AKD®2G

PROFINET Communications Manual

PI CERTIFIED
PROFIBUS • PROFINET

Edition April 2021, A

Part Number 907-200012-00

Keep all manuals as a product component during the life span of the product.
Pass all manuals to future users or owners of the product.

Record of Document Revisions

Revision	Remarks	Date
A	Preliminary Edition	4/2021

Trademarks

- AKD is a registered trademark of Kollmorgen Corporation
- 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

Technical changes which improve the performance of the device may be made without prior notice.

1 Table of Contents

2 General

2.1 About this Manual

This manual, AKD2G *PROFINET Communication*, describes the installation, setup, range of functions, and software protocol for the PROFINETAKD2G product series. All AKD2G PROFINET drives have built-in PROFINET functionality; therefore, an additional option card is not required.

A digital version of this manual (pdf format) is available on the DVD included with the drive. Updated manuals can be downloaded from the Kollmorgen website.

Related documents for the AKD2G series include:

- *AKD2G Installation Manual* This manual provides instructions for installation and drive setup.
- *AKD2G WorkBenchOnline Help*. This help system describes how to use your drive in common applications. It also provides tips for maximizing your system performance with the AKD2G. The *Online Help* includes the *Parameter and Command Reference Guide* which provides documentation for the parameters and commands used to program the AKD2G.
- *Accessories Manual*. This manual provides documentation for accessories like cables and regen resistors used with AKD2G. Regional versions of this manual exist.

Additional documentation:

- Profile-PROFIdrive (PI group, Profile-PROFIdrive_3172_v42_Oct15.pdf)

2.2 Target Group

This manual addresses personnel with the following qualifications:

- Installation: only by electrically qualified personnel.
- Setup: only by qualified personnel with extensive knowledge of electrical engineering and drive technology.
- Programming: software developers, project-planners.

The qualified personnel must know and observe the following standards:

- ISO 12100, IEC 60364 and IEC 60664.
- National accident prevention regulations.

2.3 Abbreviations Used

Abbreviation	Meaning
API	Application Process Identifier
AR	Application Relations
BF	Bus Fault
Cat	Category
DCP	Discovery and Configuration Protocol
DO	Drive Object
DSC	Dynamic Servo Control
DU	Data Unit
GSD	General Station Description
GSDML	General Station Description Markup Language
HMI	Human Machine Interface
ID	Identifier
I/O	Input/Output
IRT	Isochronous Real-Time
LED	Light Emitting Diode
MDI	Manual Data Input
NRT	Non-Real Time
NV	Non-Volatile
PAP	Programm Ablauf Protokoll (program sequence protocol)
PLC	Programmable Logic Control
PLL	Phase-Locked Loop
PNU	Parameter number
RT	Real-Time
SoC	System on a Chip
SF	System Fault
STW	Controlword
UDP	User Datagram Protocol
ZSW	Statusword

3 Installation and Setup

3.1 Important Instructions

DANGER **High Voltage up to 900 V!**

There is a danger of electrical arcing with damage to contacts and serious personal injury.

- Never undo any electrical connections to the drive while it is live.
- Wait at least seven minutes after disconnecting the drive from the main supply power before touching potentially live sections of the equipment (e.g. contacts) or undoing any connections.

WARNING **Automatic Restart!**

Risk of death or serious injury for humans working in the machine. Drives with PROFINET are remote-controlled machines. They can start to move at any time without previous warning.

- Take appropriate measures to ensure that the operating and service personnel is aware of this danger.
- Implement appropriate protective measures to ensure that any unintended start-up of the machines cannot result in dangerous situations for personnel or machinery.
- Software limit-switches are not a substitute for the hardware limit-switches in the machine.

Electronic equipment is not failure-proof. The user is responsible for ensuring, in the event of a failure of the drive, the drive is set to a state that is safe for both machinery and personnel.

NOTICE

Install the drive as described in the *Installation Manual*. The wiring for the analog setpoint input and the positioning interface, as shown in the wiring diagram in the *Installation Manual*, is not required. Never break any of the electrical connections to the drive while it is live. This action can result in destruction of the electronics.

NOTICE

The drive's status must be monitored by the PLC to acknowledge critical situations. Wire the FAULT contact in series into the emergency stop circuit of the installation. The emergency stop circuit must operate the supply contactor.

NOTE

Because of the internal representation of the position-control parameters, the position controller can only be operated if the final limit speed of the drive does not exceed:

Rotary	Linear
at sinusoidal² commutation: 7500 rpm	at sinusoidal² commutation: 4 m/s
at trapezoidal commutation: 12000 rpm.	at trapezoidal commutation: 6.25 m/s

NOTE

All the data on resolution, step size, positioning accuracy, etc. refer to calculatory values. Non-linearities in the mechanism (backlash, flexing, etc.) are not taken into account. If the final limit speed of the motor must be altered, then all the parameters previously entered for position control and motion blocks must be adapted.

3.2 PROFINET Onboard

Connection to the PROFINET network is made using the X11 or X12 connector located at the top of the drive.

3.2.1 LED Functions

The communication status is indicated by the built-in LEDs.

Connector	Name	LED	Function
X11	Port 2	Bus fault (BF)	Off: No Error
			Flashing 2Hz: No Data Exchange
			On: No configuration, no link, or low-speed physical link
		Link/Activity	Off: No physical connection
			On/Flashing: Physical link present/activity
X12	Port 1	System Fault (SF)	Off: No Error
			Flashing 1Hz: DCP blink triggered by PROFINET
			On: Watchdog timeout, system error
		Link/Activity	Off: No physical connection
			On/Flashing: Physical link present/activity

3.2.2 Connection Technology

Connection to the PROFINET network is achieved by using RJ-45 connectors with standard Cat. 5 Ethernet cables for either connection configuration.

3.3 Guide to Setup

NOTICE

Only professional personnel with extensive knowledge of control and drive technology are allowed to setup the drive.

WARNING **Automatic Restart!**

Risk of death or serious injury for humans working in the machine. Drives with PROFINET are remote-controlled machines. They can start to move at any time without previous warning.

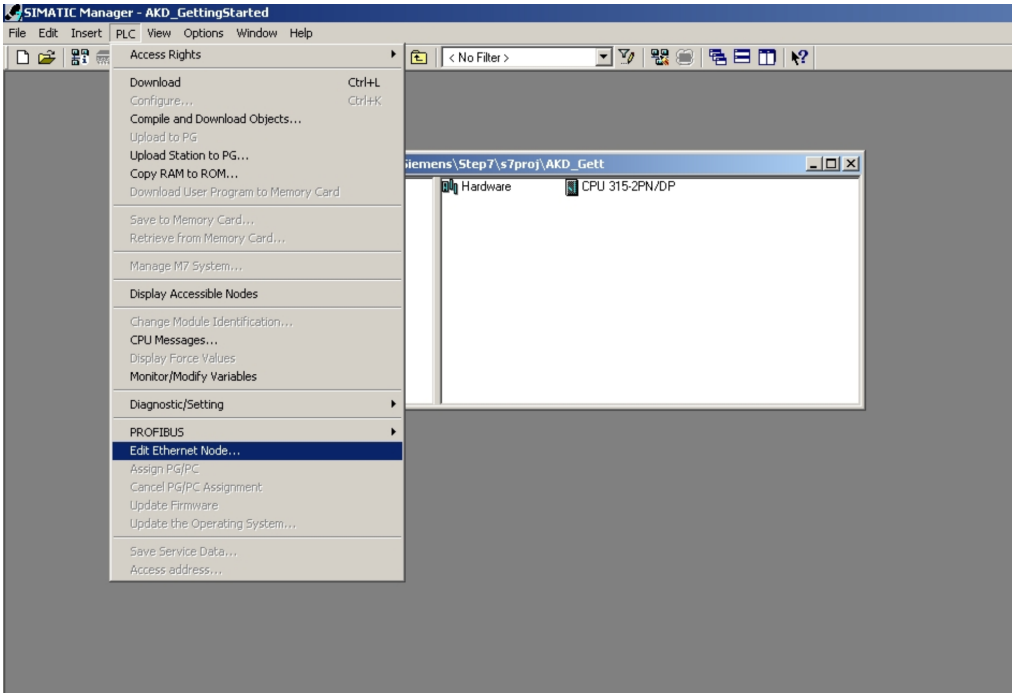
- Take appropriate measures to ensure that the operating and service personnel is aware of this danger.
- Implement appropriate protective measures to ensure that any unintended start-up of the machines cannot result in dangerous situations for personnel or machinery.
- Software limit-switches are not a substitute for the hardware limit-switches in the machine.

Electronic equipment is basically not failure-proof. The user is responsible for ensuring that, in the event of a failure of the drive, the drive is set to a state that is safe for both machinery and personnel.

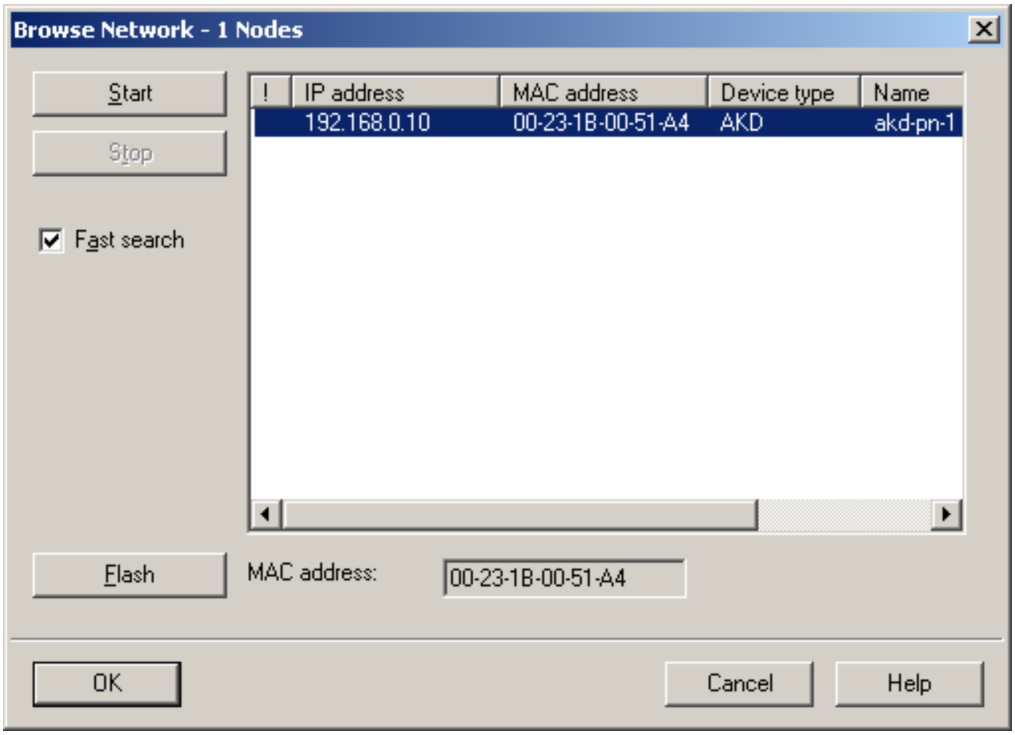
1. Check assembly and installation. Check that all the safety instructions in the product manual are observed and implemented. Check station address and baud rate setting.
2. Connect PC,start WorkBench. Use the setup software WorkBench to set the parameters for the drive.
3. Setup basic functions. Start up the basic functions of the drive and optimize the current, speed and position controllers. This section of the setup is described in the online help of the setup software.
4. Save parameters. When the parameters are optimized, save them in the drive.

3.4 Configure IP Address Parameters

1. Start the SIMATIC Manager.
2. Go to PLC > Edit Ethernet Node to assign a new IP address.



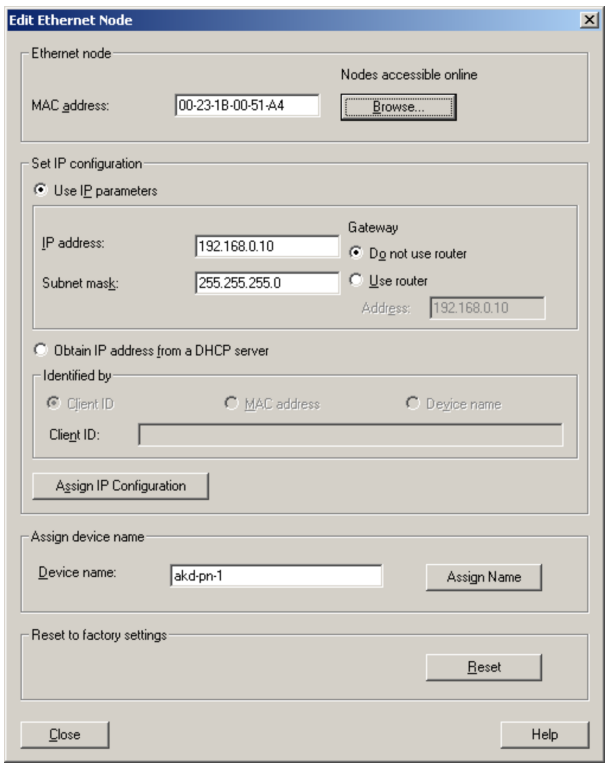
3. In the Browse Network window, click **Browse** in the Ethernet node group and look for all PROFINET devices in the network.



4. Select the AKD2G and click **OK**. If there are several AKD2Gs in the network, use the MAC address to filter one PROFINET device.

5. Click **Flash** to be sure the intended device is selected. The display of the selected device flashes if this function is active.
6. Click the **Use IP parameters** radio button, enter a new IP address and subnet mask to AKD2G.
7. Click **Assign IP Configuration**.
8. Use the same pop-up to change the device name. Type a name in the Device Name field and click **Assign Name**.

Each device connected to the same I/O connection must have a unique name. The PROFINET device name for AKD2G is derived from the AKD2G DRV.NAME parameter. The PLC, which acts as PROFINET IO-Controller, uses the Device Name address. You can change the IP address for each Device Name.

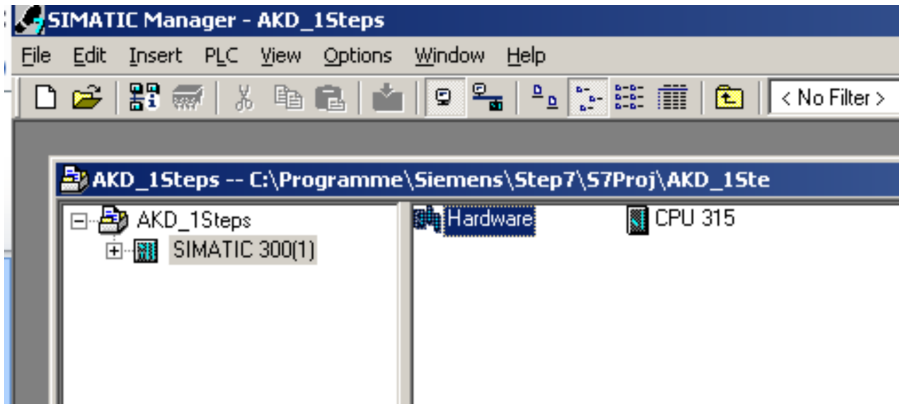


A message appears to indicate the change was successful and the AKD2G display shows the new address. If a failure message appears, make sure that no I/O connection is currently running then retry the address or name change.

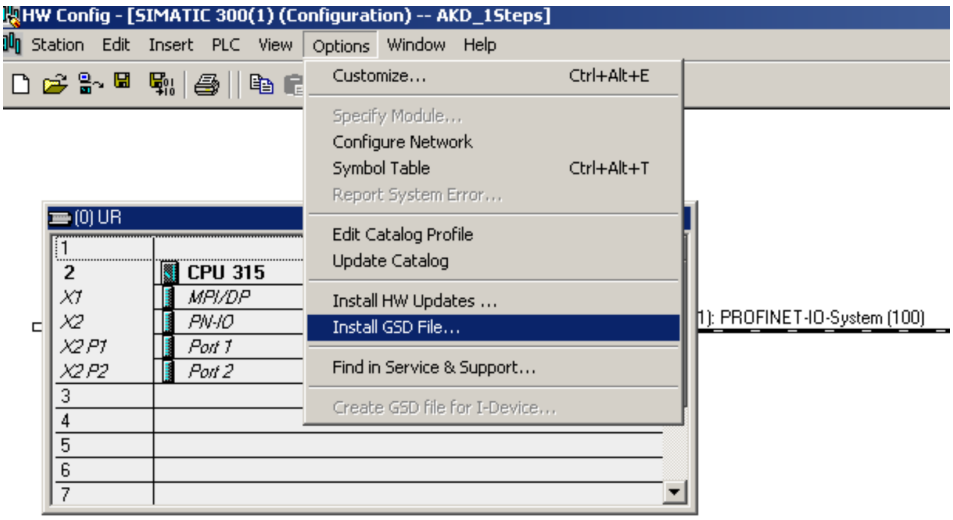
A current connection of WorkBench to AKD2G is disconnected when the IP address is changed. When this happens, reconnect to the new IP address.

3.5 Setup Step 7

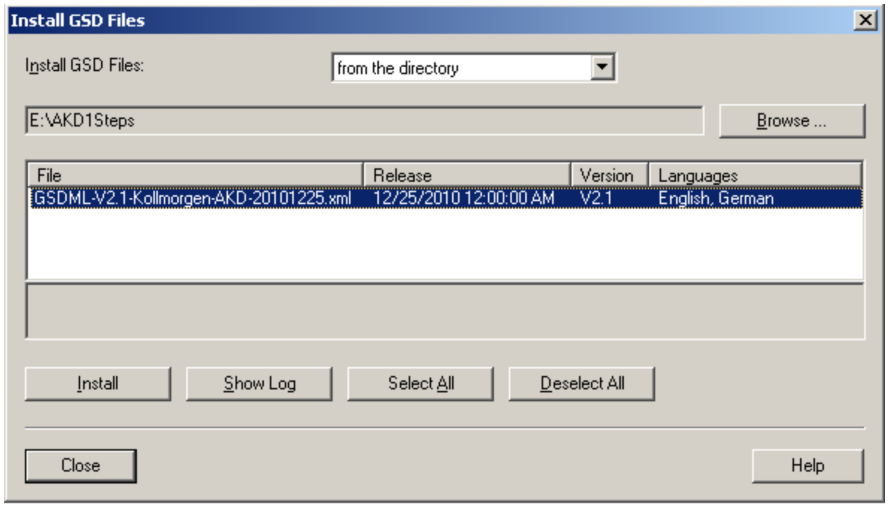
1. Start the SIMATIC Manager.
2. Open the hardware manager and double-click **Hardware**.



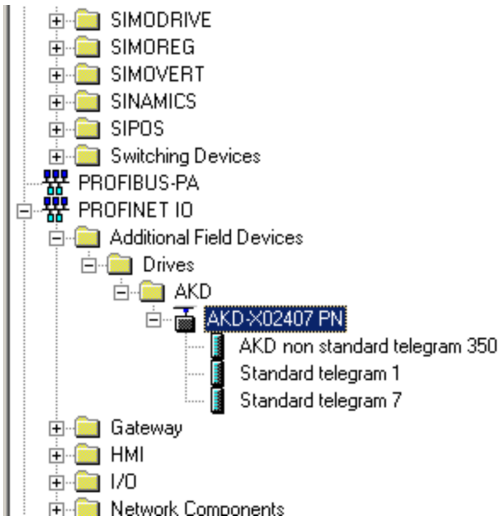
3. Go to **Options** and click **Install GSD Files**. The GSDML files for PROFINET devices can be installed.



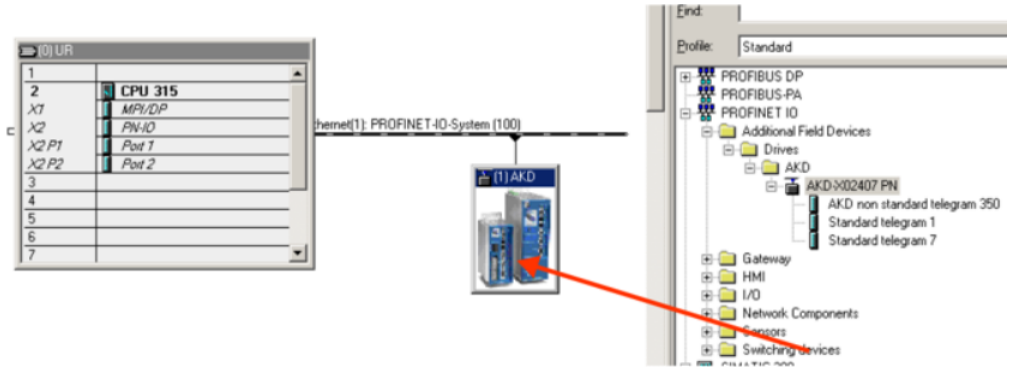
4. Browse for the latest AKD2G GSDML file and click **Install**:



5. The AKD2G GSDML file is installed and is found in the SIMATIC hardware catalog.
Open PROFINET IO > Additional Fieldbus Devices > Drives > AKD2G



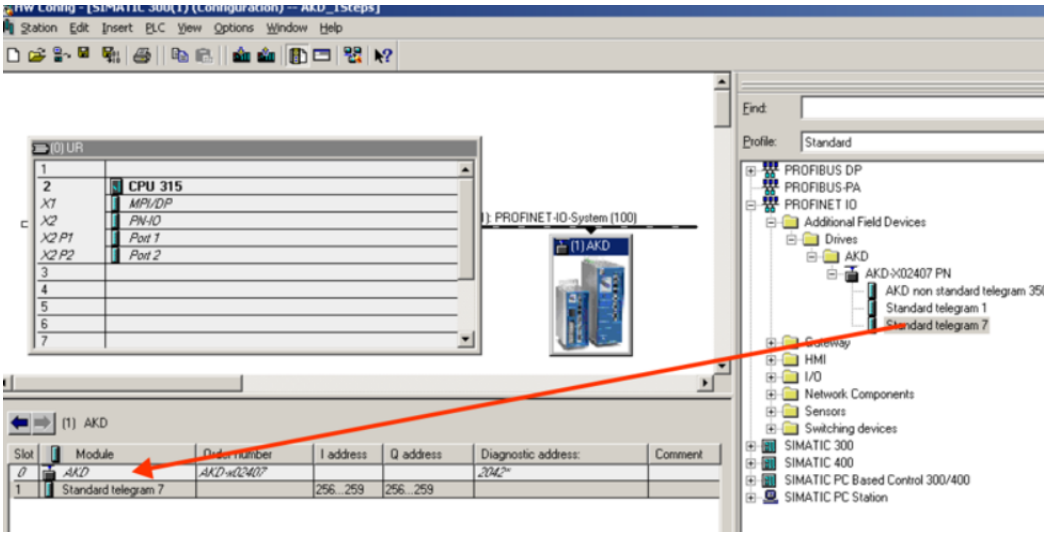
6. Click on the AKD2G device (not a telegram) and connect it to the PLC using drag and drop.



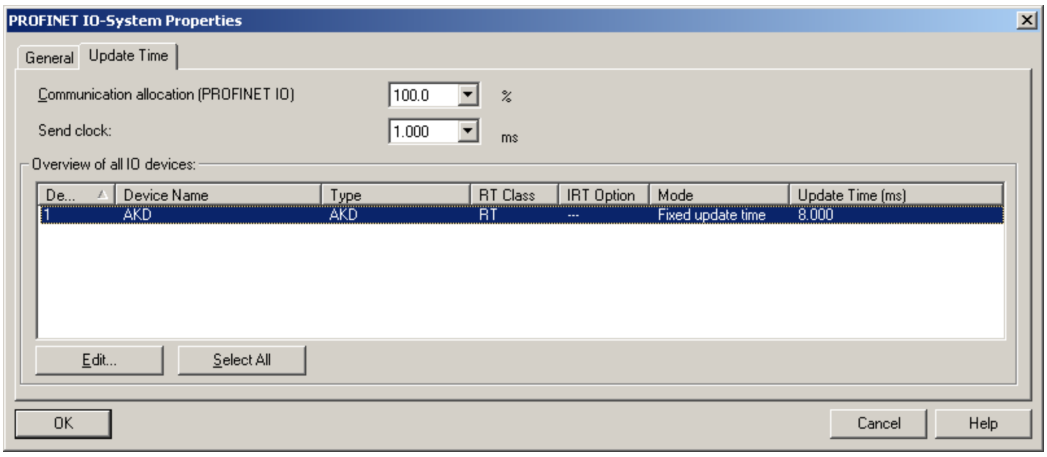
7. Configure the telegram.

