implement graphical manipulations
P-code	P-code machine or pseudo-code machine is a specification of a CPU whose instructions are expected to be executed in software rather than in hardware. Programs that have been translated to P-code are executed (interpreted) by a software program that emulates the behavior of the CPU specification
PDO	 PDO is a type of protocol frame used in some fieldbuses. A PDO contains one or more object dictionary entries, which define the application data transferred between devices. EtherCAT uses the same communication mechanisms (PDO and SDO) as CANopen. EtherCAT transfers the process data between the master and slave device cyclically. PDOs have several attributes to define their properties. PDOs can be exclusive, meaning that no additional PDOs may be assigned to a device if an exclusive PDO is assigned.
	 If the PDO type is fixed (Fixed attribute = 1) then the PDO's content cannot be changed by users.
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
PLCopen	A vendor -and product- independent worldwide association active in Industrial Control and aiming at standardizing PLC file formats based on XML
Pragma	A compiler directive communicating additional "pragmatic" information. Pragmas are processed at compile time, not at run-time. They pass information to the compiler

Term	Definition
Precedence	In arithmetic and algebra, when a number or expression is both preceded and followed by a binary operation, a rule is required for which operation must be applied first. From the earliest use of mathematical notation, multiplication took precedence over addition, whichever side of a number it appeared on. Thus $3 + 4 \times 5$ = $5 \times 4 + 3 = 23$. To change the order of operations, we use parentheses (). Thus, if we want to force addition to precede multiplication, we write $(3 + 4) \times 5 = 35$
Probe	For Softscope -Probe, Softscope - Probe
Pulse	When the step gets activated, the action is activated for a single execution, and possibly once again when the step is deactivated
Reference Counting	In computer science, reference counting is a technique of storing the number of references, pointers, or handles to a resource such as an object or block of memory. It is typically used as a means of deallocating objects which are no longer referenced
Rising Edge	A rising edge is the transition of a digital signal from low to high. It is also called positive edge
Run-time	In computer science, run-time (or run time) describes the operation of a computer program, the duration of its execution, from beginning to termination (compare compile time)
Sensor	A sensor is a type of transducer that converts one type of energy into another for various purposes including measurement or information transfer
Service Port	UDP applications use datagram sockets to establish host-to-host communications. An application binds a socket to its endpoint of data transmission, which is a combination of an IP address and a service port. A port is a software structure that is identified by the port number, a 16 bit integer value.
Servo Drive	A servo drive is a special electric amplifier used to power electric servo motors. It monitors feedback signals from the motor and continually adjusts for deviation from expected behavior
Setpoint	Setpoint is the target value that an automatic control system (for example a PID controller) aims to reach
Softscope - Channel	A Channel is used by the softscope to acquire the evolution of a variable which is plugged on it
Softscope - Probe	A device that uses onboard instruments to gather and relay a variety of measurement to controllers from remote locations. Probes can return their data over radio links or be physically tethered to controllers or another device, or to collect and return physical samples
Softscope - Sampling	To acquire the variable's evolution, samples are taken at fixed intervals. The accuracy to create the trace depends on the resolution of the acquisition. The sampling frequency must be higher than 2 times the highest frequency in the input signal. It is called the Nyquist frequency. Theoretically it is possible to reconstruct the input signal with more than 2 samples per period. In practice, 10 to 20 samples per period are recommended to be able to examine the signal thoroughly