   For example, telegram 7 for use in Position mode.

8. Drag and drop telegram 7 into slot 1.



9. Double-click on the PROFINET network which connects PLC and AKD2G and configure the update time.
10. Click **OK**.



11. Save and compile the hardware configuration.

3.6 Parameter Configuration with PROFIdrive over PROFINET IO

The AKD2G is defined as an I/O Device in PROFINET IO. A PLC or I/O-Controller can establish a connection using application relations (AR). Within this AR different profiles such as PROFIdrive, PROFIsafe etc., can be used for the communication. The PROFIdrive profile, which AKD2G supports is defined as Application Process Identifier (API) 0x3A00.

PROFINET IO divides each device in slots and subslots. Sub 0 refers to the device and returns all generic data such as vendor name, software and hardware version. The subslots within the device are used with different real and virtual modules. For example, each module and functional component can be a digital I/O or Telegram with Position values.

AKD2G provides several virtual modules that can be used in Slot 1 and for the real time data exchange.

For read or write parameters to or from the AKD2G the global base mode parameter access is used. The parameter manager is accessed through Slot 1 and a non real time channel must be used. The AKD2G supports the record data 47 and the local Base Mode Parameter Access with the record data object, with index 0xB02E, which is used to address the Parameter numbers (PNUs).

Base mode parameter access provides the construction of the telegram:

Block definition	Byte n+1	Byte n	n
Request header	Request Reference	Request ID	0
	Axis-No. / DO-ID	No. of Parameters = n	2
1st Parameter Address	Attribute	No. of Elements	4
	Parameter Number (PNU)		
	Subindex		
nth Parameter Address	...		4 + 6 x (n-1)
1st Parameter Value(s)	Format	No. of Values	4 + 6 x n
(only for request „Change parameter“)	Values		
	...		
nth Parameter Value(s)	...		
			4 + 6 x n +...+ (Format_n x Qty_n)

The following PROFIdrive services are supported:

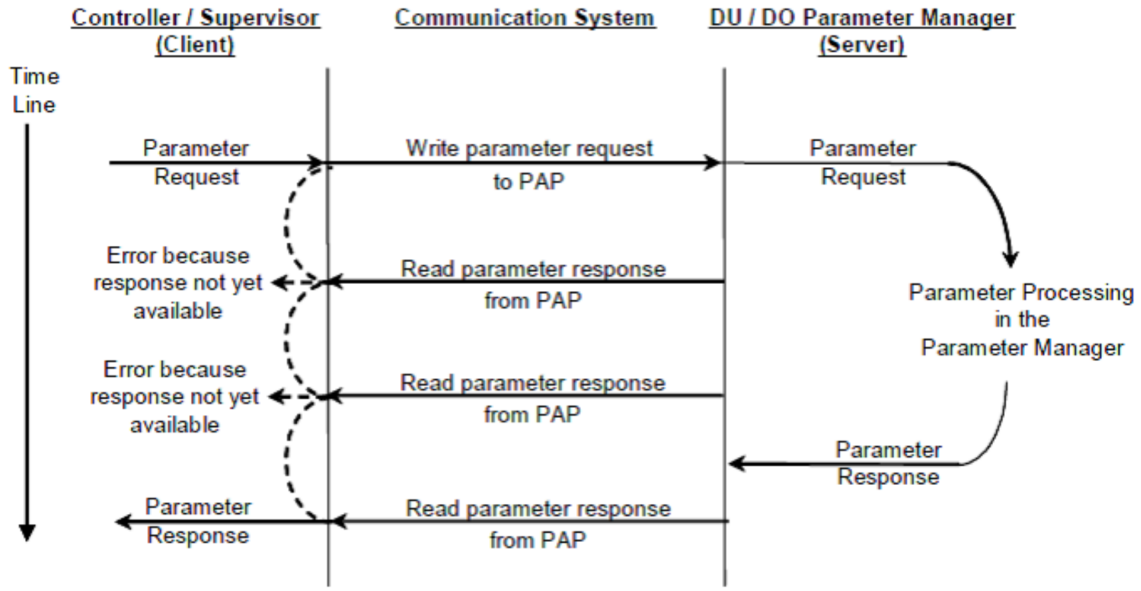
- Single parameter value request
- Multiple parameter value request
- Single parameter change request
- Multiple parameter change request

Record data fields

The table provides the structure and supported fields in the AKD2G for a parameter request.

Field	Data type	Values	Comment
Request Reference	Unsigned8	0x00 reserved 0x01 – 0xFF	
Response ID	Unsigned8	0x01 Request parameter (+) 0x02 Change parameter (+) 0x81 Request parameter (-) 0x82 Change parameter (-)	
Axis/DO-ID	Unsigned8	0x01 .. 0x02	Axis 1 or Axis 2
Number of Para-meters	Unsigned8	0x01.. 0x27	
Attribute	Unsigned8	0x00 reserved 0x10 Value 0x20 Description	
Number of Elements	Unsigned8	0x01.. 0xEA Quantity	
Parameter Number	Unsigned16	0x0001 .. 0xFFFF PNU	
Subindex	Unsigned16	0x0000 .. 0xFFFE	

3.6.1 Parameter Configuration

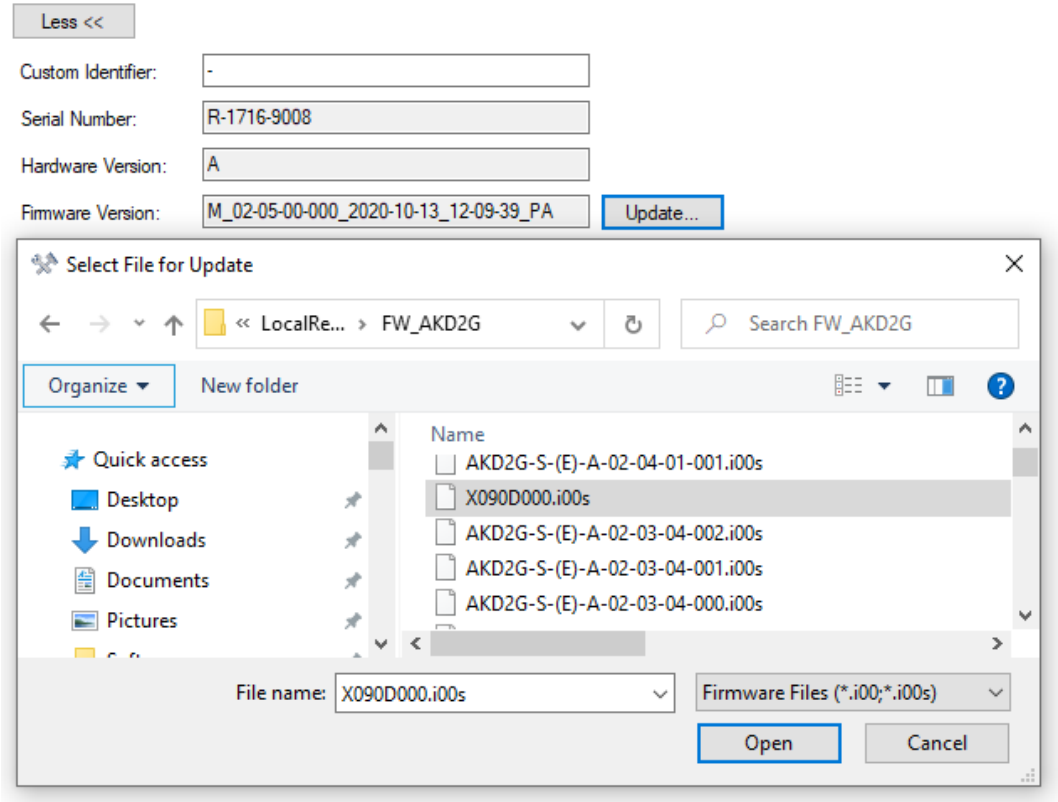


3.7 Use WorkBench to Download PROFINET Firmware

AKD2G allows the PROFINET IO-Device stack (X090D000.i00s) to be upgraded in the field using WorkBench.

3.7.0.1 Example using WorkBench

1. Open WorkBench and go to the Drive Overview screen.
2. Click **Update** and navigate to X090D000.i00s.
3. Select the file and click **Open**.



When opened, the download process begins. The file is transferred to drive, validated, and flashed to onboard System on a Chip (SoC). SoC installs and starts new firmware. The drive re-initializes the PROFINET stack and the download process is complete.

NOTE

The drive does not need to be rebooted after the download.

4 PROFINET IO

4.1 Introduction

PROFINET IO is a real time protocol based on Ethernet. It is used as a high level network for industrial automation applications. PROFINET IO is similar to PROFIbus and focuses on the data exchange for programmable controller.

A PROFINET IO network consists of following devices:

- I/O controller: This is typically the Programmable Logic Controller (PLC) which controls the whole application.
- I/O device: A decentralized I/O device (e.g., drive, encoder, sensor) controlled by the I/O controller.
- I/O supervisor: Human Machine Interface (HMI) or PC for diagnostic purposes or commissioning.

The real time channel (RT) is used for I/O data and alarm mechanism. In PROFINET IO RT (conformance class A and B), the RT data is transferred using a prioritized Ethernet frame. No special hardware is required. Due to this prioritization a cycle time < 10ms can be achieved.

- PROFINET IO IRT is used for higher timing requirements. Cycle times < 1ms is possible, but also specific hardware for I/O devices and switches are required.

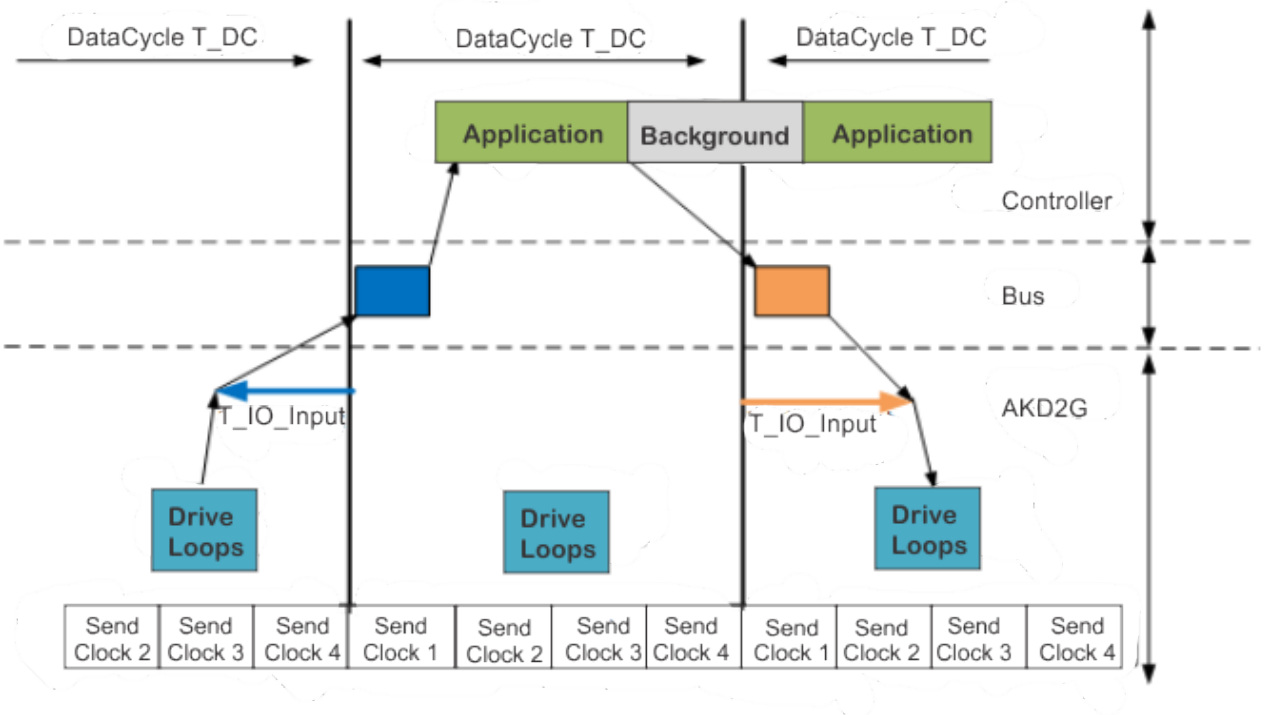
All diagnostic and configuration data is transferred using the non-real time channel (NRT). The common UDP protocol is used for this purpose. Timing determinism cannot be guaranteed and typical cycle times are > 100ms.

4.2 IRT Cycle Timing

When using PROFINET IRT mode the AKD2G synchronizes control loops with the controller. It synchronizes to the fieldbus cycle using a Phase-Locked Loop (PLL) and faults if synchronization is lost. Synchronization can be checked with PROFINET.PLLSTATE.

When the controller establishes a connection to the AKD2G with IRT enabled, it assigns offset times from the fieldbus cycle that the AKD2G must process inputs and outputs. The AKD2G aligns the position loop execution with the processing of drive inputs/Programmable Logic Controller (PLC) outputs (T_IO_Output).

Sequence of IRT Cycle:



Definitions for values in the diagram:

T_DC (Data cycle time)

Data cycle time for IRT period set by control system

T_IO_Input

Time before the start of a data cycle for the acquisition of PLC inputs (AKD2G outputs)

T_IO_Output

Time after the start of a data cycle for PLC outputs to be sent to the AKD2G inputs.

4.3 Restrictions and Requirements

4.3.1 Conformance Classes

AKD2G supports Conformance Classes A, B and C. The PROFIdrive parameters are configured over the PROFINET network, faults are delivered, cyclic data channel functions are available, and the drive is synchronized over the network with IRT.

4.3.2 Cycle Time of RT/IRT Data

The fastest cycle time over RT is 1 millisecond and the fastest IRT cycle is 250 microseconds.

4.3.2.1 250µs Cycle Support

AKD2G supports data cycles up to 250 microseconds; however, due to timing requirements within the drive, the PLC inputs are sent prior to receiving the Programmable Logic Controller (PLC) outputs within a given cycle. Typical timing values setup by the PLC are T_IO_Input time (drive outputs) of 187.5 µs before the end of the cycle and T_IO_Output time (drive inputs) of 125 µs from the beginning of the cycle, which means the drive prepares outputs at 62.5 µs and inputs are processed at 125 µs from the start of the cycle.

4.3.3 Connector

The PROFINET motionbus connectors are located at the top of the drive and labeled as X11 (Port 1) and X12.

5 PROFIdrive over PROFINET IO

5.1 Introduction

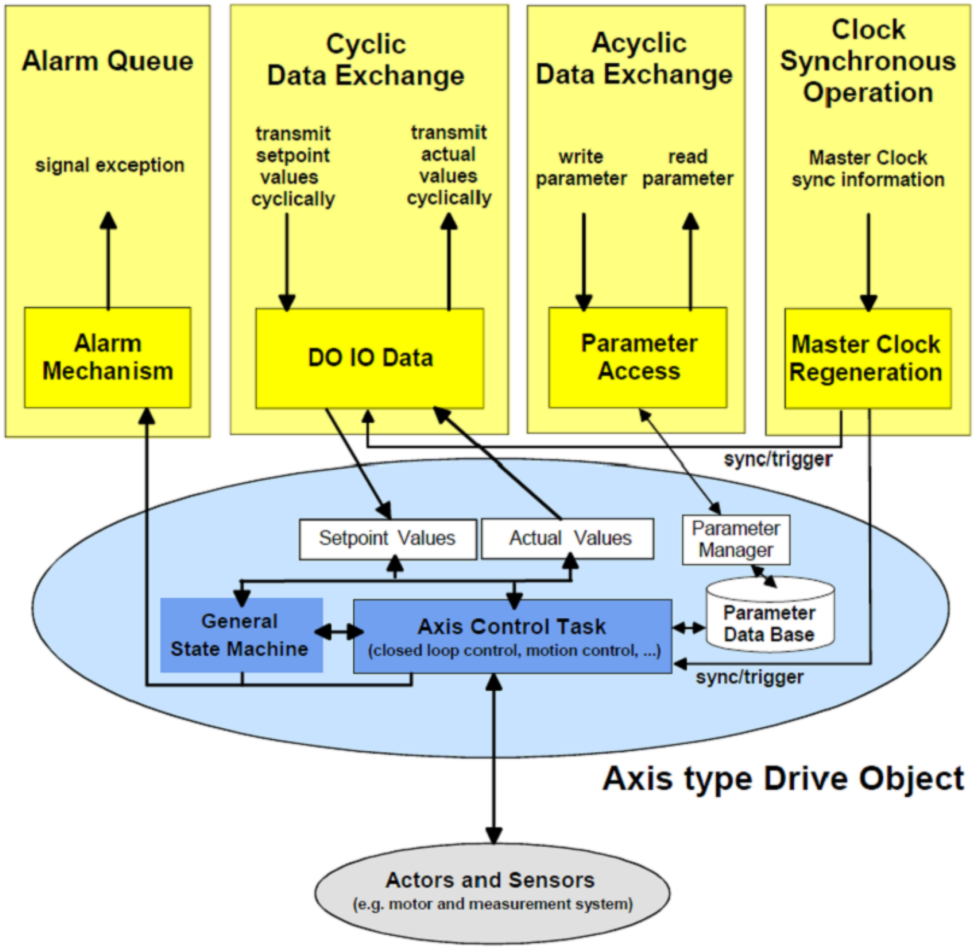
The AKD2G supports the PROFIdrive profile for accessing and configuring standard and manufacturer parameters using PROFINET IO to start, stop and configure motion control tasks.

The profile defines a main element, the Drive Object (DO), which controls the motion related parameters. It is important to understand that PROFIdrive is only a user profile used with PROFINET IO.

NOTE

The AKD2G supports all mandatory functionality of the PROFIdrive profile but not all optional functionality.

5.2 AKD2G as Drive Object (DO)



The AKD2G PROFINET device consists of one drive unit which contains one drive object for every axis of the drive. Each drive object contains the following:

- General state machine
- Axis control task
- Parameter manager with parameter database

Multiple communication channels are used for read/write data values over PROFINET IO. The drive object is accessed using:

- Cyclic data exchange
- Acyclic data exchange

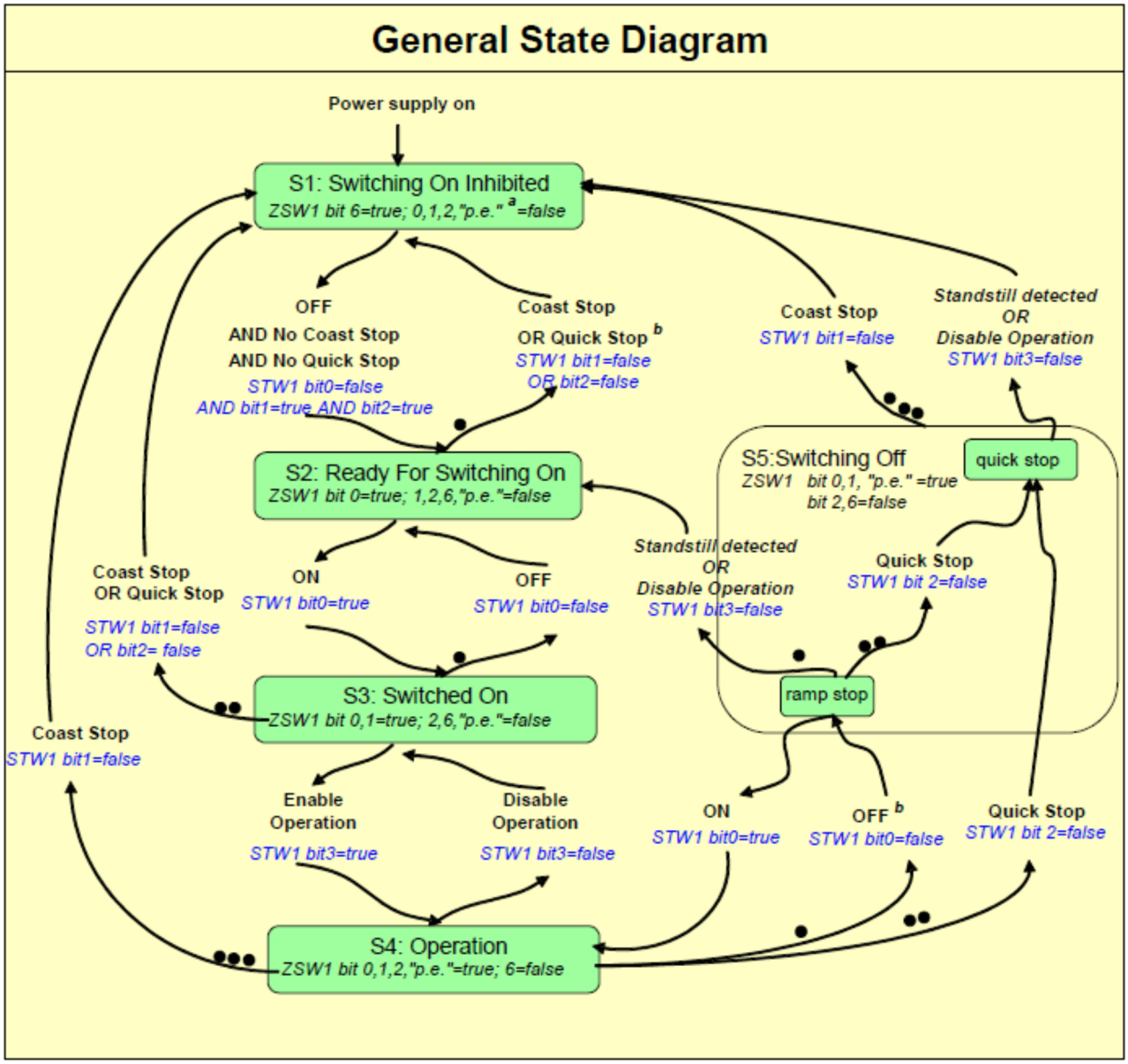
- Fault Buffer
- Clock synchronous operation

The cyclic data exchange includes the transmission and reception of data values like setpoint values (e.g., Position setpoint, velocity setpoint or Controlword) and actual values (actual position value, actual velocity or Statusword) between the master and the drive object. These values are called I/O data and are transferred in real time.

The acyclic data is used for configuring the drive which typically is not time critical. Each DO has its own parameter manager which handles the access. The non real time channel is used for this in PROFINET IO.

The alarm queue is used for signaling the master in exception situations generated through the state machine or the axis control task.

5.3 PROFINET General State Machine



5.3.0.1 State Machine States

State	AKD2G Reaction
S1: Switching On Inhibited	Axis is disabled, bus voltage may or may not be present
S2: Ready For Switching On	Axis is disabled, bus voltage may or may not be present
S3: Switched On	Axis is disabled, bus voltage is present
S4: Operation	Axis is enabled
S51: Ramp Stop	Axis is ramping to a stop following the AXIS#.FBUS.DEC (P5101) rate
S52: Quick Stop	Axis is ramping to a stop following the AXIS#.CS.DEC (P5300) rate
Coast Stop	Coast Stop Axis is disabled and behaves according to AXIS#.DISMODE

5.4 Controlword Bits (STW1)

The S7 application must set the bits in Controlword 1 to go through the PROFIdrive standard state machine to enable mode (complying with the PROFIdrive standard 6.3.2). Bits 0-3 control the state machine state.

Controlword (STW1) defines the following general functions:

General Controlword Bits		
Bit Number	**Description**	**Comment**
0	STW1 on/off	ON/OFF
1	STW1 no coast stop	The axis will not coast stop if this bit is set.
2	STW1 no quick stop	The axis will not execute quick stop if this bit is set. Quick stop executes a controlled stop on the axis.
3	STW1 enable operation	The axis will enable and execute command if all pre-conditions are set.
7	Fault acknowledge	Set this bit to reset faults on the drive.
10	Control by Programmable Logic Controller (PLC)	When not set, no command is accepted from the PLC.

In Velocity mode:

STW1 Special Bits (Velocity mode)		
Bit Number	**Description**	**Comment**
4	Enable ramp generator of the drive	The axis will ramp following AXIS#.FBUS.ACC/AXIS#.FBUS.DEC when new velocity setpoints are received.
5	Unfreeze the ramp generator in the drive	If frozen, the axis stays at current velocity without continuing to ramp up or down. This overrides "Enable setpoint." For example, if the ramp generator is frozen disabling, the setpoint has no effect.
6	Enable setpoint	The axis accepts setpoints from the master. If this bit is not set, the velocity will be 0.

STW1 Special Bits (Velocity mode)		
Bit Number	**Description**	**Comment**
8	Jog 1 on/off	The axis runs up/brakes along the ramp to jogging setpoint 1/standstill. Prerequisite: Operation is enabled, axis is in standstill and STW1 bit 4, 5, 6 = 0. See AXIS#.PROFINET.JOG1V .
9	Jog 2 on/off	The axis runs up/brakes along the ramp to jogging setpoint 2/standstill. Prerequisite: Operation is enabled, axis is in standstill and STW1 bit 4, 5, 6 = 0. See AXIS#.PROFINET.JOG2V .
11-15	Device specific	Not implemented.

Controlword 1 must also set bits 4,5,6 (for speed control – in velocity operation mode) to enable ramp generator and bit 10 to set the axis to be controlled by the PLC.

Bit 7 is used to acknowledge faults. The AKD2G clears faults and automatically changes the S1 state after faults are cleared.

The optional jog bits 8 and 9 can be used for the jogging functionality in Velocity mode.
AXIS#.PROFINET.JOG1V (P5202) and
AXIS#.PROFINET.JOG2V (P5203) define the jogging setpoints 1 and 2. Setting both jog bits 8 and 9 at the same time has no effect, e.g., the drive continues the motion that was already active or does not start any motion if none was active.

In Position mode:

STW1 Special Bits (Position mode)		
Bit Number	**Name**	**Description**
4	Do Not Reject Traversing Task	A traversing task is activated using the positive signal edge at bit 6.
5	No Intermediate Stop	Traversing task can be interrupted and continued.
6	Activate Traversing Task	Positive signal edge enables a traversing task.
8	Jog 1 on/off	The axis runs up/brakes along the ramp to jogging setpoint 1/standstill. Prerequisite: Operation is enabled, axis is in standstill and STW1 bit 4, 5, 6 = 0. See AXIS#.PROFINET.JOG1V .
9	Jog 2 on/off	The axis runs up/brakes along the ramp to jogging setpoint 2/standstill. Prerequisite: Operation is enabled, axis is in standstill and STW1 bit 4, 5, 6 = 0. See AXIS#.PROFINET.JOG2V .
11	Start Homing Procedure	Homing mode is active. If this bit is cleared, the homing is aborted and the drive stops.
12	Real Time Jogging	Jogging data taken from MDI_ACC, MDI_DEC, MDI_ VELOCITY.
13	Real Time Jogging Direction	0: Clockwise 1: Counterclockwise
14-15	Device-specific	Not implemented.

The optional jog bits 8 and 9 can be used for the jogging functionality in Position mode. AXIS#.PROFINET.JOG1V (P5202) and AXIS#.PROFINET.JOG2V (P5203) define the jogging setpoints 1 and 2. Jogging in Position mode uses ramp rates defined by AXIS#.JOG.ACC (P6900) and AXIS#.JOG.DEC (P6901).

5.5 Statusword Bits (ZSW1)

All statusword 1 bits are implemented according to the PROFIdrive standard. All mandatory bits are implemented.

The Statusword (ZSW1) defines the following functions:

General Statusword Bits		
Bit Number	**Description**	**Comment**
0	ZSW1 drive ready to switch on	Ready To Switch On /Not Ready To Switch On.
1	ZSW1 drive ready to operate	Ready To Operate/Not Ready To Operate.
2	ZSW1 operation enabled	Operation Enabled (drive follows velocity setpoint)/Operation Disabled.
3	Fault present	A fault is present on the axis.
4	Coast stop not activated	No coast stop is executed.
5	Quick stop not activated	No quick stop is executed.
6	Switching on inhibited	
7	Warning present	
9	Control requested by the master	When AXIS#.CMDSOURCE = Fieldbus, this bit is set, otherwise it is 0 to indicate to the Programmable Logic Controller (PLC) the axis is ignoring commands.

In Velocity mode:

ZSW1 Special Bits (Velocity mode)		
Bit Number	**Description**	**Comment**
8	Velocity error within range	Difference between target velocity and actual velocity is smaller than tolerance configured in P5200 (default: 10 [RPM]). (AXIS#.PROFINET.VELCOMP)
10	Target velocity reached	Actual velocity is at or above the comparison velocity configured in P5201 (default 1500 [RPM]). (AXIS#.PROFINET.VELTOLERANCE)
11-15	Device specific	Not implemented.

In Position mode:

ZSW1 Special Bits (Position mode)		
Bit Number	**Name**	**Description**
8	Following error in range	Error window. (AXIS#.PL.ERR and AXIS#.PL.ERRFTHRESH)
10	Target position reached	Bit 11 (Motion task target position is reached). (AXIS#.MOTIONSTAT)
11	Home position set	Bit 1 & 2 (Homing finished). (AXIS#.MOTIONSTAT)
12	Traversing Task acknowledgment	On positive edge, traversing task acknowledged or setpoint accepted.
13	Drive stopped	Axis is not moving. (AXIS#.ZEROREACHED)
14	Motion task active	Bit 0 (Motion task active/inactive). (AXIS#.MOTIONSTAT)
15	Device specific	Not implemented

5.6 Controlword/Statusword 2 and Sign of Life (STW2/ZSW2)

Certain telegrams contain an additional 16 bit controlword and statusword. The PROFIdrive profile specifies the use for bits 12-15 for sign-of-life behavior used in IRT, while bits 0-11 are left to the manufacturer and are not used in AKD2G.

STW2 bits

Bit Number	Description	Comment
0-12	Reserved	
12-15	Master sign of life	A 4 bit counter that increments each cycle. The drive uses this to verify synchronization. See sign-of-life description below.

ZSW2 bits

Bit Number	Description	Comment
0-11	Reserved	
11	Pulses enabled	Indicates when the power stage is enabled or not. This is similar to ZSW1 bit 2 (operation enabled) except this bit is set during certain stopping actions when ZSW1 bit 2 may not be.
12-15	Drive sign of life	A 4 bit counter that increments each cycle. The master verifies the counter increments are correct for each cycle. See "Sign of Life" (➜ p. 30) description below.

5.6.1 Sign of Life

The sign of life functionality in STW2/ZSW2 allows the drive and master to monitor the connection to one another and fault if unsynchronized behavior is detected.

The counter from STW2 is monitored to verify it increments each cycle. In each cycle, the counter is checked against an expected value. If it does not match, an error counter is incremented by 10. If it matches, the error counter is decremented by 1. The error counter cannot be less than 0.

If the error counter is exceeds 10 * max sign of life errors AXIS#.PROFINET.MAXSOLFAIL (P925) then a fault is triggered.

By default the sign of life threshold is set to 1, meaning it tolerates one failure, but if another failure occurs before the error counter decrements back to 0 a fault occurs.

The current error count can be monitored using AXIS#.PROFINET.SOLFAILCNT.

5.7 Supported Profile PNUs

The following table describes all supported PROFIdrive specific parameters. Accessing these parameters must be done by using base mode parameter access as described in Parameter Configuration with PROFIdrive over PROFINET IO.

PNU	Name	Data Type	Description
915	DO I/O Data configuring (setpoint telegram)	Array of U16	Array of signal numbers mapped into the setpoint telegram (AXIS#.PROFINET.SETPOINTMAP).
916	DO I/O Data configuring (actual value telegram)	Array of U16	Array of signal numbers mapped into the actual value telegram (AXIS#.PROFINET.ACTUALVALUEMAP).
922	Telegram selection	U16	The PROFIdrive telegram used for the I/O connection configuration.
923	List of all parameters for signals	Array of U16	All supported signals and their corresponding PNUs.
924	Statusword bit Pulses Enabled	Array[2] of U16	Bit position of pulses enabled bit, sub-index 0 returns the signal number that contains the pulses enabled bit (ZSW2 for AKD2G) and sub-index 1 returns the bit position (bit 11 for AKD2G).
925	Sign of life tolerance	U16	Number of consecutive errors in the master's sign of life (STW2 bits 12-15) that will be tolerated before creating a fault. (Defaults to 1).
928	Control priority	U16	Indicates if Programmable Logic Controller (PLC) has priority, always a 1 on AKD2G.
930	Operating mode	U16	The operation mode of the drive: 1 – velocity control, 2 – positioning mode, 3 – reduced velocity (bit 5 of STW1 is unused, otherwise behaves the same as 1 for AKD2G), 0x8000 – manufacturer specific torque mode.
944	Fault message counter	U16	See Fault Buffer.
947	Fault number	Array of U16	All active faults. See Fault Buffer.
948	Fault timestamps	Array of U32	Timestamps of active faults. See Fault Buffer.
951	Fault descriptions	Array of strings	Description for all fault codes. See Fault Buffer.
952	Fault situation counter	U16	See Fault Buffer.
964	Drive Unit Identification	Array of U16	Provides 5 entries: manufacturer id (261), unit type (2 for AKD2G), version number, year, month, and number of DO supported.

PNU	Name	Data Type	Description
965	Profile identification number	Array of U8	Returns PROFIdrive profile number (3) for sub-index 0 and profile version (42) for sub-index 1.
967	STW1	U16	Controlword
968	ZSW1	U16	Statusword
974	Parameter access service id	Array of U16	Specifies the max supported length (1024 bytes) and number of parameters (64) for multi-parameter accesses.
975	DO identification	Array of U16	Provides 7 entries: manufacturer id (261), object type (1 for axis type), version, year, month, DO type (1 for axis type), DO sub class, properties (bits 0, 2, and 3 for AC1, 3, and 4).
979	Sensor Format (P979)	Array of U16	Description of sensor 1.
980 to 989	Number list of defined parameter	Array of U16	Returns the list of PNUs supported.
60000	Reference velocity	Float	Reference velocity used for N2/N4 units of cyclic velocity commands (see Units) (AXIS#.PROFINET.REFV).

For more information, see Appendix A: Manufacturers' PNUs.

5.8 Sensor Format (P979)

979 allows reading the configuration of the sensor that is accessed by the signals G1_XIST1, G1_XIST2, G1_STW and G1_ZSW. The AKD2G only supports reading one sensor per axis and the specified sensor is always the feedback for that axis which is selected by AXIS#.PL.FBSOURCE.

The information in 979 is organized in multiple subindices:

5.8.1 Subindex 0: Header

Contains static data about the version of the sensor interface (1.2), the number of sensors described (1), and the number of subindices used for describing each sensor (5).

5.8.2 Subindex 1: Sensor Type

Contains static information that communicates to a master that some of the information in 979 is not static and can change during operation. Also contains a bit indicating whether the feedback is rotary or linear, which is set based on the motor type used on that axis.

5.8.3 Subindex 2: Sensor Resolution

This subindex contains information about how positions in G1_XIST1 and G1_XIST2 are scaled. Since the drive functions as an abstraction layer between the actual feedback device and the fieldbus for the PROFIdrive sensor interface, every feedback type is treated as digital only even if the feedback device provides an analog signal. As a result, the drive always gives the position in the same scaling.

For rotary encoders this subindex contains the number of counts per revolution (2^20).

For linear encoders it contains the distance between two position increments in nanometers, which is AXIS#.MOTOR.PITCH converted to nanometers and divided by 2^20. Since this is an integer value, some inaccuracy is introduced because the calculation requires rounding if the motor pitch is not divisible by 2^20.

5.8.4 Subindex 3 and 4: “Shift factors” for G1_XIST1 and G1_XIST2

These indices contain information about how many bits of the positions contain “quadrant information” and “fine resolution” obtained from an analog feedback signal. As provided in the description of subindex 2, the AKD2G recognizes all feedback types as only digital for the purpose of the PROFIdrive sensor interface; therefore, these subindices will always contain zero.

5.8.5 Subindex 5: Determinable Revolutions

The number of determinable revolutions in the sensor position: Zero for non-absolute feedbacks and one for absolute linear feedbacks. Absolute rotary feedbacks returns 2^12, as the positions in G1_XIST1 and G1_XIST2 have 32 bits, and 20 bits are used for the position inside one revolution.

5.9 Sensor Interface

PROFIdrive provides an interface to the drive’s position feedback by using the sensor interface.

The AKD2G implements the sensor interface state machine as it is provided in the PROFIdrive specification, including the mandatory states and optional ‘measurement on the fly’ function.

The AKD2G drive supports one sensor that is accessed by using the signals G1_STW, G1_ZSW, G1_XIST1 and G1_XIST2.

Signal	Description
G1_STW	Sensor 1 Controlword
G1_ZSW	Sensor 1 Statusword
G1_XIST1	Sensor 1 position, AXIS#.PL.FB scaled to 20 bits per revolution
G1_XIST2	Sensor 1 position 2, one of multiple functions: • 0 while in state "parking". • AXIS#.PL.FB is scaled to 20 bits per revolution in states "normal operation"or "wait for measured values" if absolute position in XIST2 is requested by Controlword bit 13. • An error code in state "error". • The requested measured position in state “"measured value in Gx_XIST2".

5.9.1 Sensor Controlword

The sensor Controlword defines the following functions:

Bit Number	Description	Comment
0	Function select	Probe 1 positive edge
1		Probe 1 negative edge
2		Probe 2 positive edge
3		Probe 2 negative edge
4	Command	1: Activate functions
5		2: Read value
6		3: Cancel functions
7	Mode	0: Reference mark search (not supported by AKD2G) 1: Measurement on the fly
8	Reserved	
9	Reserved	
10	Reserved	
11	Home Position mode	Not supported on AKD2G
12	Request set/shift home position	Not supported on AKD2G
13	Request absolute value cyclically	
14	Activate parking sensor	
15	Error acknowledge	

5.9.2 Sensor Statusword

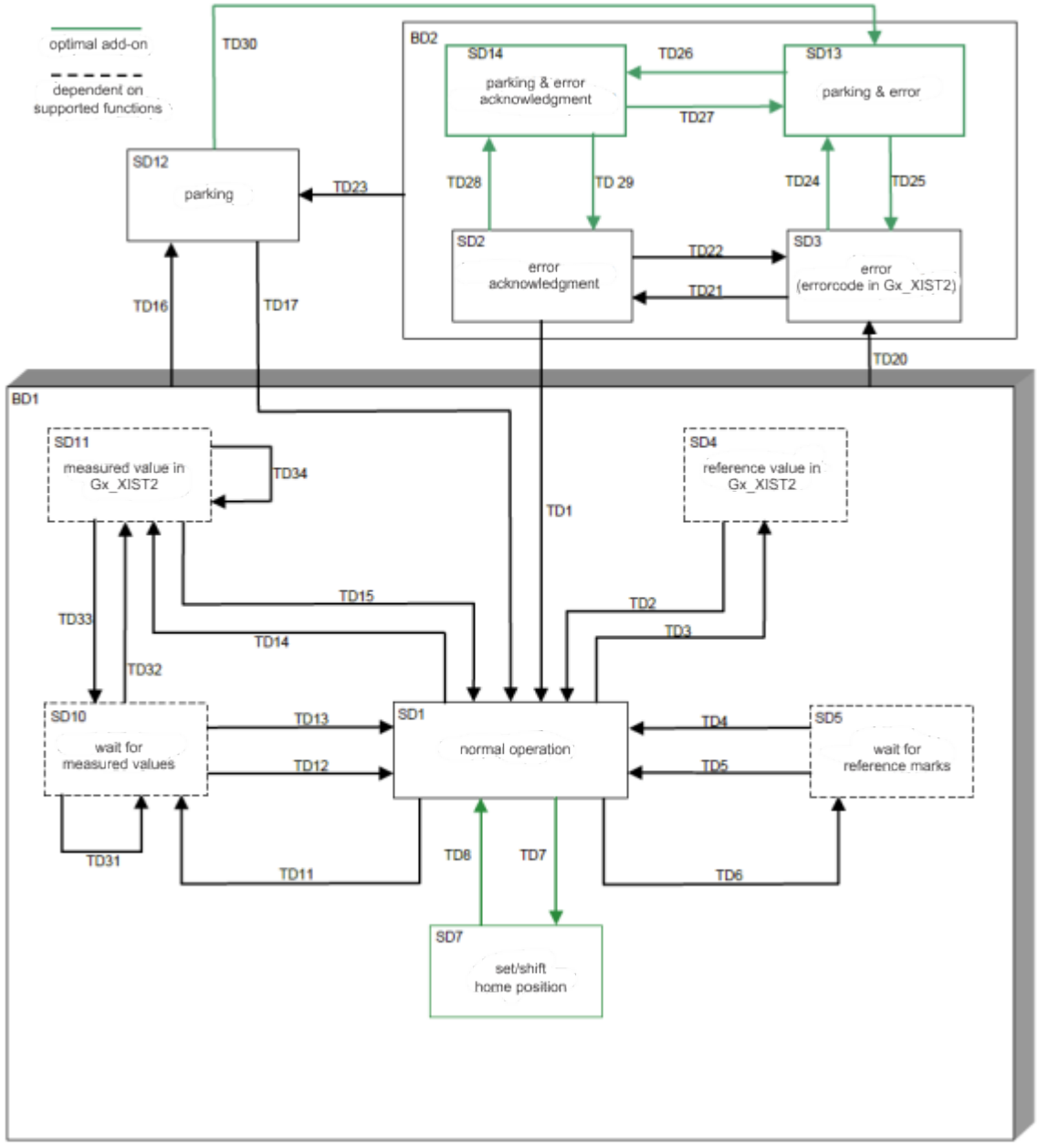
The sensor Statusword defines the following functions:

Bit Number	Description	Comment
0	Function status	Probe 1 positive edge active
1		Probe 1 negative edge active
2		Probe 2 positive edge active
3		Probe 2 negative edge active
4		Probe 1 positive edge value available
5		Probe 1 negative edge value available
6		Probe 2 positive edge value available
7		Probe 2 negative edge value available
8	Probe	Set when digital input 1 is high

Bit Number	Description	Comment
9	Probe 2 deflected	Set when digital input 2 is high
10	Reserved	
11	Error acknowledgment in progress	
12	Set/shift home position executed	Not supported on AKD2G
13	Transmit absolute value cyclically	
14	Parking sensor active	
15	Sensor error	

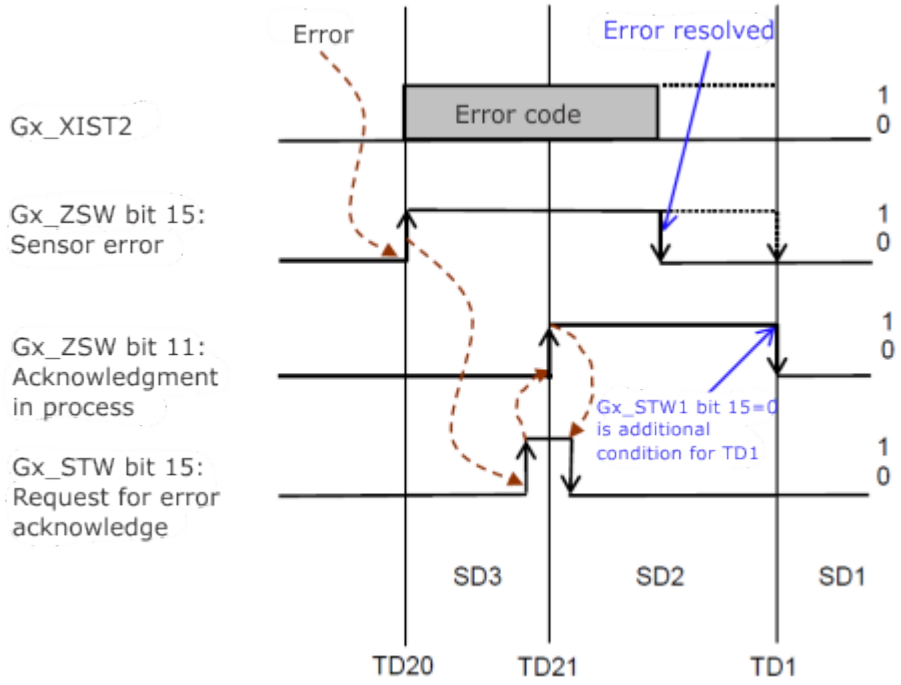
5.9.3 Sensor State Machine

This is the sensor state machine as specified by PROFIdrive. The states SD4, SD5, SD7, SD13 and SD14 are not supported by AKD2G.



5.9.4 Sensor Errors

When an error occurs, e.g., when trying to activate the unsupported reference mark feature, the sensor state machine transitions into the state "error". This is indicated by bit 15 of the sensor Statusword. While in this state, G1_XIST2 contains an error code. To acknowledge the error, bit 15 of the Controlword must be set first to transition the state machine to the “error acknowledgment” state, and then cleared again to return to “normal operation”.



5.9.5 Measurement on the Fly Mode

The AKD2G supports two probes, monitoring digital inputs 1 and 2, each with a positive and negative edge.

To activate a probe:

1. Set bit 7 of the sensor Controlword to select the “Measurement on the Fly” mode.
2. Select the probes/edges to be activated by setting bits 0 to 3 in the sensor Controlword.
3. Activate the selected probes by setting the sensor Controlword’s bit 4 to 1 and, bits 5 through 6 to 0.

The state machine transitions to the state “wait for measured values” and indicates active probes in bits 0 to 3 of the sensor Statusword.

Additional probes are activated by:

1. Set the sensor Controlword bits 4 through 6 to 0.
2. Set additional probes by setting bits 0 to 3.
3. Activate by setting the sensor Controlword’s bits 4 to 1, and bits 5 through 6 to 0.

To read a value measured by a probe:

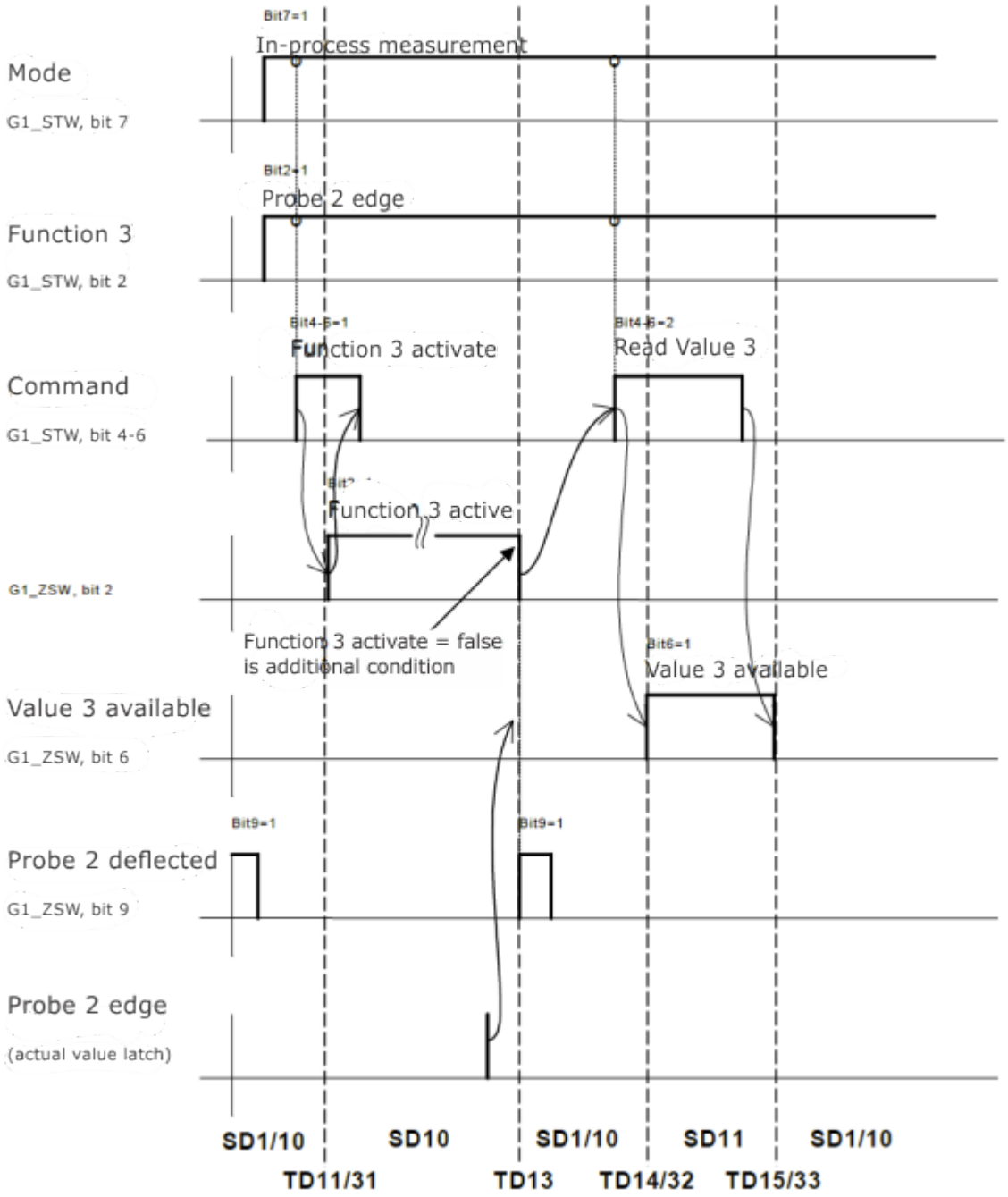
1. Select a single probe/edge by setting one of the bits 0 to 3 in the sensor Controlword.
2. Set bits 4, 6 and 7 of the sensor Controlword to 1 and bit 6 to 0.