Term	Definition
Softscope - Time-base	The time-base allows you to set the speed at which all the lines for each channel are drawn, and is calibrated in milliseconds per division
Softscope - Trace	The trace is the resulting graph of a variable's evolution against time, with the more distant past on the left and the more recent past on the right
Synchronization	Combines an axis or axes group (as slave) with an axis as master so that the slave executes its path with synchronization to the progress of the master, meaning linked to a one-dimensional source for synchronization
SynqNet	SynqNet is a digital machine control network. Built on the 100BT physical layer, SynqNet provides a synchronous real-time connection between motion controllers, servo drives, stepper drives, I/O modules, and custom devices
Тад	In the HMI context, objects connected to tags can change values in a controller, and controller values can be reflected by changing object appearance in various ways. A tag has a symbolic name and can be of different data types. Tags can belong to a connected controller, be internal or belong to the system.
Timestamp	A timestamp is a sequence of characters denoting the date and/or time at which a certain event occurred
Torque	Torque is the tendency of a force to rotate an object about an axis. Just as a force is a push or a pull, a torque can be thought of as a twist. The SI unit for torque is the newton metre (N.m).
Typecasting	In computer science, type conversion or typecasting refers to changing an entity of one data type into another. It is done to take advantage of certain features of type hierarchies. For instance, values from a more limited set, such as integers, can be stored in a more compact format and later converted to a different format enabling operations not previously possible, such as division with several decimal places' worth of accuracy. There are two types of conversion: implicit and explicit. The term for implicit type conversion is coercion. The most common form of explicit type conversion is known as casting. Explicit type conversion can also be achieved with separately defined conversion routines such as an overloaded object constructor

19 Licenses

AjaxFileUpload – MPL License

See www.mozilla.org/MPL/2.0/ for the Mozilla Public License.

Apache log4net – Apache License

Apache License Version 2.0, January 2004 http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

bsdtar & libarchive 2 - BSD License

Copyright (c) 2003-2006 Tim Kientzle All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer in this position and unchanged. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE AUTHOR(S) ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

bzip2.dll – BSD License

bzip2.dll

Copyright © 1996-2007 Julian Seward

This program, bzip2, the associated library libbzip2, and all documentation, are copyright © 1996-2007 Julian Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: • Redistributions of source code must retain the above copyright notice, this list of conditions and the following

disclaimer. • The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If

you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

• Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

• The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DockPanel Suite – MIT License

DockPanel

The MIT License

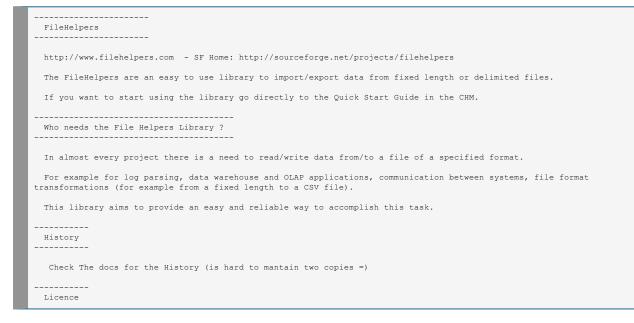
Copyright (c) 2007 Weifen Luo (email: weifenluo@yahoo.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONIMFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

FileHelpers





jQuery Cookies

Copyright (c) 2005 - 2013 James Auldridge

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

jQuery File Tree – MIT License

JQueryFileTree

Copyright (c) 2008 A Beautiful Site, LLC.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABLLITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINCEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

JavaScript Graphics Library – GNU Lesser General Public License

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. http://fsf.org/ Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

KAS IDE User Guide | 19 Licenses

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice, keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read

<http://www.gnu.org/philosophy/why-not-lgpl.html>.

Mongoose – MIT License

Copyright (c) 2004-2013 Sergey Lyubka

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MVVM Light Toolkit – MIT License

MVVM Light Toolkit

http://mvvmlight.codeplex.com/

License: The MIT License (MIT)

Copyright (c) 2009 - 2011 Laurent Bugnion

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

pugixml – MIT License

Copyright (c) 2006-2012 Arseny Kapoulkine

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONIMFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Qt – LGPL3 License

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

 a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following: •0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

•1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining

where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Qwt – GNU Lesser General Public License

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice, keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy,

or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY, for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read

<http://www.gnu.org/philosophy/why-not-lgpl.html>.