If a measured value is available, i.e., the selected probe/edge was activated and is no longer active. The state machine transitions to the state “measured value in Gx_XIST2” and the measured value is read from G1_ XIST2. Otherwise the state machine transitions to the error state.

NOTE

All active probes are canceled by setting bits 4 to 6 of the sensor Controlword to 1.

Example: Activating function 3 (probe 2 positive edge) and reading measured position:



5.10 Telegram Configuration

The telegram configuration is set according to the PROFIdrive standard. The PROFIdrive parameters used in the configuration are: AXIS#.PROFINET.TELEGRAMID (P922), AXIS#.PROFINET.SIGNALLIST (P923), AXIS#.PROFINET.SETPOINTMAP (P915), AXIS#.PROFINET.ACTUALVALUEMAP (P916) See PROFIdrive profile.

Telegram selections are made by configuring the correct sub-module in the Programmable Logic Controller (PLC) configuration or by selecting telegrams using AXIS#.PROFINET.TELEGRAMID (P922). By default, the drive uses the PLC configuration when the PLC connects to the drive. The PLC configuration will override the telegram signal mapping when not using telegram 0 free mapping.

PROFINET.USEMODULECFG (P5211) can be set to 0 to have the drive ignore the PLC configuration exchanged during connection establishment. When set to 0, the drive starts up with the telegram specified in AXIS#.PROFINET.TELEGRAMID (P922) regardless of what the sub-module configuration is in the PLC.

When telegram 0 is used, AXIS#.PROFINET.SETPOINTMAP (P915) and AXIS#.PROFINET.ACTUALVALUEMAP (P916) are writable. The PROFIdrive state machine must be in the switched on inhibited state (AXIS#.PROFINET.STATE = S1) to make changes to the signal mapping.

5.10.1 Telegram Mappings

5.10.1.1 Telegram 0

Telegram 0 is used for the free mapping of PROFIdrive signals into the PROFINET IO data. AXIS#.PROFINET.SETPOINTMAP (P915) defines then setpoint signals and AXIS#.PROFINET.ACTUALVALUEMAP (P916) the actual value signals. These parameters are saved to NV so the drive retains the mappings when rebooted.

Each direction signal mapping supports up to 16 words. The signals that can be configured are provided with AXIS#.PROFINET.SIGNALLIST (P923).

NOTE

If the telegram is changed from telegram 0 to a different one, the signal maps are overwritten to match the selected telegram. Additionally, if a save operation is subsequently performed the saved mapping gets overwritten.

5.10.1.2 Standard Telegram 1

Typically used for application class 1 (Velocity mode). The setpoint velocity value can be directly controlled by an PROFINET master.

I/O Data Number	Setpoint	Actual Values
1	STW1	ZSW1
2	NSOLL_A	NIST_A

5.10.1.3 Standard Telegram 3

32-bit speed setpoint interface with one sensor position feedback.

I/O Data Number	Setpoint	Actual Values
1	STW1	ZSW1
2	NSOLL_B	NIST_B
3		

I/O Data Number	Setpoint	Actual Values
4	G1_STW	ZSW2
5		G1_ZSW
6		G1_XIST1
7		
8		G1_XIST1
9		

5.10.1.4 Standard Telegram 5

32-bit speed setpoint interface with one sensor position feedback and additionally, position error and position gains to enable Dynamic Servo Control (DSC).

I/O Data Number	Setpoint	Actual Values
1	STW1	ZSW1
2	NSOLL_B	NIST_B
3		
4	G1_STW	ZSW2
5	XERR	G1_ZSW
6		G1_XIST1
7	KPC	
8		G1_XIST1
9		

5.10.1.5 Standard Telegram 7

Typically used for application class 3 (Position mode). Predefined motion tasks can be selected using I/O data.

I/O Data Number	Setpoint	Actual values
1	STW1	ZSW1
2	SATZANW	AKTSATZ

5.10.1.6 Standard Telegram 9

Typically used for application class 3 (Position mode). A motion task can be configured using I/O data.

I/O Data Number	Setpoint	Actual Values
1	STW1	ZSW1
2	SATZANW	AKTSATZ
3	STW2	ZSW2
4	MDI_TARPOS	XIST_A
5		XIST_A

I/O Data Number	Setpoint	Actual Values
6	MDI_VELOCITY	
7		
8	MDI_ACC	
9	MDI_DEC	
10	MDI_MOD	

5.10.1.7 Telegrams 970-973

These telegrams add additional user selectable signals to the PROFINET IO data appended after the Standard Telegram data. This telegram is inserted into subslot 3. When not configured, subslot 3 should be occupied by empty submodule (id=5000). AXIS#.PROFINET.SUPPTELEGRAMID is used to set or view currently configured supplementary telegram. The setpoint and actual value mappings for the supplemental telegram is changed by using the associated keyword or PNU similar to standard telegram 0.

Telegram Id	Setpoint Mapping Parameter	Setpoint Mapping PNU	Actual Value Mapping Parameter	Actual Value Mapping PNU	Description
5000	N/A	N/A	N/A	N/A	Default empty telegram for no supplemental data
970	AXIS#.PROFINET.SETPOINTMAP	P5213	AXIS#.PROFINET.ACTUALVALUEMAP	P5214	Supports 2 signals for both setpoint and actual value
971	AXIS#.PROFINET.SETPOINTMAP	P5215	AXIS#.PROFINET.ACTUALVALUEMAP	P5216	Supports 2 signals for setpoints and 4 for actual values
972	AXIS#.PROFINET.972SETPOINTMAP	P5217	AXIS#.PROFINET.ACTUALVALUEMAP	P5218	Supports 2 signals for setpoints and 6 for actual values

Telegram Id	Setpoint Mapping Parameter	Setpoint Mapping PNU	Actual Value Mapping Parameter	Actual Value Mapping PNU	Description
973	AXIS#.PROFINET.SETPOINTMAP	P5219	AXIS#.PROFINET.ACTUALVALUEMAP	P5220	Supports 8 signals for setpoints and 8 for actual values

The supplmental telegram data can be viewed with AXIS#.PROFINET.SUPPACTUALVALUEDATA (P5224) and AXIS#.PROFINET.SUPPSETPOINTDATA (P5225).

5.10.2 PROFIdrive Signals

The following PROFIdrive signals are permitted:

Signal Number	Signal Format	PROFIdrive Signal Name	AKD2G Signal	Description
1	WORD	STW1	AXIS#.PROFINET.STW1	Controlword 1 can only be mapped at index 0 of AXIS#.PROFINET.SETPOINTMAP (P915)
2	WORD	ZSW1	AXIS#.PROFINET.ZSW1	Statusword 1 can only be mapped at index 0 of AXIS#.PROFINET.ACTUALVALUEMAP (P916)
3	WORD	STW2	AXIS#.PROFINET.STW2	Controlword 2
4	WORD	ZSW2	AXIS#.PROFINET.ZSW2	Statusword 2
5	WORD	NSOLL_A	AXIS#.VL.CMD	16-bit target velocity
6	WORD	NIST_A	AXIS#.VL.FB	16-bit actual velocity
7	DWORD	NSOLL_B	AXIS#.FBUS.VL.CMD	32-bit target velocity
8	DWORD	NIST_B	AXIS#.VL.FB	32-bit actual velocity
9	WORD	G1_STW	AXIS#.PROFINET.G1STW	Sensor 1 Controlword
10	WORD	G1_ZSW	AXIS#.PROFINET.G1ZSW	Sensor 1 Statusword
11	DWORD	G1_XIST1	FB#.P	32-bit position value with 12 bits multiturn
12	DWORD	G1_XIST2	FB#.P	32-bit position value with 12 bits multiturn or error code

Signal Number	Signal Format	PROFIdrive Signal Name	AKD2G Signal	Description
21	WORD	E_ DIGITAL	DIN.STATES	
22	WORD	A_ DIGITAL	AXIS#.PROFINET.ADIGITAL	Digital output fieldbus states
23	WORD	E_ ANALOG	AIN1.VALUE	Scaling: mV
24	WORD	A_ ANALOG	AOUT1.VALUE	Analog output 1 fieldbus value, scaling: mV
25	WORD	XERR	AXIS#.PROFINET.XERR	System deviation/position error
26	WORD	KPC	AXIS#.PROFINET.KPC	Position controller gain
28	DWORD	XISTA	AXIS#.PL.FB	32-bit position value scaled according to position units
32	WORD	SATZANW	AXIS#.PROFINET.SATZANW	Position mode Controlword (MT select/MDI mode)
33	WORD	AKTSATZ	AXIS#.PROFINET.AKTSATZ	Position mode Statusword
34	DWORD	MDI_ TARPOS	AXIS#.PROFINET.MDITARPOS	32-bit target position
35	DWORD	MDI_ VELOCITY	AXIS#.PROFINET.MDIVELOCITY	32-bit target velocity
36	WORD	MDI_ACC	AXIS#.PROFINET.MDIACC	Acceleration
37	WORD	MDI_DEC	AXIS#.PROFINET.MDIDEC	Deceleration
38	WORD	MDI_MOD	AXIS#.PROFINET.MDIMOD	MDI mode. Bit 0: 0: Relative Positioning 1: Absolute Positioning
5025	DWORD		AXIS#.MOTIONSTAT	
5100	WORD		AXIS#.FBUS.ACC	Similar to MDIACC but updates keyword directly regardless of controlword.
5101	WORD		AXIS#.FBUS.DEC	Similar to MDIACC but updates keyword directly regardless of controlword.
5104	WORD		AXIS#.FBUS.IL.CMD	
5803	WORD		AXIS#.IL.FB	
5829	WORD		AXIS#.IL.LIMITP/N	
5901	DWORD		AXIS#.PL.ERR	

Signal Number	Signal Format	PROFIdrive Signal Name	AKD2G Signal	Description
6104	DWORD		AXIS#.HOME.DIST	
6107	WORD		AXIS.HOME.MODE	
6311	WORD		AXIS#.MT.FEEDRATE	Scaling: 100% per 65535 counts
10008	DWORD		AXIS#.SAFE.FUNCTIONSTATUS	
23101	WORD		AIN2.VALUE	Scaling: mV
24151	WORD		AOUT2.VALUE	Analog output 2 fieldbus value, scaling: mV

NOTE

The PROFIdrive signals are not mapped 1:1 to the AKD2G signals. The unit conversion for PROFIdrive must be used. For more information, see Units.

The predefined standard telegrams are used for accessing the signals or a free mapping is used with telegram 0.

5.10.3 Telegram 0 Custom Mapping

5.10.3.1 Default Mapping

Telegram 0 is the only free mappable telegram available in the AKD2G. The default mapping of Telegram 0 consists of the prior mapping of the last standard telegram to be selected. For example, if telegram 1 is selected first and telegram 0 is selected next, telegram 0 will have the same mapping as telegram 1.

Telegram 0 mapping can be customized with any available signals.

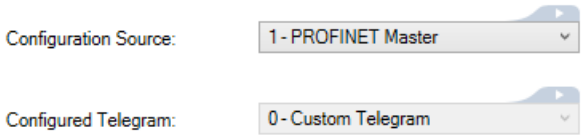
If a standard telegram has most of the signals needed in telegram 0, then select that standard telegram so the default mapping of telegram 0 starts with those signals.

5.10.3.2 Custom Mapping Telegram 0

Use the following steps to custom map telegram 0.

1. Using WorkBench, open the Telegram view.
2. Go to Device Settings > Communications > PROFINET > Axis 1 or Axis 2.
3. Use one of these ways to configure Telegram 0:

   a. Set the Configuration Source = “1 – PROFINET Master” and configure telegram 0 in the Programmable Logic Controller (PLC). The Configured Telegram = “0 – Custom Telegram” is grayed out because it is controlled by the master (PLC).

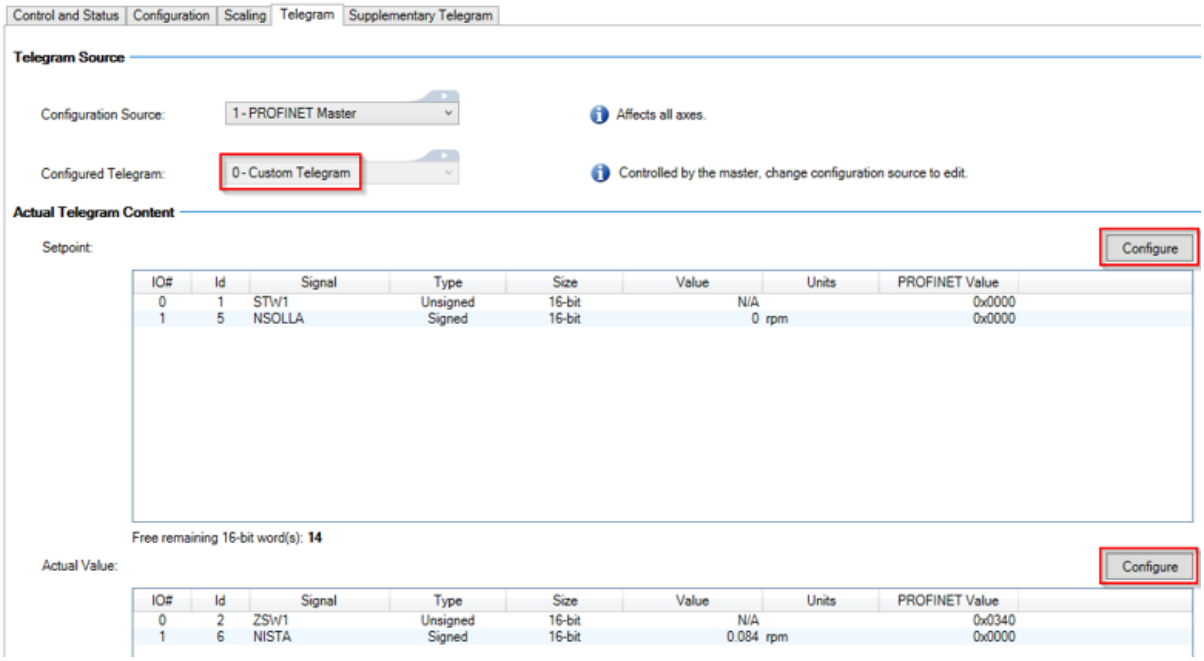


   b. Set the Configuration Source = “0 – Telegram ID” and set the Configured Telegram = “0 – Custom Telegram”.:

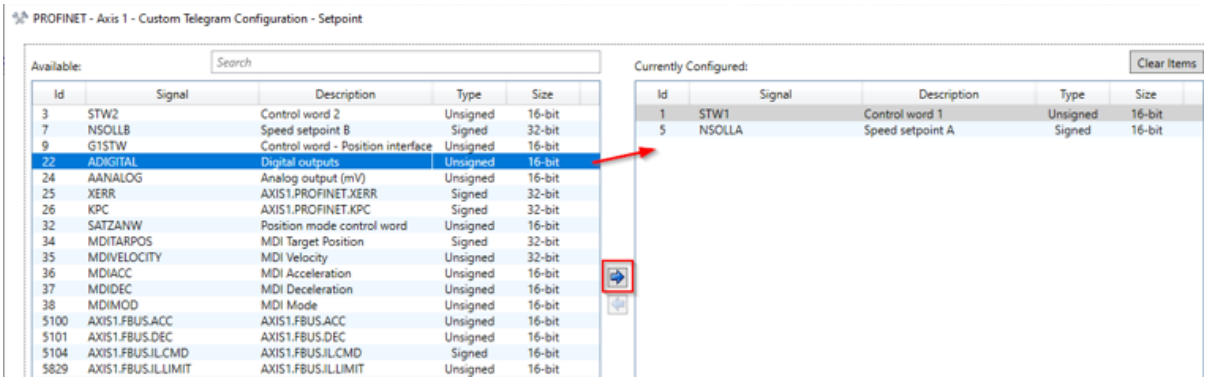
NOTE

It is recommended to configure the telegram in the PLC using option a) Configuration Source = "1 – PROFINET Master" so the PLC always has control over the telegram used. If you select the telegram in the drive, option b) Configuration Source = "0 – Telegram ID" then the PLC will not automatically set the used telegram in the drive and could result in a mismatch.

4. Click **Configure** to open the Custom Telegram Configuration dialog.



5. Highlight the signals under Available, on the left, and use the arrow to move them to the right for telegram configuration.



6. Click **OK**.

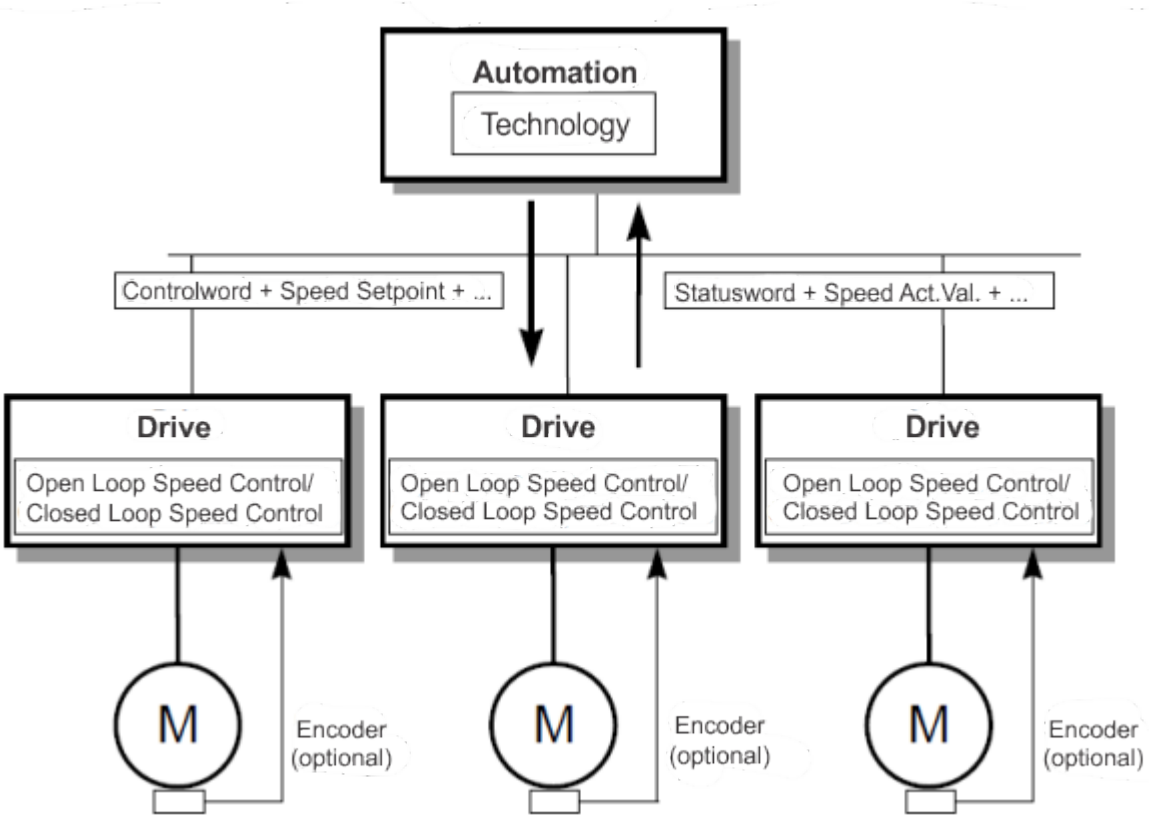
5.11 Velocity Mode

In Velocity Mode the drive is controlled using a primary setpoint (speed setpoint). The speed control is in the drive controller and there may or may not be a position loop in the Programmable Logic Controller (PLC).

Two application classes operate in Velocity mode: class 1 and class 4. In class 1, there is no synchronous operation and only velocity command/feedbacks are used for velocity loop control. Class 4 adds synchronization (using IRT) and position feedback to enable position loop control from the PLC.

5.11.1 Application Class 1

For class 1, the drive is usually configured with an RT communication cycle using telegram 1 or 2 with only a velocity command and feedback present. The drive updates the axis' internal velocity command input to the velocity loop to the velocity commanded by the fieldbus following a specified ramp rate (AXIS#.FBUS.ACC (P5100) and AXIS#.FBUS.DEC (P5101).



5.11.1.1 Application Class 1 Example

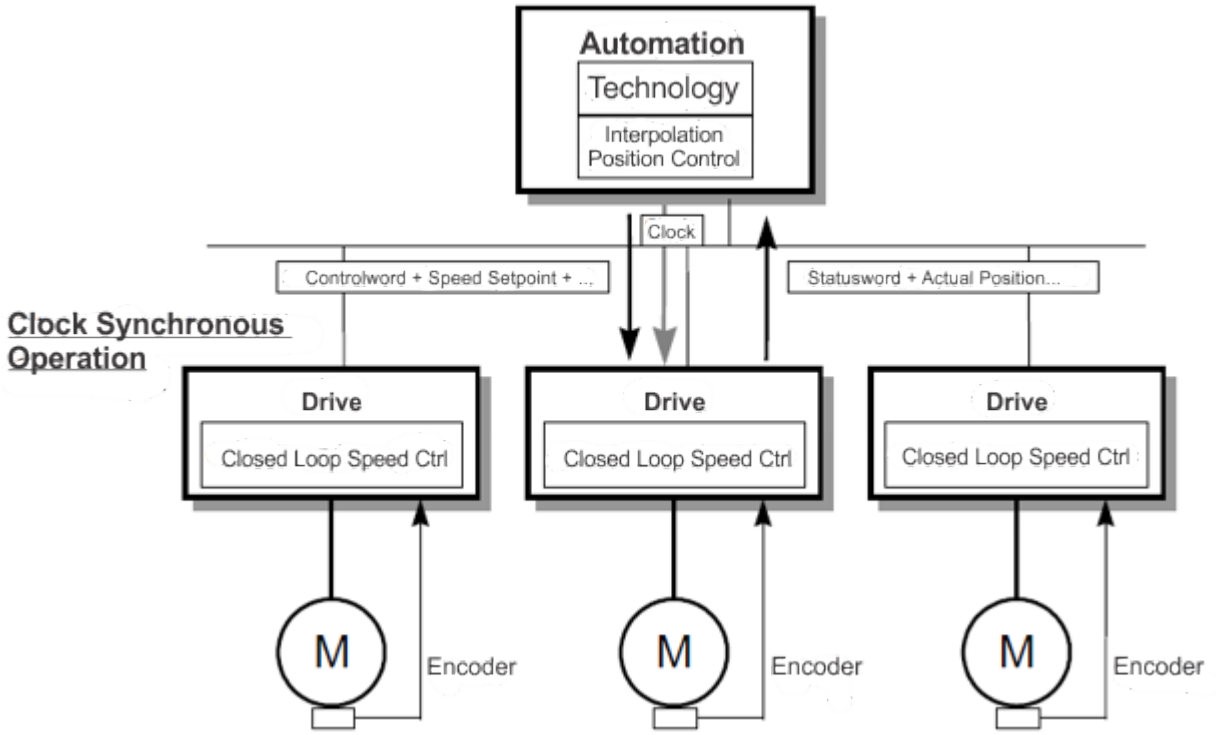
This example demonstrates enabling the drive and executing motion in Velocity mode application class 1 using standard telegram 1. This means the PLC needs to send 32 bits (16 Controlword and 16 velocity command) and read back 32 bits (16 Statusword and 16 velocity feedback)

- Send Controlword bits as follows to move the state machine to S1: 0000_0100_0111_0000. Velocity command can be zero (it is ignored at this phase)
- Send Controlword bits as follows to move the state machine to S2: 0000_0100_0111_0110. Velocity command can be zero (it is ignored at this phase)
- Send Controlword bits as follows to move the state machine to S3: 0000_0100_0111_0111. Velocity command can be zero (it is ignored at this phase)

- Send Controlword bits as follows to move the state machine to S4 and enable the drive: 0000_0100_0111_1111. Velocity command is used now, set it to 0x00A3 (30 rpm - see Cyclic Velocity Units).