ZedGraph – LGPL License

For easy distribution, this archive provides the dll form of ZedGraph only. This distribution of the ZedGraph source code includes two versions:

Version 4.6.x is the .Net 1.1 compatible version Version 5.1.x is the .Net 2.0 compatible version

Complete source code for ZedGraph is available on sourceforge here:

http://sourceforge.net/projects/zedgraph/

A wiki providing help, samples, etc. is available here:

http://zedgraph.org

934

A tutorial on using ZedGraph is available here:

http://www.codeproject.com/csharp/zedgraph.asp

GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999 Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software-to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A

"work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or inked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law. If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to

contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990 Ty Coon, President of Vice

That's all there is to it!

Zlib1.dll – BSD License

zlib1.dll

938

Copyright © 1995-2004 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler jloup@gzip.org madler@alumni.caltech.edu

If you use the zlib library in a product, we would appreciate *not* receiving lengthy legal documents to sign. The sources are provided for free but without warranty of any kind. The library has been entirely written by Jean-loup Gailly and Mark Adler; it does not include third-party code.

If you redistribute modified sources, we would appreciate that you include in the file ChangeLog history information documenting your changes. Please read the FAQ for more information on the distribution of modified source versions.

20 Index

#	
#ifdef	210
@	
@	297
1	
16#	356
2	
2# 2nd feedback	·····356 ·····882
8	
8#	
Α	
About Window	
acceleration	
accelerator keysacronyms	
actual position	417 465
pipe network	
adding	
controller	
drive	
Ι/Ο	
address	
IP address	121
AKD drive	
configure	
creation	
GUI	
offline	
online setup wizard	
toolbar	
workbench	110
AKD2G, digitizing axis	
AKT-ECT-000	107
troubleshoot	
alias	······309, 370
ALS format	242
animation	4, 231, 234, 604
PLC cycle	
UDFBarchitecture	
auto	
completion	

discovery	
recovery	
scan	
start	
autocompletion	
AutoSave	
autostart	
Axes Group	
Axes Group, new	
Axis Velocity; velocity, axis	
AXIS_GROUP_REF	
AXIS_NUM	
AXIS_REF	

В

Backup Controller	
Backup EtherCAT Devices	502
bandwidth	
BISS	
bookmarks	
BOOL	
boot	501
PDMM	
bootstrap	
breakpoint	
remove	
set	
buffer	
pipe network	
PLCopen mode	
button	33,
online change	
softscope	
BYTE	
	501

С

C code	
cam	2.0
cam profile	····· 58, 199
format	
Cam on the fly	
cam profile	
transformation	
CAN object	
CANopen	
CANopen object	
Capture Engine	
cascade	
case sensitive	
change	
online change	601
channel	
child SFC	
circular file	
Circular Move, precision requirements; Precision, Circular Move Input Red	quirements582
clock synchronism	
code	
C code	210

	color code	
	machine code	
	Normal	
(Optimized	······210 [,] 604
	P-code	
COE .		434
	start	
collar		
color	FFLD network	
	green da	0 440 004 540
	green	
	orange	
	red	
	nand position	
comn		
	FBD	281
	FFLD	
	pipe network	
Comr		140
	FFLD	
	nent, FFLD	
comp		250
	PLC programs	606
	pare Projects	
	atibility	
Comp	bile	
	iling mode	
	debug	
	release	
	letion	
condi	tional compiling	
config	gure	
/	AKD drive	
	EtherCAT XML	
	IO	
const	ant	
	ant expression	
conta	ffld	200
contr	ol word	
contr		
	creation	120
Ì	log	88.90
	version	
Contr	oller Performance, tip	
Contr	oller Properties	
contr	oller; backup	
Contr	oller; restore	
conve	ention	
١	variable naming	
1	Windows standard	
coord	linate system	
	unified	
	ights	
coupl	er	······ 766 [,] 902
CPU I	oad	···542, 878, 885

creation	
AKD drive	
controller	
controller wizard	
pipe network	
program	
structure	
variable	
CSV	
CurrentPosition	
Pipe Network	
curve	
cam profile editor	
softscope	
custom library	
online change	
cycle	
animation	
cycle time	
motion	
PLC	
Cycle Time	
CycleJitter	
cyclic	

D

dashed line	
data structure	
data types	
DC	
debug	
PLC application	
softscope	
step-by-step	
defines	
Derivation Order	
device	
EtherCAT	
diagnostics, ethercat	
dictionary	
difference	606
digitizing axis	155, 446, 449, 633
Digitizing axis	620
DINT	
DINT dip switches	
DINT	
DINT dip switches directive	
DINT dip switches directive	
DINT dip switches directive compiler directive disclaimer discovery	
DINT dip switches directive compiler directive disclaimer discovery distributed clocks	
DINT dip switches directive compiler directive disclaimer discovery distributed clocks docking windows	
DINT dip switches directive compiler directive disclaimer discovery distributed clocks	
DINT dip switches directive compiler directive disclaimer discovery distributed clocks docking windows dotted line download	
DINT dip switches directive compiler directive disclaimer discovery distributed clocks docking windows dotted line	
DINT dip switches directive compiler directive disclaimer discovery distributed clocks docking windows dotted line download	
DINT dip switches directive compiler directive disclaimer discovery distributed clocks docking windows dotted line download drive firmware HMI drag and drop	
DINT dip switches directive compiler directive disclaimer discovery distributed clocks docking windows dotted line download drive firmware HMI drag and drop drive	
DINT dip switches directive compiler directive disclaimer discovery distributed clocks docking windows dotted line download drive firmware HMI drag and drop	

AKD offline	
AKD online	
AKD setup wizard	
configure [']	
duration	105
cvcle	
cycle online change	
DWORD	
	J01

Ε

editor	
cam profile	
FBD	
FFLD	
НМІ	
IL	
pipe network	
SFC	272
ST	292
variable	
ENI File	440
enum,enumerated type	
enumeration order	
error	
EtherCAT error management	
EtherCAT error message	
PLCopen errorID	
error handling	
pipe network	
ESI file	34, 323, 326, 331, 883
ESI File	
ETG.5001 standard	
EtherCAT	
distributed clocks	
error management	
error message	
FoE	
frame	
image	
master	
modes	
online configuration mode	
PDO	122 122 122 122 1201
Process Image	442
profile	437
SDO	
slave	
status bootstrap	
status operational	
status preop	
status safeop	
topology	
unsupported device	638
EtherCAT Devices; backup	
EtherCAT Devices; restore	
EtherCAT Processing Unit Error Counter	
execution order	
expand	
FFLD network	

export	
program	
softscope data	
F	
FailSafe over EtherCAT	
faq	
Fast input Fault messages	
FBD	
editor	270
insert graphic	
feed-forward	
torque-	
feedback	
secondary	
feedback position	
FFLD	
editor	
Limitations	
filtering	
find	
case sensitive	
find and replace	
find next	
find unused	
firewall	
firmware download protocol	
download protocol	
drive download	
drive upgrade PDMM upgrade	
FoE	
forcing variable	100 222
format	
ALS project	2/12
KAS project	
Forwarded Rx Error Counter	
frame EtherCAT	
Frame Loss	
FSoE	
FSoE Address	
function	
function block	

G

generator position	
getting started	
global constant	20, 300
glossary	
green	512
AKD enable status	
background	
dashed rectangle	
FFLD network header	
grey	
AKD enable status	
grid	
5	505

grid unit	
grid unit GUI	303
AKD drive	
KAS	
guideline	51
PLC program	
project structure	
setting units	
5	110