5.11.2 Application Class 4

In this mode the drive is configured with IRT communication to synchronize with the controller and uses telegram 3 or 5. The axis still operates in Velocity mode and follows the ramp rate defined by AXIS#.FBUS.ACC (P5100) and AXIS#.FBUS.DEC (P5101) the same as application class 1. Using telegram 3 adds a position feedback signal to the PLC in order for the PLC to close the position loop.

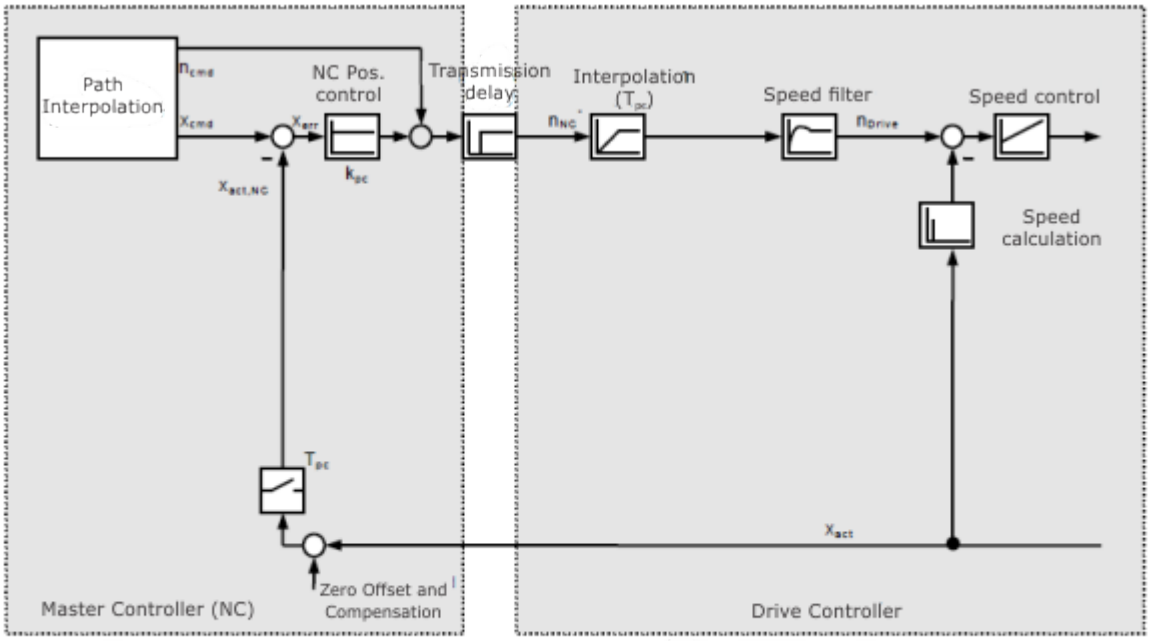


5.11.2.1 Dynamic Servo Control (DSC)

DSC is enabled automatically in the drive when the signals XERR (position error) and KPC are configured by using standard telegram 5 or when using telegram 0. This sub-mode allows the drive to calculate the position command from the PLC and interpolate between fieldbus updates to arrive to that position. Higher controller position gain settings and a higher system bandwidth is obtained using this method.

See the figures below for diagrams comparing operation with and without DSC.

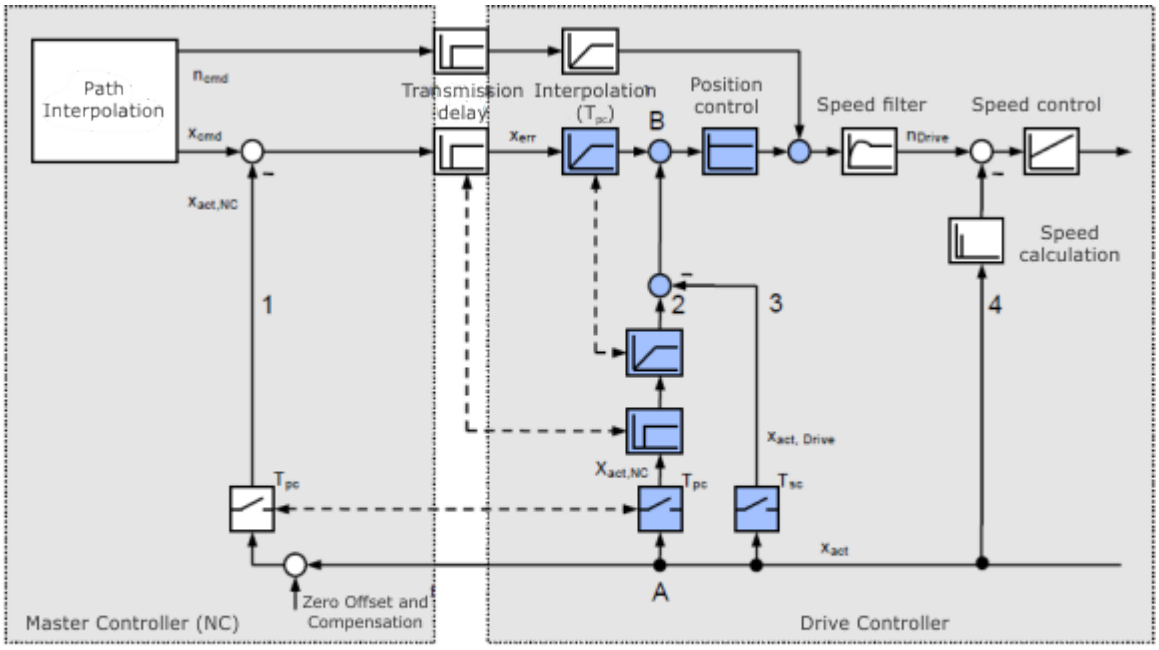
Velocity setpoint without DSC



Key: n_{cmd}: speed command
x_{cmd}: position command
x_{err}: position command
x_{act}: actual position

T_{pc}: position controller sampling time (= T_{calc})
k_{pc}: position controller gain

Velocity setpoint with DSC



Key: n_{cmd}: speed command
x_{cmd}: position command
x_{err}: position command
x_{act}: actual position

T_{sc}: speed controller sampling time
T_{pc}: position controller sampling time (= T_{calc})
k_{pc}: position controller gain

5.12 Position Mode (Application class 3)

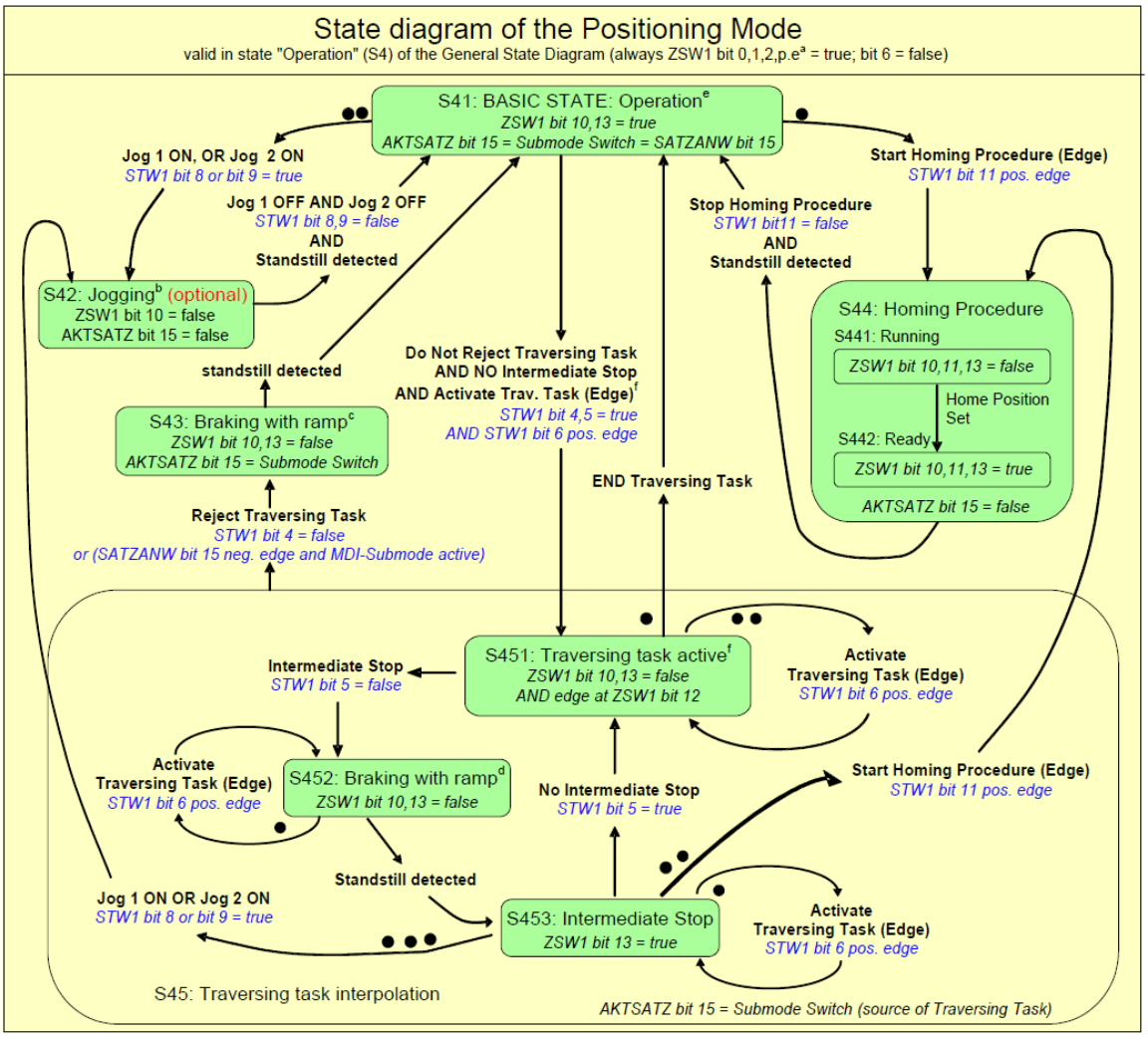
In this application class, the Drive Object (DO) provides a closed position control loop with its own position interpolation.

In PROFIdrive, two different submodes are possible which allows the controlling device to access motion task parameters using I/O messaging.

The general state machine of the drive Axis Object is extended to start, configure and stop a motion task and to execute homing procedures.

NOTE

"ONLY" in state S4 (Operational), the extended state machine can be accessed.



5.12.1 Submode "Program mode"

The Program mode can be used to start/switch to a specific predefined motion task using I/O messaging. Telegram 7 (Standard Telegram 7) or Telegram 9 is used for this purpose. For addressing the motion task signal, SATZANW is used. With signal AKTSATZ the actual running motion task number is read.

Requirements:

- Drive axis state machine needs to be in S4 (Operational)
- SATZANW bit 15 has to be 0 to select “Program mode”
- Operation mode needs to be "Position mode“
- Standard telegram 7 or 9 needs to be configured
- Axis needs to be homed (see Homing).
- Motion task needs to be configured

Start a motion task:

- Set SATZANW to the motion task number to be started.
- Set STW1 bit 4 and 5 to true (Do not reject traversing task and no intermediate stop).
- Set STW1 bit 6 from zero to one. The motion task is activated.
- ZSW1 bit 13 changes from one to zero when the axis starts moving.
- After the target position is reached, ZSW 1 bit 10 is set.

Abort or error in executing motion task:

- If the following error is not in tolerance range, ZSW1 bit 8 is set.

Warning or Fault handling:

- Case of warning, ZSW1 bit 7 is set (See also Statusword Bits (ZSW1))
- Case of fault, ZSW1 bit 3 is set (See also Statusword Bits (ZSW1))

5.12.2 Submode: Manual Data Input (MDI)

The Manual Data Input mode is used to run a motion task directly configured through I/O data. Telegram 9 is used for this purpose and defines the motion task specific signals like acceleration (MID_ACC), deceleration (MID_DEC), velocity (MDI_VEL) and target position (MDI_TAR_POS). The MDI mode is activated by setting bit 15 in signal “SATZANW”.

Requirements:

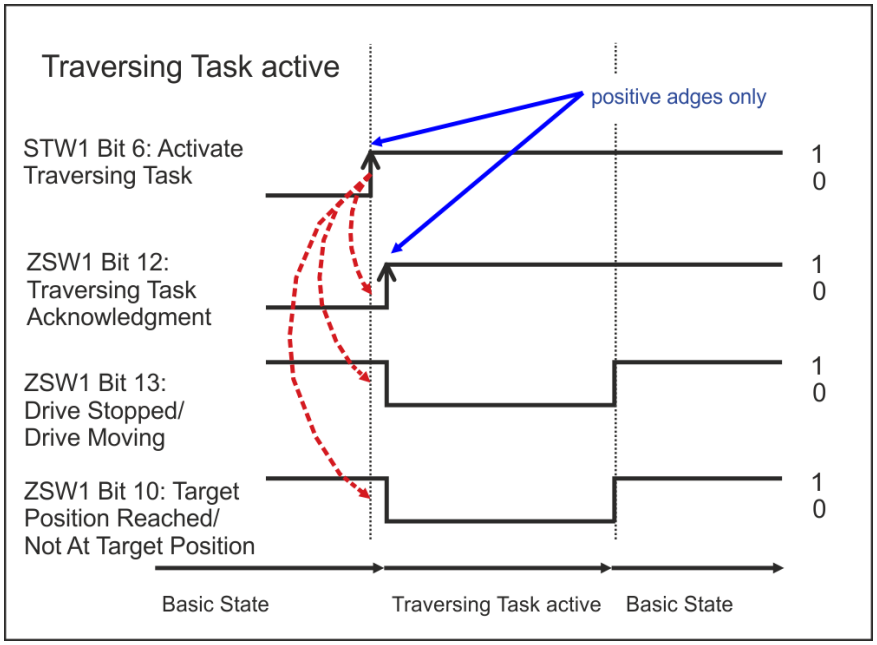
- Drive axis state machine must be in S4 (Operational).
- Operation mode must be Position mode.
- Standard telegram 9 needs to be configured.
- Axis needs to be homed (ZSW1 bit 11 set, see Statusword Bits (ZSW1).

Run a motion task:

- Set bit 15 in SATZANW to 1
- Configure all setpoint value in telegram 9 like MDI_ACC, MDI_DEC, MDI_MOD etc..
- Set STW1 its 4 and 5 to true (Do not reject traversing task and no intermediate stop).
- Set STW1 bit 6 from 0 to 1, the motion task is loaded.
- ZSW1 bit 12 is set to 1 when the drive has started the new motion task.
- ZSW1 bit 13 is set to 1 when the drive is moving after the target position is reached, ZSW 1 bit 10 is set.

The activation of a new motion task (or traversing task) is done through a hand shake algorithm. After setting the Activate Traversing Task bit 6 in STW1, the signals MDI_TARPOS, MDI_VELOCITY, MDI_ACC, MDI_DEC and MDI_MOD with their current values are loaded into the drive. The Traversing Task Acknowledgment bit 12 in ZSW1 indicates the point in time when the motion task is starting execution.

The figure describes the behavior:



If another motion task is executed before the current motion task has finished (Change on the fly), the Activate Traversing Task Bit in STW1 is directly cleared (after the setting of bit 12 in ZSW1 was detected). In addition, the drive clears bit 12 in ZSW1. After this another motion task is loaded. If an error occurs or the configured motion task cannot be executed, bit 12 in ZSW1 are not set.

5.12.3 Homing

Requirements:

- Drive axis state machine needs to be in S4 (Operational).
- The homing mode needs to be configured using AXIS#.HOME.MODE (P1001).
- No motion task is active or active task is stopped (State S453: Intermediate Stop).
- Operation mode must be Position mode.

Home procedure:

- STW1 bit 11 set to one.
- ZSW1 bit 10, 11 ,13 is set to false if homing is running.
- ZSW1 bit 10,11,13 is set to true if homing is finished.

Abort homing:

- While the homing is running, clear STW1 bit 11.

If the controller aborts a running home procedure or a homing error occurs in the drive, the home position set flag (ZSW1 bit 11) remains cleared.

Warning or Fault handling:

- Case of warning, ZSW1 bit 7 is set.
- Case of fault, ZSW1 bit 3 is set.

Mapping to AKD2G specific commands:

An activation of the homing procedure using STW1 bit 11 corresponds to the AKD2G specific command AXIS#.HOME.MOVE. When the homing procedure is finished, the AKD2G set the bits 2 and bit 4 in AXIS#.MOTIONSTAT. Only if these two bits are set, the PROFIdrive specific homing flag ZSW1 bit 11 (home position) is set.

5.12.4 Jogging

In Position mode, STW1 bits 8 and 9 is used to start a jog move while in state S41 or S453.

These bits work similarly to how they work in Velocity mode:

- Setting only bit 8 activates jog setpoint 1 defined in AXIS#.PROFINET.JOG1V (P5202).
- Setting only bit 9 activates jog setpoint 2 defined in AXIS#.PROFINET.JOG2V (P5203).
- Setting bit 8 and 9 does not change the jog setpoint (but transitions to state S42 if not already in that state).
- Clearing both bits stops the currently active jog and transitions to state S41 once the motor has stopped.

Changing a setpoint by using the Parameter Number (PNU) while that setpoint is active immediately changes the velocity command without having to re-start the jog.

Jogging in position mode uses AXIS#.JOG.ACC and AXIS#.JOG.DEC as acceleration and deceleration.

5.13 Units

5.13.1 Acyclic Units (Except position)

All parameters, except for position values that are accessed via PNU read/write record requests, are presented in units according to the associated AXIS#.UNIT.* keywords and match associated keyword values provided in WorkBench. Position values are scaled using DS402 type units to provide the required precision in 32-bits that floats cannot provide.

5.13.2 Cyclic Velocity Units

All cyclic velocities are normalized per PROFIdrivespec N2 (NSOLL_A/NIST_A) or N4 (NSOLL_B/NIST_B/MDI_VEL) data normalization. These values are scaled as a percent of a reference value and are either 2 bytes (N2) or 4 bytes (N4). For N2 values, 100% of the reference velocity corresponds to 2^14 = 0x4000 and for N4 values, 100% corresponds to 2^30 = 0x40000000. For velocities, the reference velocity is stored at p60000 and is 3k rpm by default but is writable to allow for different resolutions.

$$N2\ value = \frac{Desired\ RPM}{3000\ rpm} \times 0x4000$$

$$N4\ value = \frac{Desired\ RPM}{3000\ rpm} \times 0x40000000$$

Range and resolution using 3k rpm default:

Type	Resolution	Range
N2	5.46 RPM per count ((1/3k)*2^14)	+/- 6k RPM
N4	0.01 RPM per count ((1/3k)*2^30)	+/- 6k RPM

Example

p60000 = 3k rpm (default)

Desired RPM is 1k rpm

$$Profinet\ N2\ value = \frac{1000\ rpm}{3000\ rpm} \times 0x4000 = 5,461$$

$$Profinet\ N4\ value = \frac{1000\ rpm}{3000\ rpm} \times 0x40000000 = 357,913,941$$

5.13.3 Cyclic Acceleration Units

Cyclic acceleration signals MDI_ACC and MDI_DEC use unsigned ProfiDrive N2 normalization similar to velocity values. The reference acceleration is P5208 and is defaulted to 65,536 RPM/s.

$$N2\ value = \frac{Desired\ RPM/s}{65536\ rpm} \times 0x4000$$

Range and resolution using 65,536 RPM/s default reference accel:

Type	Resolution	Range
N2	0.25 RPM/s per count (1/65536)*2^14	0.25 RPM/s to 262,144 rpm/s

Example

P5208 = 65,536 rpm/s (default)

If desired accel is 10k rpm/s

$$Profinet\ N2\ value = \frac{10{,}000\ rpm}{3000\ rpm} \times 0x4000 = 54,613$$

5.13.4 Cyclic Current Units

Cyclic current signals use signed N2 units similar to acceleration and velocity values. The reference current is P5208 and matches AXIS#.IPEAK in Amps by default.

NOTE

IPEAK changes to match the drive model in use (i.e., a 3 Amp AKD2G will have IPEAK of 9 Amps)

$$N2\ value = \frac{Desired\ Amps}{AXIS\#.IPEAK} \times 0x4000$$

Example

Using a 3 Amp AKD2G:

P5208 = 9.000 amps (default AXIS#.IPEAK for 3A AKD2G)

If desired current is 500 milliamps:

$$Profinet\ N2\ value = \frac{0.5\ amps}{9\ amps} \times 0x4000 = 910$$

5.13.5 Position Units

The signals G1_XIST1 and G1_XIST2 are signed 32-bit position values. The resolution for these signals is fixed at 2^20 counts per revolution.

All other position values, including MDI_TARPOS, are signed 32-bit position values that are scaled according to DS402 units. These are described in examples below.

General Formulas

Given	Convert to
Motor/Drive required value (physical units)	Command value in master (fieldbus units)
Feedback value in master (fieldbus units)	Motor/Drive actual value (physical units)

Setpoint: Calculate the position command in the master given a required motor position setpoint
$$fieldbus\ position = physical\ position * \left[\frac{feed\ ratio}{encoder\ resolution*gear\ ratio}\right]$$

Actual Value: Calculate the motor feedback position given a feedback position value in the master
$$physical\ position = fieldbus\ position * \left[\frac{encoder\ resolution*gear\ ratio}{feed\ ratio}\right]$$

5.13.5.1 Feed Ratio

The Feed Ratio sets the position resolution in the fieldbus. The feed specifies the number of counts and the driving shaft revolutions specifies the number of revs. The units are counts/rev. The Feed Ratio can be any arbitrary value. It does not need to be associated with any physical units or scaling. The master controller can be programmed in 20 bits per rev = 1048576 counts/rev or 10000 counts/rev, etc. It is purely arbitrary.

$$feed\ constant = \frac{feed}{driving\ shaft\ revolutions}$$

$$feed\ constant = \frac{AXIS\#.PROFINET.FEEDRATIO.FEED}{AXIS\#.PROFINET.FEEDRATIO.SHAFTREV} = \frac{p5204}{p5205}$$

5.13.5.2 Encoder Resolution

Position encoder resolution used in these calculations is always 2^32 counts/revolution for the internal AKD2G position values regardless of the actual feedback resolution.

$$encoder\ resolution = 2^{32}\ counts/rev$$

5.13.5.3 Gear Ratio

The Gear Ratio is a position ratio that can be used to scale the fieldbus position resolution. It can be used to account for a physical gear ratio but it is not necessarily related to physical mechanics. It can be arbitrary. It can be set to any value regardless of the system mechanics.

$$gear\ ratio = \frac{motor\ shaft\ revolutions}{driving\ shaft\ revolutions}$$

$$gear\ ratio = \frac{AXIS\#.PROFINET.GEARRATIO.MOTORREV}{AXIS\#.PROFINET.GEARRATIO.SHAFTREV} = \frac{p5204}{p5052}$$

Complete Formulas

Convert to fieldbus units
fieldbus position $$position = physical\ position * \left[\frac{\frac{AXIS\#.PROFINET.FEEDRATIO.FEED}{AXIS\#.PROFINET.FEEDRATIO.SHAFTREV}}{2^3 2 * \frac{AXIS\#.PROFINET.GEARRATIO.MOTORREV}{AXIS\#.PROFINET.GEARRATIO.SHAFTREV}}\right]$$
$$fieldbus\ position = physical\ position * \left[\frac{\frac{p5042}{p5205}}{2^3 2 * \frac{p5206}{p5207}}\right]$$

Convert to physical units
physical Position $= fieldbus\ Position * \left[\frac{2^3 2 * \frac{AXIS\#.PROFINET.GEARRATIO.MOTORREV}{AXIS\#.PROFINET.GEARRATIO.SHAFTREV}}{\frac{AXIS\#.PROFINET.FEEDRATIO.FEED}{AXIS\#.PROFINET.FEEDRATIO.SHAFTREV}} \right]$
$physical\ Position = fieldbus\ Position * \left[\frac{2^3 2 * \frac{p5206}{p5207}}{\frac{p5204}{p5205}} \right]$

Real-World Units

Fieldbus position is always in units of counts. All position values in the formulas must be in units of counts converting between fieldbus position (counts) and physical position (counts).