Η

hexadecimal	
add device	
download	
Holding Registers	
homing	
how to	
list of	

Ι

I/O local I/O terminal icon TL import import program57 initialization

	nce		
inter	lisense	•••••	287
Inter IO	polation		515
	adding IO		.188
	configure		
	local		
	mapping IO		
	onboard		186
	PDMM onboard		100
	unmapping IO		
IO te	rminal		514
	coupler	766	002
	isolation		
	module		
	stepper		
	thermocouple		
IP ad	Idress	/05/	90Z
isola	tion	50, 121,	498
1301a	uon	/66/	902
I I			
J			
jerk .			524
		000	
Κ			
	F		
	format		
KVB .			549
L			
latch	۱	206	125
	icy		
			401
libra			491
	ر ا custom library		
			100
lifati	toolbox		195
meun	toolbox		69
limit	toolbox		69
limita	toolbox me ations		69 390
limita	toolbox me		69 390 -403
limita	toolbox me ations acceleration animation		69 390 -403 -231
limita	toolbox me		69 390 -403 -231 222
limita	toolbox me		69 390 -403 -231 222 -221
limita	toolbox me		69 390 -403 -231 222 -221 458
limita	toolbox me ations acceleration animation breakpoint in SFC breakpoint with online change EtherCAT in Op state HMI variable mapping	215,	69 390 -403 -231 222 -221 458 -172
limita	toolbox	215,	69 390 -403 -231 -222 -221 458 -172 -287
limit	toolbox	215,	69 390 -403 -231 -222 -221 458 -172 -287 -403
limit	toolbox	215,	69 390 -403 -231 -222 -221 -458 -172 -287 -403 -186
limit	toolbox	215,	69 390 -403 -231 -222 -221 458 -172 -287 -403 -186 -602
limit	toolbox	215,	69 390 -231 222 -221 458 -172 -287 -403 -186 -602 -323
limit	toolbox me ations acceleration animation breakpoint in SFC breakpoint with online change EtherCAT in Op state HMI variable mapping intellisense jerk onboard IO online change online detection PDMM onboard IO	215,	69 -403 -231 -222 -221 -458 -172 -287 -403 -186 -602 -323 -484
limit	toolbox	215,	69 390 -403 -231 222 -221 458 -172 -287 -403 -186 -602 -323 -484 -367
limit	toolbox me ations acceleration animation breakpoint in SFC breakpoint with online change EtherCAT in Op state HMI variable mapping intellisense jerk onboard IO online change online detection PDMM onboard IO PLC program print preview	215,	69 390 -403 -231 222 -221 458 -172 -287 -403 -186 -602 -323 -484 -367 237
limit	toolbox me ations acceleration animation breakpoint in SFC breakpoint with online change EtherCAT in Op state HMI variable mapping intellisense jerk onboard IO online change online detection PDMM onboard IO PLC program print preview project files	215,	69 390 -403 -231 222 -221 458 -172 -287 -403 -186 -602 -323 -484 -367 -237 235
limit	toolboxme	215,	69 390 -403 -231 222 -221 458 -172 -287 -403 -186 -602 -323 -484 -367 -237 -235 95
limit	toolbox	215,	69 390 -403 -231 222 -221 458 -172 -287 -403 -186 602 -323 -484 -367 -237 235 95 -323
limit	toolbox me ations acceleration animation breakpoint in SFC breakpoint with online change EtherCAT in Op state HMI variable mapping intellisense jerk onboard IO online change online detection PDMM onboard IO PLC program print preview project files replace scan device search and replace	215,	69 390 -403 -231 -222 -221 458 -172 -287 -403 -186 602 -323 -484 -367 -237 -235 95 -323 95
limit	toolbox	215,	69 390 -403 -231 -222 -221 458 -172 -287 -403 -186 602 -323 -484 -367 -237 -235 95 -323 95 -222

line	structure UDFB	
	dashed line	465
	dotted line	
	normal line	
	solid line	
Link	Loss/Device Fault	
Link	Lost Counter	
LINT		
list o		
	how to	
	literature	
litera	ature	
	list of-	
local	l constant	
local		
local	l logs	
locat		
	find and replace	95
	library	195
	project	
locki log	ng variable	······108, 233
	circular file	90
	controller log	
	filtering	
	level	
	local logs	
	log file	
	scrolling	
	source	
	timestamp	
LREA	NL	
LREA	AL, data type	356

Μ

machine code manuals mapping	
HMI variable	
I/Oonboard I/O	
PDMM onboard I/O	
master	
MDP	
message	
circular file	
filtering	
level	
local logs	
log file	
scrolling	
source	
timestamp	
MLPN_ACTIVATE	
MLPN_CONNECT	
MLPN_CREATE_OBJECTS	

MLPN_DEACTIVATE	
MLPN_POWER_OFF	
MLPN_POWER_ON	
MLPR_CREATE_PROFILES	
modbus	
Modbus	
master, multiple	
module	
modulo	
Modulo Period	
motion	
initialization	
profile	
restart	
start	
MotionExecTime	
multi-dimension	
multiple	

Ν

Ν

SFC step	
Name, node	
New Axis	
New Program dialog	306
node name	54
Normal code	
normal line	
NVRAM	
AKD parameter	
calculate space	

0

Object Index	
Object Name	
octal	
offline	
AKD drive	
onboard I/O	
online	100
AKD drive	
online change	490 601
online config	
online configuration mode	195, 252, 157, 725, 977
Op	//8 יכב/ י/כב יכסן יוסייבט יכסיים און איניביע איז
open from controller	
Optimized code	245
option	210/ 004
orange	209
5	
ordering variables	2/9 دح
oscilloscope	
oscilloscope	528

output parameter overload CPU	
P	
P-code P0	
SFC step P1	
SFC step Pack bits panning	
cam profile editor	
input output Parameters dialog	
password	·····508 ·····875
PDF PDMM onboard I/O PDO	
PDO configuration PDO Index	
period periodic phase	
Physical Link status PID	
pipe position PLC	
cycle	209
PLCopen introduction queuing	
S-curve Trapezoidal PLCProgExecTime	401
plugging a probe port	······535 ·····878, 898
ports, used by KAS position actual position	
feedback position	
pipe position	
Position Actual Command	
CurrentPosition Power ON Delta Offset	
Zero Offset pou Power ON Delta Offset	
Pipe Network	

pragma	
pre-op	
precision	
preview	
print	
preview	
project	
setup	
printed material	
priority	
private variable	
probe	
profile	333
cam profile	58, 199
EtherCAT	
motion profile	277
program	
Program	
Limitations	267
Program Properties	206
proxy	
pulse	
	(0)
online change	602

Q

aueuina	
•	27. 500

R

read only	
REAL	
REAL, data type	356
RealTimeMargin	534, 544, 885
recovery	
red	242, 24,, 8,0
AKD enable status	110
background	111
difference	
line	
text	
reference manual	
reference position	296 465
Registers, Holding	
Registers, Input	
registration	
regulation	
release	
remote I/O	
remote version	
replace all	
replace next	
Reset IDE Layout	
restart	
motion	
Restore Controller	
Restore EtherCAT Devices	
retain variable	
calculate space	

starting application	
variable editor	
revert online change	605
root	508
Rotary Switch	
PDMM	
runtime	
Rx Error Counter	
Rx Invalid Frame Counter	
RxPDO	

S

S-curve	
\$300	······872, 903
\$700	······872, 903
safe-op	
SafePLC2	
Safety	640
save	
scan	
scope	······225, 528
scrolling	
SCU-1-EC	
SDO	
AKD capture engine	
update rate	
secondary feedback	
servo axis	
set number of inputsetup	194
print	226
SFC	236
breakpoint	222
child	
editor	
Limitations	
Program size	
timeout	
when using SFC	
shortcut	205
FBD	
FFLD	
graphic	
SFC	
ST	
table	
simulation	
EtherCAT slave	
simulator	
SINT	
slave	123, 433, 438
softscope	
solid line	
sort variables	
source	90, 491
source code	147
Spacebar	
splitter	
cam profile editor	513

softscope	528
editor	
start	
motion	
Start	
state machine	
application structure	
ĊĂNopen	
EtherĊAT	
online change	
pipe network	
PLCopen	
status bar	
status word	
step-by-step debugging	
stepper	766 002
STRING	902
structure	202 256
creation	
subprogram	
synchronization	
syntax	
conditional compiling	210
edit variable	
ST coloring	
system requirements	