A physical position scale of 2^32 physical counts/rev is used in the conversion between counts and real-world units.

Physical position can be converted to/from any physical units including:

- inches
- mm
- degrees

Units	Conversion from real-world units to counts for physical position “X”
inches	$physical\ position\ (counts) = X\ inches * \left(\frac{2^{32} counts}{rev} \right) * \left(Y \frac{revs}{inch} \right)$
mm	$physical\ position\ (counts) = X\ mm * \left(\frac{\frac{2^{32} counts}{rev}}{Y \frac{mm}{rev}} \right)$
degrees	$physical\ position\ (counts) = X\ deg * \left(\frac{\frac{2^{32} counts}{rev}}{360 \frac{deg}{rev}} \right)$

Example 1

- Desired Target Position = 180 degrees
- Fieldbus Position units in counts based on 20 bits per rev = 1048576 counts/rev

Values assumed:

PNU	Drive Parameter	Value
P5204	AXIS#.PROFINET.FEEDRATIO.FEED	20 bits per rev = 1048576 counts/rev
P5205	AXIS#.PROFINET.FEEDRATIO.SHAFTREV	1
P5206	AXIS#.PROFINET.GEARRATIO.MOTORREV	1
P5207	AXIS#.PROFINET.GEARRATIO.SHAFTREV	1

Formula for physical position in degrees; feed constant = 20 bits per rev = 1048576 counts/rev
$fieldbus\ position\ (counts) = physical\ position\ (deg) * \left(\frac{20 bits per rev = 1048576 counts/rev}{360} \right)$

Derivation:

Convert position from desired real-world units to counts based on 2^32 physical counts per rev.

$$physical\ position\ \ (counts) = 180\ deg * \frac{\frac{2^{32}\ counts}{rev}}{360\frac{deg}{rev}}$$

Convert from physical units to fieldbus units.

$$fieldbus\ position = physical\ position * \left[\frac{\frac{AXIS\#.CANOPEN.FCPRIMARY.FEED}{AXIS\#.CANOPEN.FCPRIMARY.SHAFTREV}}{2^{32} * \frac{AXIS\#.CANOPEN.GEARPRIMARY.MOTORREV}{AXIS\#.CANOPEN.GEARPRIMARY.SHAFTREV}}\right]$$

$$fieldbus\ position = physical\ position\ (counts) * \left[\frac{\frac{20\ bits\ per\ rev{=}1048576\ counts/rev\ counts}{1\ rev} * \frac{1}{1}}{2^{32}\ counts/rev * \frac{1\ rev}{1\ rev}}\right]$$

$$fieldbus\ position = physical\ position\ (counts) * \left[\frac{20\ bits\ per\ rev{=}1048576\ counts/rev\ counts/rev}{2^{32}\ counts/rev}\right]$$

Substituting physical position (counts):

$$fieldbus\ position = 180\ deg * \frac{\frac{2^{32}\ counts}{rev}}{360\frac{deg}{rev}} * \left[\frac{20\ bits\ per\ rev{=}1048576\ counts/rev}{2^{32}\ counts/rev}\right]$$

$$fieldbus\ position = 180\ deg * \left[\frac{20\ bits\ per\ rev{=}1048576\ counts/rev\ \frac{counts}{rev}}{360\frac{deg}{rev}}\right]$$

$$fieldbus\ position = 32768\ counts$$

This means a position command of 32768 sent over the fieldbus will command a target position of 180 degrees.

Example 2

- Fieldbus Position units in counts equivalent to hundredths of a degree (36,000 increments per rev)
- Desired Target Position = 180 degrees

Values assumed:

Object	Drive Parameter	Value
P5204	AXIS#.PROFINET.FEEDRATIO.FEED	36000
P5205	AXIS#.PROFINET.FEEDRATIO.SHAFTREV	1
P5206	AXIS#.PROFINET.GEARRATIO.MOTORREV	1
P5207	AXIS#.PROFINET.GEARRATIO.SHAFTREV	1

Formula for physical position in degrees; feed constant = 36000 counts/rev
$fieldbus\ position\ \ (counts) = physical\ Position\ \ (deg) * 100$

Derivation:

Convert position from desired real-world units to counts based on 2^32 physical counts per rev.

$$physical\ position\ \ (counts) = 180\ deg * \frac{\frac{2^{32}\ counts}{rev}}{360\frac{deg}{rev}}$$

Convert from physical units to fieldbus units.

$$fieldbus\ position = physical\ position * \left[\frac{\frac{AXIS\#.CANOPEN.FCPRIMARY.FEED}{AXIS\#.CANOPEN.FCPRIMARY.SHAFTREV}}{2^{32} * \frac{AXIS\#.CANOPEN.GEARPRIMARY.MOTORREV}{AXIS\#.CANOPEN.GEARPRIMARY.SHAFTREV}}\right]$$

$$fieldbus\ Position = physical\ position\ (counts) * \left[\frac{\frac{36000\ counts}{1\ rev}}{2^{32} counts/rev * \frac{1\ rev}{1\ rev}}\right]$$

$$fieldbus\ position = physical\ position\ (counts) * \left[\frac{36000\ counts/rev}{2^{32} counts/rev}\right]$$

$$fieldbus\ position = 180\ deg * \frac{2^{32} counts}{360 \frac{deg}{rev}} * \left[\frac{36000\ counts/rev}{2^{32} counts/rev}\right]$$

$$fieldbus\ position = 180\ deg * 100\ (counts/\deg)$$

$$fieldbus\ position = 18000\ (counts)$$

Example 3

- Ballscrew lead = 2 mm/rev
- Gearbox ratio = 5:1
- Desired Target Position = 25000 microns
- Fieldbus Position units in counts equivalent to microns

Values assumed:

Object	Drive Parameter	Value
P5204	AXIS#.PROFINET.FEEDRATIO.FEED	2000
P5205	AXIS#.PROFINET.FEEDRATIO.SHAFTREV	1
P5206	AXIS#.PROFINET.GEARRATIO.MOTORREV	5
P5207	AXIS#.PROFINET.GEARRATIO.SHAFTREV	1

Formula for physical position in microns; feed constant = 2000 counts/rev; 5:1 gear ratio
$fieldbus\ position\ (counts) = physical\ Position\ (microns)$

Derivation:

Convert position from desired real-world units to counts based on 2^32 physical counts per rev.

$$physical\ position\ (counts) = 25\ mm * \frac{2^{32} counts}{2 \frac{mm}{rev}} * 5$$

Convert from physical units to fieldbus units.

$$fieldbus\ position = physical\ position * \left[\frac{\frac{AXIS\#.CANOPEN.FCPRIMARY.FEED}{AXIS\#.CANOPEN.FCPRIMARY.SHAFTREV}}{2^{32} * \frac{AXIS\#.CANOPEN.GEARPRIMARY.MOTORREV}{AXIS\#.CANOPEN.GEARPRIMARY.SHAFTREV}}\right]$$

$$fieldbus\ Position = physical\ position\ (counts) * \left[\frac{\frac{2000\ counts}{1\ rev}}{2^{32} counts/rev * \frac{5\ rev}{1\ rev}}\right]$$

$$fieldbus\ position = physical\ position\ (counts) * \left[\frac{2000\ counts/rev}{\frac{2^{32} counts}{rev} * 5}\right]$$

$$fieldbus\ position = 25\ mm * \frac{\frac{2^{32} counts}{rev}}{2\frac{mm}{rev}} * 5 * \left[\frac{2000\ counts/rev}{\frac{2^{32} counts}{rev} * 5}\right]$$

$$fieldbus\ position = 25\ mm * 1000\ (counts/mm)$$

$$fieldbus\ position = 25000\ (counts)$$

5.14 Fault Buffer

PROFIDrive specifies a fault buffer mechanism that allows accessing current faults as well as a history of acknowledged faults.

The buffer can be accessed with the following PNUs:

PNU	Message
P944	Fault message counter
P947	List of fault numbers
P948	List of fault timestamps
P951	List of fault descriptions
P952	Fault situation counter

5.14.0.1 Fault message counter (P944)

The fault message counter in P944 is incremented every time the content of P947/P948 changes, i.e., a new fault occurs or faults are acknowledged.

5.14.0.2 List of fault numbers and P948: List of fault timestamps (P947)

The lists of fault numbers/timestamps are organized in 8 so called “fault situations” with 8 faults each.

The first fault situation (subindex 0 to 7) contains the currently active faults with the latest fault in the highest subindex. If more than 8 faults are triggered before faults are acknowledged, the last fault (subindex 7) is over-written.

The fault numbers in the list are the same that are displayed in WorkBench (Faults and Warning Messages). The timestamp is the number of milliseconds that passed since booting the drive when the fault occurred. A fault number/timestamp of 0 means that this fault entry is empty.

When acknowledging faults (positive edge on STW1 bit 7) with a least one active fault, a new, empty fault situation is created at the start of the lists of fault numbers/timestamps pushing back the old situations to higher subindices. The second to eighth situations each contain the list of faults active when faults were acknowledged.

Faults that could not be cleared by the acknowledge are immediately reinstated in the new fault situation.

The following table provides an example of a fault buffer:

Subindex	P947	P948	
0	4012 4015 4016	123356 123456 123567	Active faults
1			
2			
3			
4			
5			
6			
7			
8	6000	98765	Most recently acknowledged faults
9			
10			
11			
12			
13			
14			
15			
⋮	⋮	⋮	
56	5501 5505	54321 54421	Least recently acknowledged faults
57			
58			
59			
60			
61			
62			
63			

5.14.0.3 List of fault descriptions (P951)

P951 implements a static list that provides a descriptive text for every fault number.

To obtain the fault description, the additional text of P951 can be read with subindex = fault number. The text read matches the description returned by AXIS#.FAULTS but is limited to the first 16 characters of that description. If a description is unclear by shortening, refer to Faults and Warning Messages.

5.14.0.4 Fault situation counter (P952)

The fault situation counter in P952 is incremented when the first fault is entered into a new fault situation.

It can also be set to 0 to reset the whole fault buffer. This sets all counters, fault numbers and fault timestamp entries to 0.

5.15 Drive Reboot

5.15.1 Drive Reset/Reboot (P972)

The drive is rebooted by writing a 1 to P972. The drive immediately reboots and closes the connection. No response is sent to the write request and the controller is required to re-establish communication after the drive completes startup. On startup the value is 0, indicating successful completion of the command.

5.16 Parameter Reset/Load/Save

5.16.1 Reset Parameters (P976) to Default and NV Load

P976 can be used to reset parameters or load parameters from NV. Once a value is written, the value remains until it completes. The controller can continue to read back the value to detect completion of the reset/load.

Value	Description
0	Idle or successfully completed last operation
1	Perform DRV.RSTVAR
2	Perform DRV.NVLOAD
255	Failed last operation

5.16.2 NV Save (P977)

PNU 977 can be used to save parameters to NV. Once a value is written, the value remains until it completes. The controller can continue to read back the value to detect completion of the save.

Value	Description
0	Idle or successfully completed last operation
1	Perform DRV.NVSAVE
255	Failed last operation

6 PROFINET Parameters

The following parameters are provided to configure and inspect the PROFINET interface.

Parameters	Description
AXIS#.PROFINET.ACTUALVALUEDATA	Reads cyclic data sent from the drive.
AXIS#.PROFINET.ACTUALVALUEMAP	Specifies signals currently mapped to the actual value telegram of the axis.
AXIS#.PROFINET.FEEDRATIO.FEED	Count of feed input for the position feed ratio calculation.
AXIS#.PROFINET.FEEDRATIO.SHAFTREV	Count of driving shaft revolutions for the position feed ratio calculation.
AXIS#.PROFINET.GEARRATIO.MOTORREV	Count of motor shaft revolutions for the position gear ratio calculation.
AXIS#.PROFINET.GEARRATIO.SHAFTREV	Count of driving shaft revolutions for the position gear ratio calculation.
AXIS#.PROFINET.JOG1V	Velocity target used when JOG1 is triggered.
AXIS#.PROFINET.JOG2V	Velocity target used when JOG2 is triggered.
AXIS#.PROFINET.KPCSCALED	KPC controller position gain signal from telegram 5, scaled into Hz
AXIS#.PROFINET.MAXSOLFAIL	Maximum times the sign of life value can be incorrect before faulting.
AXIS#.PROFINET.REFACC	Acceleration used to calculate cyclic velocity values.
AXIS#.PROFINET.REFV	Velocity used to calculate cyclic velocity values.
AXIS#.PROFINET.SETPOINTDATA	Reads cyclic data sent from the controller to the drive.
AXIS#.PROFINET.SETPOINTMAP	Specifies signals currently mapped to the setpoint telegram of the axis.
AXIS#.PROFINET.SIGNALLIST	Lists signals available to map in free telegrams.
AXIS#.PROFINET.SOLFAILCNT	Indicates current error counter of the sign of life monitoring of STW2.
AXIS#.PROFINET.STATE	Indicates current state of the associated axis.
AXIS#.PROFINET.STW1	Allows recording, triggering, and viewing.
AXIS#.PROFINET.STW2	Allows recording, triggering, and viewing.
AXIS#.PROFINET.SUPPACTUALVALUEDATA	Reads cyclic data sent from the drive.
AXIS#.PROFINET.SUPPSETPOINTDATA	Reads cyclic data sent from the controller.
AXIS#.PROFINET.SUPPTELEGRAMID	Selects or identifies current supplementary data telegram.
AXIS#.PROFINET.TELEGRAMID	Allows reading or changing the active telegram.
AXIS#.PROFINET.VELCOMP	Velocity used to set bit 10 of thestatus word ZSW1.

Parameters	Description
AXIS#.PROFINET.VELTOLERANCE	Velocity used to set bit 8 of the status word ZSW1.
AXIS#.PROFINET.ZSW1	Allows recording, triggering, and viewing.
AXIS#.PROFINET.ZSW2	Allows recording, triggering, and viewing.
PROFINET.INFO	Provides information about the operation of the drive.
PROFINET.PLLMODE	Indicates if the PLL is running.
PROFINET.PLLSTATE	Reads state of the PLL.
PROFINET.USEMODULECFG	Controls where the drive obtains the telegram configuration.

7 Appendix A: Manufacturers' PNUs

The following table lists and describes all manufacturers' specific PNUs.

Parameter	PNU	Data Type	Access	Units
AXIS#.PROFINET.STW1	1	Unsigned16	Read Only	
AXIS#.PROFINET.ZSW1	2	Unsigned16	Read Only	
AXIS#.PROFINET.STW2	3	Unsigned16	Read Only	
AXIS#.PROFINET.ZSW2	4	Unsigned16	Read Only	
AXIS#.PROFINET.NSOLLA	5	Signed16	Read Only	
AXIS#.PROFINET.NISTA	6	Signed16	Read Only	
AXIS#.PROFINET.NSOLLB	7	Signed32	Read Only	
AXIS#.PROFINET.NISTB	8	Signed32	Read Only	
AXIS#.PROFINET.G1STW	9	Unsigned16	Read Only	
AXIS#.PROFINET.G1ZSW	10	Unsigned16	Read Only	
AXIS#.PROFINET.G1XIST1	11	Signed32	Read Only	
AXIS#.PROFINET.G1XIST2	12	Signed32	Read Only	
AXIS#.PROFINET.XISTA	28	Signed32	Read Only	
AXIS#.PROFINET.SATZANW	32	Unsigned16	Read Only	
AXIS#.PROFINET.AKTSATZ	33	Unsigned16	Read Only	
AXIS#.PROFINET.MDITARPOS	34	Signed32	Read Only	
AXIS#.PROFINET.MDIVELOCITY	35	Unsigned32	Read Only	
AXIS#.PROFINET.MDIACC	36	Unsigned16	Read Only	
AXIS#.PROFINET.MDIDEC	37	Unsigned16	Read Only	
AXIS#.PROFINET.MDIMOD	38	Unsigned16	Read Only	
AXIS#.PROFINET.MAXSOLFAIL	925	Unsigned16	Read/Write	
DRV.DIS	2000	Unsigned8	Read/Write	
DRV.BLINKDISPLAY	2001	Unsigned8	Read/Write	
DRV.CLRFAULTS	2002	Unsigned8	Read/Write	
DRV.TYPE	2003	Unsigned8	Read Only	
DRV.RSTVAR	2004	Unsigned8	Read/Write	
DRV.NVSAVE	2005	Unsigned8	Read/Write	
DRV.STOP	2006	Unsigned8	Read/Write	
DRV.NVLOAD	2007	Unsigned8	Read/Write	
DRV.RUNTIME	2008	Unsigned32	Read Only	

Parameter	PNU	Data Type	Access	Units
DRV.NVCHECK	2009	Unsigned32	Read Only	
DRV.CUSTOMIDENTIFIER	2010	String	Read/Write	
DRV.NAME	2011	String	Read/Write	
DRV.NVVER	2012	String	Read/Write	
DRV.REBOOT	2013	Unsigned32	Read/Write	
DRV.DOWNLOADALLOWED	2014	Unsigned8	Read Only	
DRV.ADDUSER	2015	String	Read/Write	
DRV.DELUSER	2016	String	Read/Write	
DRV.LOGOUT	2017	Unsigned8	Read/Write	
DRV.SETUSERPWD	2018	String	Read/Write	
IP.DEFAULTINTERFACE	2200	Unsigned8	Read/Write	
IP.MODE	2201	Unsigned16	Read/Write	
IP.ADDRESS	2202	Unsigned32	Read/Write	
IP.GATEWAY	2203	Unsigned32	Read/Write	
IP.PROTOCOL	2204	Unsigned8	Read/Write	
IP.RESET	2205	Unsigned8	Read/Write	
IP.SUBNET	2206	Unsigned32	Read/Write	
REC.ACTIVE	2300	Unsigned8	Read Only	
REC.DONE	2301	Unsigned8	Read Only	
REC.GAP	2302	Unsigned16	Read/Write	
REC.NUMPOINTS	2303	Unsigned16	Read/Write	
REC.OFF	2304	Unsigned8	Read/Write	
REC.STOPTYPE	2305	Unsigned8	Read/Write	
REC.TRIG	2306	Unsigned8	Read/Write	
REC.TRIGPOS	2307	Unsigned8	Read/Write	
REC.RETRIEVESIZE	2308	Unsigned16	Read/Write	
REC.TRIGSLOPE	2309	Unsigned8	Read/Write	
REC.TRIGTYPE	2310	Unsigned8	Read/Write	
REC.TRIGVAL	2311	Float	Read/Write	
REC.CH1	2312	String	Read/Write	
REC.CH2	2313	String	Read/Write	
REC.CH3	2314	String	Read/Write	
REC.CH4	2315	String	Read/Write	
REC.CH5	2316	String	Read/Write	

Parameter	PNU	Data Type	Access	Units
REC.CH6	2317	String	Read/Write	
REC.RETRIEVEFRMT	2318	Unsigned8	Read/Write	
REC.TRIGMASK	2319	Unsigned32	Read/Write	
REC.TRIGPARAM	2320	String	Read/Write	
REGEN.POWER	2400	Unsigned32	Read Only	
REGEN.REXT	2401	Unsigned16	Read/Write	
REGEN.TEXT	2402	Float	Read/Write	
REGEN.TYPE	2403	Signed8	Read/Write	
REGEN.WATTEXT	2404	Unsigned16	Read/Write	
REGEN.POWERFILTERED	2405	Unsigned32	Read Only	
VBUS.VALUE	2500	Float	Read Only	
VBUS.CAP	2501	Unsigned32	Read Only	
VBUS.DCOPERATION	2502	Unsigned8	Read/Write	
VBUS.ICAP	2503	Float	Read Only	
VBUS.ICAPLIMIT	2504	Float	Read Only	
VBUS.INRUSHOFF	2505	Unsigned16	Read Only	
VBUS.INRUSHON	2506	Unsigned16	Read Only	
VBUS.OVFTHRESH	2507	Unsigned16	Read Only	
VBUS.OVWTHRESH	2508	Unsigned16	Read/Write	
VBUS.THREEPHASE	2509	Unsigned8	Read/Write	
VBUS.UVFTHRESH	2510	Unsigned16	Read/Write	
VBUS.UVWTHRESH	2511	Unsigned16	Read/Write	
VBUS.UVMODE	2512	Unsigned8	Read/Write	
VBUS.ACNOMINAL	2513	Unsigned16	Read/Write	
VBUS.DCNOMINAL	2514	Unsigned16	Read/Write	
FBUS.TYPE	2600	Unsigned8	Read Only	
DIN.STATES	2700	Unsigned32	Read Only	
DIO.STATES	2701	Unsigned32	Read Only	
DOUT.STATES	2702	Unsigned32	Read Only	
DOUT.STATESU	2703	Unsigned32	Read/Write	
MW.USERBUFFER	2800	Float[32]	Read/Write	
MW.MODEL1.STATE	2801	Unsigned8	Read/Write	
MW.MODEL2.STATE	2802	Unsigned8	Read/Write	
HW.FANSPEED1	2900	Float	Read Only	

Parameter	PNU	Data Type	Access	Units
GANTRY.PL.ERR	3000	Signed32	Read Only	Position
GANTRY.PL.ERRFTHRESH	3001	Signed32	Read/Write	Position
GANTRY.PL.ERRWTHRESH	3002	Signed32	Read/Write	Position
GANTRY.STATE	3003	Unsigned8	Read Only	
BRAKE1.AXIS	3100	Unsigned8	Read/Write	
BRAKE2.AXIS	3101	Unsigned8	Read/Write	
BRAKE1.STATE	3105	Unsigned8	Read Only	
BRAKE2.STATE	3106	Unsigned8	Read Only	
USER.INT1	3200	Signed16	Read/Write	
USER.INT2	3201	Signed16	Read/Write	
USER.INT3	3202	Signed16	Read/Write	
USER.INT4	3203	Signed16	Read/Write	
USER.INT5	3204	Signed16	Read/Write	
USER.INT6	3205	Signed16	Read/Write	
USER.INT7	3206	Signed16	Read/Write	
USER.INT8	3207	Signed16	Read/Write	
USER.INT9	3208	Signed16	Read/Write	
USER.INT10	3209	Signed16	Read/Write	
PROFINET.PLLMODE	3300	Unsigned8	Read Only	
PROFINET.PLLSTATE	3301	Unsigned8	Read Only	
PROFINET.SAMPLEPERIOD	3302	Unsigned32	Read/Write	
SD.LOGEN	3400	Unsigned8	Read/Write	
SD.STATUS	3401	Unsigned8	Read Only	
LOG.SOURCE	3500	Unsigned32	Read/Write	
X22.MODE	3600	Unsigned8	Read/Write	
X23.MODE	3601	Unsigned8	Read/Write	
SAFE.ID	3800	Unsigned8	Read Only	
SAFE.CRC	3801	Unsigned32	Read Only	
SAFE.STATE	3802	Unsigned8	Read Only	
SAFE.ERROR	3803	Unsigned16	Read Only	
SAFE.SMMHWREVISION	3804	String	Read Only	
SAFE.SMMSWREVISION	3805	Unsigned16	Read Only	
AXIS#.NAME	5000	String	Read/Write	
AXIS#.ZEROT	5001	Unsigned32	Read/Write	