Т

tables	
tag	
IO mapping	
tasking	
technical reference	
Temperature Limit	
template	
2 axes templates	
select template	
Templates; Project Templates	690
thermocouple	
tile	248
ТІМЕ	262
time-frame	522
Time Scale Offset	527
timeout	
SFC	882
timestamp	
toolbar	
toolbar AKD drive	
AKD drive	
AKD drive FBD	
AKD drive FBD FFLD	
AKD drive FBD FFLD IDE	
AKD drive FBD FFLD IDE SFC	
AKD drive FBD FFLD IDE SFC toolbox	
AKD drive FBD FFLD IDE SFC toolbox tooltip	
AKD drive FBD FFLD IDE SFC toolbox tooltip topology	
AKD drive FBD FFLD IDE SFC toolbox tooltip	

torque feed-forward	
scaling	
scaling	
TraceTimes	
track selection	
trademarks	
troubleshooting	
tunnel	
ТхРОО	443
	-++5

U

UDFB	
animation	
UDINT	
UDP	
UINT	
ULINT	
undocking windows	
unified coordinate system	
unit per ulision	
units	
unmap	101
і́/О	
unsupported EtherCAT device	
unused	000
find variable	
upgrade	55
drive firmware	
user manual	
User Units	
USINT	

V

variable	
animation	
create structure	
creation	
dictionary	
FBD	
forcing	
locking	
mapping I/O	
mapping onboard I/O	
mapping PDMM onboard I/O	
monitoring	
naming convention	
plugging a probe	
sorting	
unmapping I/O	
variable selector	.
map I/O	
velocity	
Velocity compensation	
Velocity Compensation Factor	
Velocity Compensation Filter	
versinfo.xml	
virtual machine	
Virtual machine	

Virtual Servo	155
VPN	
	0.00

W

warm start	
watch window web server	
change IP address	
upgrade firmware	
Webserver	190
log in; log in	
fail	
window	
cascade	
MDI	
tile	
wizard	210
AKD setup	
controller	
WORD	
Working Counter	
Wrong/Missing Device	
······································	

Х

X32 . X32,		71
	X3263	37
, -	configuration file	25 38

Ζ

Zero Offset	
Pipe Network	
zoom	
FBD	
FFLD	
SFC	
softscope	
•	555

About KOLLMORGEN

Kollmorgen is a leading provider of motion systems and components for machine builders. Through world-class knowledge in motion, industry-leading quality and deep expertise in linking and integrating standard and custom products, Kollmorgen delivers breakthrough solutions that are unmatched in performance, reliability and ease-of-use, giving machine builders an irrefutable marketplace advantage.



 Join the Kollmorgen Developer Network for product support. Ask the community questions, search the knowledge base for
 answers, get downloads, and suggest improvements.



North America KOLLMORGEN 201 West Rock Road Radford, VA 24141, USA

Web:www.kollmorgen.comMail:support@kollmorgen.com

- **Tel.:** +1 540 633 3545
- **Fax:** +1 540 639 4162

South America

KOLLMORGEN Avenida João Paulo Ablas, 2970 Jardim da Glória, Cotia – SP

CEP 06711-250, Brazil

web:	www.kollmorgen.com
Mail:	contato@kollmorgen.com
Tel.:	+55 11 4615-6300

Europe KOLLMORGEN Europe GmbH Pempelfurtstr. 1 40880 Ratingen, Germany

Web: www.kollmorgen.com

Mail: technik@kollmorgen.com

- **Tel.:** +49 2102 9394 0
- **Fax:** +49 2102 9394 3155

China and SEA KOLLMORGEN

Room 302, Building 5, Lihpao Plaza, 88 Shenbin Road, Minhang District, Shanghai, China.

Web:	www.kollmorgen.cn
Mail:	sales.china@kollmorgen.com
Tel.:	+86 - 400 668 2802
Fax:	+86 - 21 6248 5367