Parameter	PNU	Data Type	Access	Units
AXIS#.ZEROV	5002	Float	Read/Write	Velocity
AXIS#.ACTIVE	5003	Unsigned8	Read Only	
AXIS#.CMDSOURCE	5004	Unsigned8	Read/Write	
AXIS#.DBILIMIT	5005	Float	Read/Write	
AXIS#.DIR	5006	Unsigned8	Read/Write	
AXIS#.DIS	5007	Unsigned8	Read/Write	
AXIS#.DISMODE	5008	Unsigned8	Read/Write	
AXIS#.DISSOURCES	5009	Unsigned16	Read Only	
AXIS#.DISTO	5010	Unsigned32	Read/Write	
AXIS#.EN	5011	Unsigned8	Read/Write	
AXIS#.ENDEFAULT	5012	Unsigned8	Read/Write	
AXIS#.ICONT	5013	Float	Read Only	
AXIS#.IPEAK	5014	Float	Read Only	
AXIS#.OPMODE	5015	Unsigned8	Read/Write	
AXIS#.STOP	5016	Unsigned8	Read/Write	
AXIS#.SETUPREQBITS	5017	Unsigned32	Read Only	
AXIS#.WARNING1	5018	Unsigned32	Read Only	
AXIS#.WARNING2	5019	Unsigned32	Read Only	
AXIS#.WARNING3	5020	Unsigned32	Read Only	
AXIS#.MOTIONDISSOURCES	5021	Unsigned16	Read Only	
AXIS#.CLRFAULTS	5022	Unsigned8	Read/Write	
AXIS#.DISSOURCESMASK	5023	Unsigned16	Read Only	
AXIS#.FAULTED	5024	Unsigned8	Read Only	
AXIS#.MOTIONSTAT	5025	Unsigned32	Read Only	
AXIS#.TEMP	5026	Float	Read Only	
AXIS#.TEMPFTHRESH	5027	Float	Read Only	
AXIS#.TEMPWTHRESH	5028	Float	Read Only	
AXIS#.UTFTHRESH	5029	Float	Read Only	
AXIS#.UTWTHRESH	5030	Float	Read Only	
AXIS#.ZEROACC	5031	Float	Read/Write	Acceleration
AXIS#.ZEROREACHED	5032	Unsigned8	Read Only	
AXIS#.FBUS.ACC	5100	Float	Read/Write	Acceleration
AXIS#.FBUS.DEC	5101	Float	Read/Write	Acceleration
AXIS#.FBUS.PROTECTION	5102	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
AXIS#.FBUS.BLOCKING	5103	Unsigned8	Read Only	
AXIS#.PROFINET.VELCOMP	5200	Float	Read/Write	Velocity
AXIS#.PROFINET.VELTOLERANCE	5201	Float	Read/Write	Velocity
AXIS#.PROFINET.JOG1V	5202	Float	Read/Write	Velocity
AXIS#.PROFINET.JOG2V	5203	Float	Read/Write	Velocity
AXIS#.PROFINET.FEEDRATIO.FEED	5204	Unsigned32	Read/Write	
AXIS#.PROFINET.FEEDRATIO.SHAFTREV	5205	Unsigned32	Read/Write	
AXIS#.PROFINET.GEARRATIO.MOTORREV	5206	Unsigned32	Read/Write	
AXIS#.PROFINET.GEARRATIO.SHAFTREV	5207	Unsigned32	Read/Write	
AXIS#.PROFINET.REFACC	5208	Float	Read/Write	Acceleration
AXIS#.PROFINET.SOLFAILCNT	5209	Unsigned16	Read Only	
AXIS#.CS.DEC	5300	Float	Read/Write	Acceleration
AXIS#.CS.STATE	5301	Unsigned8	Read Only	
AXIS#.GUI.PARAM01	5400	Signed32	Read/Write	
AXIS#.GUI.PARAM02	5401	Signed32	Read/Write	
AXIS#.GUI.PARAM03	5402	Signed32	Read/Write	
AXIS#.GUI.PARAM04	5403	Signed32	Read/Write	
AXIS#.GUI.PARAM05	5404	Signed32	Read/Write	
AXIS#.GUI.PARAM06	5405	Signed32	Read/Write	
AXIS#.GUI.PARAM07	5406	Signed32	Read/Write	
AXIS#.GUI.PARAM08	5407	Signed32	Read/Write	
AXIS#.GUI.PARAM09	5408	Signed32	Read/Write	
AXIS#.GUI.PARAM10	5409	Signed32	Read/Write	
AXIS#.GEAR.ACC	5500	Float	Read/Write	Acceleration
AXIS#.GEAR.DEC	5501	Float	Read/Write	Acceleration
AXIS#.GEAR.IN	5502	Unsigned16	Read/Write	
AXIS#.GEAR.MOVE	5503	Unsigned8	Read/Write	
AXIS#.GEAR.OUT	5504	Signed16	Read/Write	
AXIS#.GEAR.FBSOURCE	5505	Unsigned8	Read/Write	
AXIS#.GEAR.STATE	5506	Signed16	Read Only	
AXIS#.HWLS.NEGSTATE	5600	Unsigned8	Read Only	
AXIS#.HWLS.POSSTATE	5601	Unsigned8	Read Only	
AXIS#.HWLS.NEGSOURCE	5602	Unsigned8	Read/Write	
AXIS#.HWLS.POSSOURCE	5603	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
AXIS#.LOAD.INERTIA	5700	Float	Read/Write	
AXIS#.IL.BUSFF	5800	Float	Read Only	
AXIS#.IL.CMD	5801	Float	Read Only	
AXIS#.IL.CMDU	5802	Float	Read/Write	
AXIS#.IL.FB	5803	Float	Read Only	
AXIS#.IL.FF	5804	Float	Read Only	
AXIS#.IL.FOLDFTHRESH	5805	Float	Read Only	
AXIS#.IL.FOLDFTHRESHU	5806	Float	Read/Write	
AXIS#.IL.FOLDWTHRESH	5807	Float	Read/Write	
AXIS#.IL.FRICTION	5808	Float	Read/Write	
AXIS#.IL.IFOLD	5809	Float	Read Only	
AXIS#.IL.KACCFF	5810	Float	Read/Write	
AXIS#.IL.AINSOURCE	5811	Unsigned8	Read/Write	
AXIS#.IL.KVFF	5812	Float	Read/Write	
AXIS#.IL.LIMITN	5813	Float	Read/Write	
AXIS#.IL.LIMITP	5814	Float	Read/Write	
AXIS#.IL.MIFOLD	5815	Float	Read Only	
AXIS#.IL.OFFSET	5816	Float	Read/Write	
AXIS#.IL.VCMD	5817	Float	Read Only	
AXIS#.IL.BW	5818	Float	Read/Write	
AXIS#.IL.KP	5819	Float	Read Only	
AXIS#.IL.DIFOLD	5820	Float	Read Only	
AXIS#.IL.FBSOURCE	5821	Unsigned8	Read/Write	
AXIS#.IL.CMDACC	5822	Float	Read Only	Acceleration
AXIS#.IL.PWMFREQ	5823	Float	Read Only	
AXIS#.IL.KPLOOKUP	5824	Float[256]	Read Only	
AXIS#.IL.DI2T	5825	Float	Read Only	
AXIS#.IL.AINSCALE	5826	Float	Read/Write	
AXIS#.IL.MI2T	5827	Float	Read Only	
AXIS#.IL.PWMQUIET	5828	Unsigned8	Read/Write	
AXIS#.PL.CMD	5900	Signed32	Read Only	Position
AXIS#.PL.ERR	5901	Signed32	Read Only	Position
AXIS#.PL.ERRFTHRESH	5902	Signed32	Read/Write	Position
AXIS#.PL.ERRWTHRESH	5903	Signed32	Read/Write	Position

Parameter	PNU	Data Type	Access	Units
AXIS#.PL.FB	5904	Signed32	Read Only	Position
AXIS#.PL.FBSOURCE	5905	Unsigned8	Read/Write	
AXIS#.PL.INTOUTMAX	5906	Float	Read/Write	Velocity
AXIS#.PL.KI	5907	Float	Read/Write	
AXIS#.PL.KP	5908	Float	Read/Write	
AXIS#.PL.MODP1	5909	Signed32	Read/Write	Position
AXIS#.PL.MODP2	5910	Signed32	Read/Write	Position
AXIS#.PL.MODPDIR	5911	Unsigned8	Read/Write	
AXIS#.PL.MODPEN	5912	Unsigned8	Read/Write	
AXIS#.PL.AINSCALE	5913	Signed32	Read/Write	Position
AXIS#.PL.FILTERTIME	5914	Float	Read/Write	
AXIS#.PL.AINSOURCE	5915	Unsigned8	Read/Write	
AXIS#.PL.KITHRESH	5916	Float	Read/Write	Velocity
AXIS#.PL.OFFSET	5917	Signed32	Read/Write	Position
AXIS#.VL.ARPF1	6000	Float	Read/Write	
AXIS#.VL.ARPF2	6001	Float	Read/Write	
AXIS#.VL.ARPF3	6002	Float	Read/Write	
AXIS#.VL.ARPF4	6003	Float	Read/Write	
AXIS#.VL.ARPQ1	6004	Float	Read/Write	
AXIS#.VL.ARPQ2	6005	Float	Read/Write	
AXIS#.VL.ARPQ3	6006	Float	Read/Write	
AXIS#.VL.ARPQ4	6007	Float	Read/Write	
AXIS#.VL.ARTYPE1	6008	Signed8	Read/Write	
AXIS#.VL.ARTYPE2	6009	Signed8	Read/Write	
AXIS#.VL.ARTYPE3	6010	Signed8	Read/Write	
AXIS#.VL.ARTYPE4	6011	Signed8	Read/Write	
AXIS#.VL.ARZF1	6012	Float	Read/Write	
AXIS#.VL.ARZF2	6013	Float	Read/Write	
AXIS#.VL.ARZF3	6014	Float	Read/Write	
AXIS#.VL.ARZF4	6015	Float	Read/Write	
AXIS#.VL.ARZQ1	6016	Float	Read/Write	
AXIS#.VL.ARZQ2	6017	Float	Read/Write	
AXIS#.VL.ARZQ3	6018	Float	Read/Write	
AXIS#.VL.ARZQ4	6019	Float	Read/Write	

Parameter	PNU	Data Type	Access	Units
AXIS#.VL.BUSFF	6020	Float	Read/Write	Velocity
AXIS#.VL.CMD	6021	Float	Read Only	Velocity
AXIS#.VL.CMDU	6022	Float	Read/Write	Velocity
AXIS#.VL.ERR	6023	Float	Read Only	Velocity
AXIS#.VL.FB	6024	Float	Read Only	Velocity
AXIS#.VL.FBFILTER	6025	Float	Read Only	Velocity
AXIS#.VL.FBSOURCE	6026	Unsigned8	Read/Write	
AXIS#.VL.FF	6027	Float	Read Only	Velocity
AXIS#.VL.KI	6028	Float	Read/Write	
AXIS#.VL.KP	6029	Float	Read/Write	
AXIS#.VL.KVFF	6030	Float	Read/Write	
AXIS#.VL.LIMITN	6031	Float	Read/Write	Velocity
AXIS#.VL.LIMITP	6032	Float	Read/Write	Velocity
AXIS#.VL.LMJR	6033	Float	Read/Write	
AXIS#.VL.THRESH	6034	Float	Read/Write	Velocity
AXIS#.VL.FBUNFILTERED	6035	Float	Read Only	Velocity
AXIS#.VL.VFTHRESH	6036	Float	Read Only	Velocity
AXIS#.VL.AINSCALE	6037	Float	Read/Write	Velocity
AXIS#.VL.AINACC	6038	Float	Read/Write	Acceleration
AXIS#.VL.AINDEC	6039	Float	Read/Write	Acceleration
AXIS#.VL.AINSOURCE	6040	Unsigned8	Read/Write	
AXIS#.VL.ACCFILTERED	6041	Float	Read Only	Acceleration
AXIS#.VL.KIMODE	6042	Unsigned8	Read/Write	
AXIS#.HOME.ACC	6100	Float	Read/Write	Acceleration
AXIS#.HOME.AUTOMOVE	6101	Unsigned8	Read/Write	
AXIS#.HOME.DEC	6102	Float	Read/Write	Acceleration
AXIS#.HOME.DIR	6103	Unsigned16	Read/Write	
AXIS#.HOME.DIST	6104	Signed32	Read/Write	Position
AXIS#.HOME.CREEPFACTOR	6105	Unsigned32	Read/Write	
AXIS#.HOME.IPEAK	6106	Float	Read/Write	
AXIS#.HOME.MODE	6107	Unsigned16	Read/Write	
AXIS#.HOME.MOVE	6108	Unsigned8	Read/Write	
AXIS#.HOME.P	6109	Signed32	Read/Write	Position
AXIS#.HOME.PERRTHRESH	6110	Signed32	Read/Write	Position

Parameter	PNU	Data Type	Access	Units
AXIS#.HOME.SET	6111	Unsigned8	Read/Write	
AXIS#.HOME.V	6112	Float	Read/Write	Velocity
AXIS#.HOME.MAXDIST	6113	Signed32	Read/Write	Position
AXIS#.HOME.CLEAR	6114	Unsigned8	Read/Write	
AXIS#.HOME.MULTITURNMODE	6115	Unsigned8	Read/Write	
AXIS#.HOME.OFFSET	6116	Signed32	Read Only	Position
AXIS#.HOME.OFFSETUSER	6117	Signed32	Read/Write	Position
AXIS#.HOME.SWITCHSOURCE	6118	Unsigned8	Read/Write	
AXIS#.HOME.SWITCHSTATE	6119	Unsigned8	Read Only	
AXIS#.MOTOR.AUTOSET	6200	Unsigned8	Read/Write	
AXIS#.MOTOR.BRAKE	6201	Unsigned8	Read Only	
AXIS#.MOTOR.BRAKECONTROL	6202	Unsigned8	Read/Write	
AXIS#.MOTOR.CTF0	6203	Float	Read Only	
AXIS#.MOTOR.ICONT	6204	Float	Read Only	
AXIS#.MOTOR.IDDATAVALID	6205	Unsigned8	Read Only	
AXIS#.MOTOR.INERTIA	6206	Float	Read Only	
AXIS#.MOTOR.IPEAK	6207	Float	Read Only	
AXIS#.MOTOR.KT	6208	Float	Read Only	
AXIS#.MOTOR.LQLL	6209	Float	Read Only	
AXIS#.MOTOR.PHASE	6210	Unsigned16	Read/Write	
AXIS#.MOTOR.PITCH	6211	Float	Read/Write	
AXIS#.MOTOR.POLES	6212	Unsigned16	Read Only	
AXIS#.MOTOR.R	6213	Float	Read Only	
AXIS#.MOTOR.RTYPE	6214	Unsigned8	Read/Write	
AXIS#.MOTOR.TBRAKEAPP	6215	Unsigned16	Read Only	
AXIS#.MOTOR.TBRAKERLS	6216	Unsigned16	Read Only	
AXIS#.MOTOR.TEMP	6217	Unsigned32	Read Only	
AXIS#.MOTOR.TEMPFAULT	6218	Unsigned32	Read Only	
AXIS#.MOTOR.TEMPWARN	6219	Unsigned32	Read/Write	
AXIS#.MOTOR.TYPE	6220	Unsigned8	Read Only	
AXIS#.MOTOR.VMAX	6221	Unsigned16	Read Only	
AXIS#.MOTOR.VOLTMAX	6222	Unsigned16	Read Only	
AXIS#.MOTOR.TBRAKETO	6223	Signed32	Read/Write	
AXIS#.MOTOR.BRAKEIMM	6224	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
AXIS#.MOTOR.VOLTMIN	6225	Unsigned16	Read/Write	
AXIS#.MOTOR.VOLTRATED	6226	Unsigned16	Read/Write	
AXIS#.MOTOR.VRATED	6227	Float	Read/Write	
AXIS#.MOTOR.IMTR	6228	Unsigned16	Read/Write	
AXIS#.MOTOR.IMID	6229	Float	Read/Write	
AXIS#.MOTOR.LDLL	6230	Float	Read Only	
AXIS#.MOTOR.LISAT	6231	Float	Read Only	
AXIS#.MOTOR.IDMAX	6232	Float	Read Only	
AXIS#.MOTOR.PHSADVK1	6233	Float	Read Only	
AXIS#.MOTOR.PHSADVK2	6234	Float	Read Only	
AXIS#.MOTOR.TEMPC	6235	Signed16	Read Only	
AXIS#.MOTOR.FIELDWEAKENING	6236	Unsigned8	Read/Write	
AXIS#.MOTOR.NAME	6237	String	Read Only	
AXIS#.MOTOR.BRAKEPOWERDELAY	6238	Unsigned16	Read Only	
AXIS#.MOTOR.BRAKEPOWERLOW	6239	Float	Read Only	
AXIS#.MOTOR.BRAKEPOWERSAVING	6240	Unsigned8	Read Only	
AXIS#.MOTOR.KE	6241	Float	Read Only	
AXIS#.MOTOR.SERIALNUM	6242	String	Read Only	
AXIS#.MT.ACC	6300	Float[32]	Read/Write	Acceleration
AXIS#.MT.CLEAR	6301	Unsigned8 [32]	Read/Write	
AXIS#.MT.CNTL	6302	Unsigned32 [32]	Read/Write	
AXIS#.MT.CONTINUE	6303	Unsigned8	Read/Write	
AXIS#.MT.DEC	6304	Float[32]	Read/Write	Acceleration
AXIS#.MT.MOVE	6305	Unsigned8 [32]	Read/Write	
AXIS#.MT.MTNEXT	6306	Signed8[32]	Read/Write	
AXIS#.MT.P	6307	Signed32 [32]	Read/Write	Position
AXIS#.MT.TNEXT	6308	Unsigned16 [32]	Read/Write	
AXIS#.MT.V	6309	Float[32]	Read/Write	Velocity
AXIS#.MT.VCMD	6310	Float	Read Only	Velocity
AXIS#.MT.FEEDRATE	6311	Float	Read/Write	

Parameter	PNU	Data Type	Access	Units
AXIS#.MT.CAP	6312	Unsigned8 [32]	Read/Write	
AXIS#.MT.CLEARALL	6313	Unsigned8	Read/Write	
AXIS#.MT.DISALLOWINTERRUPT	6314	Unsigned8 [32]	Read/Write	
AXIS#.MT.DISALLOWZEROSTARTVEL	6315	Unsigned8 [32]	Read/Write	
AXIS#.MT.RUNNINGTASK	6316	Signed8	Read Only	
AXIS#.MT.TRANSITION	6317	Unsigned8 [32]	Read/Write	
AXIS#.MT.TYPE	6318	Unsigned8 [32]	Read/Write	
AXIS#.SM.I1	6400	Float	Read/Write	
AXIS#.SM.I2	6401	Float	Read/Write	
AXIS#.SM.MODE	6402	Unsigned16	Read/Write	
AXIS#.SM.MOVE	6403	Unsigned8	Read/Write	
AXIS#.SM.T1	6404	Unsigned16	Read/Write	
AXIS#.SM.T2	6405	Unsigned16	Read/Write	
AXIS#.SM.V1	6406	Float	Read/Write	Velocity
AXIS#.SM.V2	6407	Float	Read/Write	Velocity
AXIS#.SM.ACC	6408	Float	Read/Write	Acceleration
AXIS#.SM.DEC	6409	Float	Read/Write	Acceleration
AXIS#.SWLS.EN	6500	Unsigned8	Read/Write	
AXIS#.SWLS.LIMIT0	6501	Signed32	Read/Write	Position
AXIS#.SWLS.LIMIT1	6502	Signed32	Read/Write	Position
AXIS#.SWLS.STATE	6503	Unsigned8	Read Only	
AXIS#.UNIT.ACCLINEAR	6600	Unsigned8	Read/Write	
AXIS#.UNIT.ACCROTARY	6601	Unsigned8	Read/Write	
AXIS#.UNIT.PIN	6602	Unsigned32	Read/Write	
AXIS#.UNIT.PLINEAR	6603	Unsigned8	Read/Write	
AXIS#.UNIT.POUT	6604	Unsigned32	Read/Write	
AXIS#.UNIT.PROTARY	6605	Unsigned8	Read/Write	
AXIS#.UNIT.VLINEAR	6606	Unsigned8	Read/Write	
AXIS#.UNIT.VROTARY	6607	Unsigned8	Read/Write	
AXIS#.UNIT.LABEL	6608	String	Read/Write	
AXIS#.WS.ARM	6700	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
AXIS#.WS.DISTMAX	6701	Signed32	Read/Write	Position
AXIS#.WS.DISTMIN	6702	Signed32	Read/Write	Position
AXIS#.WS.IMAX	6703	Float	Read/Write	
AXIS#.WS.MODE	6704	Unsigned8	Read/Write	
AXIS#.WS.NUMLOOPS	6705	Unsigned8	Read/Write	
AXIS#.WS.STATE	6706	Unsigned8	Read Only	
AXIS#.WS.T	6707	Unsigned8	Read/Write	
AXIS#.WS.TDELAY1	6708	Unsigned8	Read/Write	
AXIS#.WS.TDELAY2	6709	Unsigned8	Read/Write	
AXIS#.WS.TDELAY3	6710	Unsigned16	Read/Write	
AXIS#.WS.VTHRESH	6711	Float	Read/Write	Velocity
AXIS#.WS.DISARM	6712	Unsigned8	Read/Write	
AXIS#.WS.FREQ	6713	Float	Read/Write	
AXIS#.WS.TDELAY4	6714	Unsigned16	Read/Write	
AXIS#.WS.CHECKT	6715	Unsigned16	Read/Write	
AXIS#.WS.CHECKV	6716	Float	Read/Write	Velocity
AXIS#.WS.TSTANDSTILL	6717	Unsigned16	Read/Write	
AXIS#.WS.TIRAMP	6718	Unsigned16	Read/Write	
AXIS#.WS.CHECKMODE	6719	Unsigned8	Read/Write	
AXIS#.FAULT1	6800	Unsigned16	Read Only	
AXIS#.FAULT2	6801	Unsigned16	Read Only	
AXIS#.FAULT3	6802	Unsigned16	Read Only	
AXIS#.FAULT4	6803	Unsigned16	Read Only	
AXIS#.FAULT5	6804	Unsigned16	Read Only	
AXIS#.FAULT6	6805	Unsigned16	Read Only	
AXIS#.FAULT7	6806	Unsigned16	Read Only	
AXIS#.FAULT8	6807	Unsigned16	Read Only	
AXIS#.FAULT9	6808	Unsigned16	Read Only	
AXIS#.FAULT10	6809	Unsigned16	Read Only	
AXIS#.JOG.ACC	6900	Float	Read/Write	Acceleration
AXIS#.JOG.DEC	6901	Float	Read/Write	Acceleration
AXIS#.JOG.MOVEN	6902	Unsigned8	Read/Write	
AXIS#.JOG.MOVEP	6903	Unsigned8	Read/Write	
AXIS#.JOG.V	6904	Float	Read/Write	Velocity

Parameter	PNU	Data Type	Access	Units
AXIS#.HWEN.MODE	7000	Unsigned8	Read/Write	
AXIS#.HWEN.SOURCE	7001	Unsigned8	Read/Write	
AXIS#.HWEN.STATE	7002	Unsigned8	Read Only	
AXIS#.SETTLE.P	7100	Signed32	Read/Write	Position
AXIS#.SETTLE.V	7101	Float	Read/Write	Velocity
AXIS#.FAULT6004.ACTION	7200	Unsigned8	Read/Write	
AXIS#.SAFE.STO.A	10000	Unsigned8	Read Only	
AXIS#.SAFE.STO.B	10001	Unsigned8	Read Only	
AXIS#.SAFE.STO.ACTIVE	10002	Unsigned8	Read Only	
AXIS#.SAFE.STO.REPORTFAULT	10003	Unsigned8	Read/Write	
AXIS#.SAFE.SBT.MOVE	10004	Unsigned8	Read/Write	
AXIS#.SAFE.SBT.STATE	10005	Unsigned8	Read Only	
AXIS#.SAFE.SBT.ERROR	10006	Unsigned8	Read Only	
BRAKE1.SAFE.SBT.STATE	10007	Unsigned8	Read Only	
AXIS#.SAFE.FUNCTIONSTATUS	10008	Unsigned32	Read Only	
BRAKE1.SAFE.SBT.ELAPSED	10009	Unsigned32	Read Only	
BRAKE2.SAFE.SBT.ELAPSED	10010	Unsigned32	Read Only	
DIN1.STATE	20000	Unsigned8	Read Only	
DIN2.STATE	20001	Unsigned8	Read Only	
DIN3.STATE	20002	Unsigned8	Read Only	
DIN4.STATE	20003	Unsigned8	Read Only	
DIN5.STATE	20004	Unsigned8	Read Only	
DIN6.STATE	20005	Unsigned8	Read Only	
DIN7.STATE	20006	Unsigned8	Read Only	
DIN8.STATE	20007	Unsigned8	Read Only	
DIN9.STATE	20008	Unsigned8	Read Only	
DIN10.STATE	20009	Unsigned8	Read Only	
DIN11.STATE	20010	Unsigned8	Read Only	
DIN12.STATE	20011	Unsigned8	Read Only	
DIN1.INV	20050	Unsigned8	Read/Write	
DIN2.INV	20051	Unsigned8	Read/Write	
DIN3.INV	20052	Unsigned8	Read/Write	
DIN4.INV	20053	Unsigned8	Read/Write	
DIN5.INV	20054	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
DIN6.INV	20055	Unsigned8	Read/Write	
DIN7.INV	20056	Unsigned8	Read/Write	
DIN8.INV	20057	Unsigned8	Read/Write	
DIN9.INV	20058	Unsigned8	Read/Write	
DIN10.INV	20059	Unsigned8	Read/Write	
DIN11.INV	20060	Unsigned8	Read/Write	
DIN12.INV	20061	Unsigned8	Read/Write	
DIN1.FILTER	20100	Unsigned8	Read/Write	
DIN2.FILTER	20101	Unsigned8	Read/Write	
DIN3.FILTER	20102	Unsigned8	Read/Write	
DIN4.FILTER	20103	Unsigned8	Read/Write	
DIN5.FILTER	20104	Unsigned8	Read/Write	
DIN6.FILTER	20105	Unsigned8	Read/Write	
DIN7.FILTER	20106	Unsigned8	Read/Write	
DIN8.FILTER	20107	Unsigned8	Read/Write	
DIN9.FILTER	20108	Unsigned8	Read/Write	
DIN10.FILTER	20109	Unsigned8	Read/Write	
DIN11.FILTER	20110	Unsigned8	Read/Write	
DIN12.FILTER	20111	Unsigned8	Read/Write	
DOUT1.STATE	21000	Unsigned8	Read Only	
DOUT2.STATE	21001	Unsigned8	Read Only	
DOUT3.STATE	21002	Unsigned8	Read Only	
DOUT4.STATE	21003	Unsigned8	Read Only	
DOUT5.STATE	21004	Unsigned8	Read Only	
DOUT6.STATE	21005	Unsigned8	Read Only	
DOUT7.STATE	21006	Unsigned8	Read Only	
DOUT8.STATE	21007	Unsigned8	Read Only	
DOUT9.STATE	21008	Unsigned8	Read Only	
DOUT1.STATEU	21050	Unsigned8	Read/Write	
DOUT2.STATEU	21051	Unsigned8	Read/Write	
DOUT3.STATEU	21052	Unsigned8	Read/Write	
DOUT4.STATEU	21053	Unsigned8	Read/Write	
DOUT5.STATEU	21054	Unsigned8	Read/Write	
DOUT6.STATEU	21055	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
DOUT7.STATEU	21056	Unsigned8	Read/Write	
DOUT8.STATEU	21057	Unsigned8	Read/Write	
DOUT9.STATEU	21058	Unsigned8	Read/Write	
DOUT1.SOURCE	21150	Unsigned8	Read/Write	
DOUT2.SOURCE	21151	Unsigned8	Read/Write	
DOUT3.SOURCE	21152	Unsigned8	Read/Write	
DOUT4.SOURCE	21153	Unsigned8	Read/Write	
DOUT5.SOURCE	21154	Unsigned8	Read/Write	
DOUT6.SOURCE	21155	Unsigned8	Read/Write	
DOUT7.SOURCE	21156	Unsigned8	Read/Write	
DOUT8.SOURCE	21157	Unsigned8	Read/Write	
DOUT9.SOURCE	21158	Unsigned8	Read/Write	
DOUT1.SOURCEID	21200	Unsigned8	Read/Write	
DOUT2.SOURCEID	21201	Unsigned8	Read/Write	
DOUT3.SOURCEID	21202	Unsigned8	Read/Write	
DOUT4.SOURCEID	21203	Unsigned8	Read/Write	
DOUT5.SOURCEID	21204	Unsigned8	Read/Write	
DOUT6.SOURCEID	21205	Unsigned8	Read/Write	
DOUT7.SOURCEID	21206	Unsigned8	Read/Write	
DOUT8.SOURCEID	21207	Unsigned8	Read/Write	
DOUT9.SOURCEID	21208	Unsigned8	Read/Write	
DIO1.STATE	22000	Unsigned8	Read Only	
DIO2.STATE	22001	Unsigned8	Read Only	
DIO3.STATE	22002	Unsigned8	Read Only	
DIO4.STATE	22003	Unsigned8	Read Only	
DIO5.STATE	22004	Unsigned8	Read Only	
DIO6.STATE	22005	Unsigned8	Read Only	
DIO1.DIR	22050	Unsigned8	Read/Write	
DIO2.DIR	22051	Unsigned8	Read/Write	
DIO3.DIR	22052	Unsigned8	Read/Write	
DIO4.DIR	22053	Unsigned8	Read/Write	
DIO5.DIR	22054	Unsigned8	Read/Write	
DIO6.DIR	22055	Unsigned8	Read/Write	
DIO1.INV	22100	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
DIO2.INV	22101	Unsigned8	Read/Write	
DIO3.INV	22102	Unsigned8	Read/Write	
DIO4.INV	22103	Unsigned8	Read/Write	
DIO5.INV	22104	Unsigned8	Read/Write	
DIO6.INV	22105	Unsigned8	Read/Write	
DIO1.STATEU	22150	Unsigned8	Read/Write	
DIO2.STATEU	22151	Unsigned8	Read/Write	
DIO3.STATEU	22152	Unsigned8	Read/Write	
DIO4.STATEU	22153	Unsigned8	Read/Write	
DIO5.STATEU	22154	Unsigned8	Read/Write	
DIO6.STATEU	22155	Unsigned8	Read/Write	
DIO1.SOURCE	22200	Unsigned8	Read/Write	
DIO2.SOURCE	22201	Unsigned8	Read/Write	
DIO3.SOURCE	22202	Unsigned8	Read/Write	
DIO4.SOURCE	22203	Unsigned8	Read/Write	
DIO5.SOURCE	22204	Unsigned8	Read/Write	
DIO6.SOURCE	22205	Unsigned8	Read/Write	
DIO1.FILTER	22250	Unsigned8	Read/Write	
DIO2.FILTER	22251	Unsigned8	Read/Write	
DIO3.FILTER	22252	Unsigned8	Read/Write	
DIO4.FILTER	22253	Unsigned8	Read/Write	
DIO5.FILTER	22254	Unsigned8	Read/Write	
DIO6.FILTER	22255	Unsigned8	Read/Write	
DIO1.SOURCEID	22300	Unsigned8	Read/Write	
DIO2.SOURCEID	22301	Unsigned8	Read/Write	
DIO3.SOURCEID	22302	Unsigned8	Read/Write	
DIO4.SOURCEID	22303	Unsigned8	Read/Write	
DIO5.SOURCEID	22304	Unsigned8	Read/Write	
DIO6.SOURCEID	22305	Unsigned8	Read/Write	
DIO1.TERM	22350	Unsigned8	Read/Write	
DIO2.TERM	22351	Unsigned8	Read/Write	
DIO3.TERM	22352	Unsigned8	Read/Write	
DIO4.TERM	22353	Unsigned8	Read/Write	
DIO5.TERM	22354	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
DIO6.TERM	22355	Unsigned8	Read/Write	
AIN1.CUTOFF	23000	Float	Read/Write	
AIN2.CUTOFF	23001	Float	Read/Write	
AIN1.OFFSET	23050	Float	Read/Write	
AIN2.OFFSET	23051	Float	Read/Write	
AIN1.VALUE	23100	Float	Read Only	
AIN2.VALUE	23101	Float	Read Only	
AIN1.DEADBAND	23150	Float	Read/Write	
AIN2.DEADBAND	23151	Float	Read/Write	
AIN1.DEADBANDMODE	23200	Unsigned16	Read/Write	
AIN2.DEADBANDMODE	23201	Unsigned16	Read/Write	
AIN1.ZERO	23250	Unsigned8	Read/Write	
AIN2.ZERO	23251	Unsigned8	Read/Write	
AOUT1.VALUE	24000	Float	Read Only	
AOUT2.VALUE	24001	Float	Read Only	
AOUT1.SOURCE	24050	Unsigned8	Read/Write	
AOUT2.SOURCE	24051	Unsigned8	Read/Write	
AOUT1.CUTOFF	24100	Float	Read/Write	
AOUT2.CUTOFF	24101	Float	Read/Write	
AOUT1 Fbus Value	24150	Float	Read/Write	
AOUT2 Fbus Value	24151	Float	Read/Write	
AOUT1.OFFSET	24200	Float	Read/Write	
AOUT2.OFFSET	24201	Float	Read/Write	
FB1.IDENTIFIED	30000	Signed16	Read Only	
FB2.IDENTIFIED	30001	Signed16	Read Only	
FB3.IDENTIFIED	30002	Signed16	Read Only	
FB4.IDENTIFIED	30003	Signed16	Read Only	
FB5.IDENTIFIED	30004	Signed16	Read Only	
FB1.LASTIDENTIFIED	30020	Signed16	Read/Write	
FB2.LASTIDENTIFIED	30021	Signed16	Read/Write	
FB3.LASTIDENTIFIED	30022	Signed16	Read/Write	
FB4.LASTIDENTIFIED	30023	Signed16	Read/Write	
FB5.LASTIDENTIFIED	30024	Signed16	Read/Write	
FB1.SELECT	30040	Signed16	Read/Write	

Parameter	PNU	Data Type	Access	Units
FB2.SELECT	30041	Signed16	Read/Write	
FB3.SELECT	30042	Signed16	Read/Write	
FB4.SELECT	30043	Signed16	Read/Write	
FB5.SELECT	30044	Signed16	Read/Write	
FB1.HALLSTATE	30060	Unsigned8	Read Only	
FB2.HALLSTATE	30061	Unsigned8	Read Only	
FB3.HALLSTATE	30062	Unsigned8	Read Only	
FB4.HALLSTATE	30063	Unsigned8	Read Only	
FB5.HALLSTATE	30064	Unsigned8	Read Only	
FB1.ENCLINES	30080	Unsigned32	Read/Write	
FB2.ENCLINES	30081	Unsigned32	Read/Write	
FB3.ENCLINES	30082	Unsigned32	Read/Write	
FB4.ENCLINES	30083	Unsigned32	Read/Write	
FB5.ENCLINES	30084	Unsigned32	Read/Write	
FB1.POLES	30100	Unsigned16	Read/Write	
FB2.POLES	30101	Unsigned16	Read/Write	
FB3.POLES	30102	Unsigned16	Read/Write	
FB4.POLES	30103	Unsigned16	Read/Write	
FB5.POLES	30104	Unsigned16	Read/Write	
FB1.RES	30120	Unsigned32	Read Only	
FB2.RES	30121	Unsigned32	Read Only	
FB3.RES	30122	Unsigned32	Read Only	
FB4.RES	30123	Unsigned32	Read Only	
FB5.RES	30124	Unsigned32	Read Only	
FB1.RESKTR	30140	Float	Read/Write	
FB2.RESKTR	30141	Float	Read/Write	
FB3.RESKTR	30142	Float	Read/Write	
FB4.RESKTR	30143	Float	Read/Write	
FB5.RESKTR	30144	Float	Read/Write	
FB1.RESREFPHASE	30160	Float	Read/Write	
FB2.RESREFPHASE	30161	Float	Read/Write	
FB3.RESREFPHASE	30162	Float	Read/Write	
FB4.RESREFPHASE	30163	Float	Read/Write	
FB5.RESREFPHASE	30164	Float	Read/Write	

Parameter	PNU	Data Type	Access	Units
FB1.TRACKINGCAL	30180	Unsigned8	Read/Write	
FB2.TRACKINGCAL	30181	Unsigned8	Read/Write	
FB3.TRACKINGCAL	30182	Unsigned8	Read/Write	
FB4.TRACKINGCAL	30183	Unsigned8	Read/Write	
FB5.TRACKINGCAL	30184	Unsigned8	Read/Write	
FB1.CALTHRESH	30200	Unsigned16	Read/Write	
FB2.CALTHRESH	30201	Unsigned16	Read/Write	
FB3.CALTHRESH	30202	Unsigned16	Read/Write	
FB4.CALTHRESH	30203	Unsigned16	Read/Write	
FB5.CALTHRESH	30204	Unsigned16	Read/Write	
FB1.MECHPOS	30220	Unsigned32	Read Only	
FB2.MECHPOS	30221	Unsigned32	Read Only	
FB3.MECHPOS	30222	Unsigned32	Read Only	
FB4.MECHPOS	30223	Unsigned32	Read Only	
FB5.MECHPOS	30224	Unsigned32	Read Only	
FB1.P	30240	Signed32	Read Only	Position
FB2.P	30241	Signed32	Read Only	Position
FB3.P	30242	Signed32	Read Only	Position
FB4.P	30243	Signed32	Read Only	Position
FB5.P	30244	Signed32	Read Only	Position
FB1.SIGNALAMPLITUDE	30260	Float	Read Only	
FB2.SIGNALAMPLITUDE	30261	Float	Read Only	
FB3.SIGNALAMPLITUDE	30262	Float	Read Only	
FB4.SIGNALAMPLITUDE	30263	Float	Read Only	
FB5.SIGNALAMPLITUDE	30264	Float	Read Only	
FB1.SIGNALCOS	30280	Float	Read Only	
FB2.SIGNALCOS	30281	Float	Read Only	
FB3.SIGNALCOS	30282	Float	Read Only	
FB4.SIGNALCOS	30283	Float	Read Only	
FB5.SIGNALCOS	30284	Float	Read Only	
FB1.SIGNALSIN	30300	Float	Read Only	
FB2.SIGNALSIN	30301	Float	Read Only	
FB3.SIGNALSIN	30302	Float	Read Only	
FB4.SIGNALSIN	30303	Float	Read Only	

Parameter	PNU	Data Type	Access	Units
FB5.SIGNALSIN	30304	Float	Read Only	
FB1.SINGLETURNBITS	30320	Unsigned8	Read/Write	
FB2.SINGLETURNBITS	30321	Unsigned8	Read/Write	
FB3.SINGLETURNBITS	30322	Unsigned8	Read/Write	
FB4.SINGLETURNBITS	30323	Unsigned8	Read/Write	
FB5.SINGLETURNBITS	30324	Unsigned8	Read/Write	
FB1.MULTITURNBITS	30340	Unsigned8	Read/Write	
FB2.MULTITURNBITS	30341	Unsigned8	Read/Write	
FB3.MULTITURNBITS	30342	Unsigned8	Read/Write	
FB4.MULTITURNBITS	30343	Unsigned8	Read/Write	
FB5.MULTITURNBITS	30344	Unsigned8	Read/Write	
FB1.MECHTYPE	30360	Unsigned8	Read/Write	
FB2.MECHTYPE	30361	Unsigned8	Read/Write	
FB3.MECHTYPE	30362	Unsigned8	Read/Write	
FB4.MECHTYPE	30363	Unsigned8	Read/Write	
FB5.MECHTYPE	30364	Unsigned8	Read/Write	
FB1.LINEPITCH	30380	Float	Read/Write	
FB2.LINEPITCH	30381	Float	Read/Write	
FB3.LINEPITCH	30382	Float	Read/Write	
FB4.LINEPITCH	30383	Float	Read/Write	
FB5.LINEPITCH	30384	Float	Read/Write	
FB1.BITS	30400	Unsigned16	Read/Write	
FB2.BITS	30401	Unsigned16	Read/Write	
FB3.BITS	30402	Unsigned16	Read/Write	
FB4.BITS	30403	Unsigned16	Read/Write	
FB5.BITS	30404	Unsigned16	Read/Write	
FB1.FAULTS	30420	Unsigned32 [5]	Read Only	
FB2.FAULTS	30421	Unsigned32 [5]	Read Only	
FB3.FAULTS	30422	Unsigned32 [5]	Read Only	
FB4.FAULTS	30423	Unsigned32 [5]	Read Only	
FB5.FAULTS	30424	Unsigned32 [5]	Read Only	

Parameter	PNU	Data Type	Access	Units
CAP1.ARM	31000	Unsigned8	Read/Write	
CAP2.ARM	31001	Unsigned8	Read/Write	
CAP1.COUNT	31010	Unsigned16	Read Only	
CAP2.COUNT	31011	Unsigned16	Read Only	
CAP1.EDGE	31020	Unsigned8	Read/Write	
CAP2.EDGE	31021	Unsigned8	Read/Write	
CAP1.P	31030	Signed32	Read Only	Position
CAP2.P	31031	Signed32	Read Only	Position
CAP1.PREEDGE	31040	Unsigned8	Read/Write	
CAP2.PREEDGE	31041	Unsigned8	Read/Write	
CAP1.PREMODE	31050	Unsigned8	Read/Write	
CAP2.PREMODE	31051	Unsigned8	Read/Write	
CAP1.PRESELECT	31060	Unsigned8	Read/Write	
CAP2.PRESELECT	31061	Unsigned8	Read/Write	
CAP1.REARM	31070	Unsigned8	Read/Write	
CAP2.REARM	31071	Unsigned8	Read/Write	
CAP1.SOURCE	31080	Unsigned8	Read/Write	
CAP2.SOURCE	31081	Unsigned8	Read/Write	
CAP1.STATE	31090	Unsigned8	Read Only	
CAP2.STATE	31091	Unsigned8	Read Only	
CAP1.T	31100	Unsigned32	Read Only	
CAP2.T	31101	Unsigned32	Read Only	
CAP1.TRIGGER	31110	Unsigned8	Read/Write	
CAP2.TRIGGER	31111	Unsigned8	Read/Write	
EEO1.DIR	32000	Unsigned8	Read/Write	
EEO2.DIR	32001	Unsigned8	Read/Write	
EEO1.LINES	32005	Unsigned32	Read/Write	
EEO2.LINES	32006	Unsigned32	Read/Write	
EEO1.MODE	32010	Unsigned8	Read/Write	
EEO2.MODE	32011	Unsigned8	Read/Write	
EEO1.PULSEWIDTH	32015	Float	Read/Write	
EEO2.PULSEWIDTH	32016	Float	Read/Write	
EEO1.SOURCE	32020	Unsigned8	Read/Write	
EEO2.SOURCE	32021	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
EEO1.ZMODE	32025	Unsigned8	Read/Write	
EEO1.ZOFFSET	32030	Unsigned32	Read/Write	
EEO1.ZP	32035	Signed32	Read/Write	
ACTION.RUNNING	40000	Unsigned32	Read Only	
ACTION1.ACTIVE	40100	Unsigned8	Read/Write	
ACTION2.ACTIVE	40101	Unsigned8	Read/Write	
ACTION3.ACTIVE	40102	Unsigned8	Read/Write	
ACTION4.ACTIVE	40103	Unsigned8	Read/Write	
ACTION5.ACTIVE	40104	Unsigned8	Read/Write	
ACTION6.ACTIVE	40105	Unsigned8	Read/Write	
ACTION7.ACTIVE	40106	Unsigned8	Read/Write	
ACTION8.ACTIVE	40107	Unsigned8	Read/Write	
ACTION9.ACTIVE	40108	Unsigned8	Read/Write	
ACTION10.ACTIVE	40109	Unsigned8	Read/Write	
ACTION11.ACTIVE	40110	Unsigned8	Read/Write	
ACTION12.ACTIVE	40111	Unsigned8	Read/Write	
ACTION13.ACTIVE	40112	Unsigned8	Read/Write	
ACTION14.ACTIVE	40113	Unsigned8	Read/Write	
ACTION15.ACTIVE	40114	Unsigned8	Read/Write	
ACTION16.ACTIVE	40115	Unsigned8	Read/Write	
ACTION17.ACTIVE	40116	Unsigned8	Read/Write	
ACTION18.ACTIVE	40117	Unsigned8	Read/Write	
ACTION19.ACTIVE	40118	Unsigned8	Read/Write	
ACTION20.ACTIVE	40119	Unsigned8	Read/Write	
ACTION21.ACTIVE	40120	Unsigned8	Read/Write	
ACTION22.ACTIVE	40121	Unsigned8	Read/Write	
ACTION23.ACTIVE	40122	Unsigned8	Read/Write	
ACTION24.ACTIVE	40123	Unsigned8	Read/Write	
ACTION25.ACTIVE	40124	Unsigned8	Read/Write	
ACTION26.ACTIVE	40125	Unsigned8	Read/Write	
ACTION27.ACTIVE	40126	Unsigned8	Read/Write	
ACTION28.ACTIVE	40127	Unsigned8	Read/Write	
ACTION29.ACTIVE	40128	Unsigned8	Read/Write	
ACTION30.ACTIVE	40129	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
ACTION31.ACTIVE	40130	Unsigned8	Read/Write	
ACTION32.ACTIVE	40131	Unsigned8	Read/Write	
ACTION1.CONDITION	40200	Unsigned8	Read/Write	
ACTION2.CONDITION	40201	Unsigned8	Read/Write	
ACTION3.CONDITION	40202	Unsigned8	Read/Write	
ACTION4.CONDITION	40203	Unsigned8	Read/Write	
ACTION5.CONDITION	40204	Unsigned8	Read/Write	
ACTION6.CONDITION	40205	Unsigned8	Read/Write	
ACTION7.CONDITION	40206	Unsigned8	Read/Write	
ACTION8.CONDITION	40207	Unsigned8	Read/Write	
ACTION9.CONDITION	40208	Unsigned8	Read/Write	
ACTION10.CONDITION	40209	Unsigned8	Read/Write	
ACTION11.CONDITION	40210	Unsigned8	Read/Write	
ACTION12.CONDITION	40211	Unsigned8	Read/Write	
ACTION13.CONDITION	40212	Unsigned8	Read/Write	
ACTION14.CONDITION	40213	Unsigned8	Read/Write	
ACTION15.CONDITION	40214	Unsigned8	Read/Write	
ACTION16.CONDITION	40215	Unsigned8	Read/Write	
ACTION17.CONDITION	40216	Unsigned8	Read/Write	
ACTION18.CONDITION	40217	Unsigned8	Read/Write	
ACTION19.CONDITION	40218	Unsigned8	Read/Write	
ACTION20.CONDITION	40219	Unsigned8	Read/Write	
ACTION21.CONDITION	40220	Unsigned8	Read/Write	
ACTION22.CONDITION	40221	Unsigned8	Read/Write	
ACTION23.CONDITION	40222	Unsigned8	Read/Write	
ACTION24.CONDITION	40223	Unsigned8	Read/Write	
ACTION25.CONDITION	40224	Unsigned8	Read/Write	
ACTION26.CONDITION	40225	Unsigned8	Read/Write	
ACTION27.CONDITION	40226	Unsigned8	Read/Write	
ACTION28.CONDITION	40227	Unsigned8	Read/Write	
ACTION29.CONDITION	40228	Unsigned8	Read/Write	
ACTION30.CONDITION	40229	Unsigned8	Read/Write	
ACTION31.CONDITION	40230	Unsigned8	Read/Write	
ACTION32.CONDITION	40231	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
ACTION1.CONDITIONVALUE	40300	Float	Read/Write	
ACTION2.CONDITIONVALUE	40301	Float	Read/Write	
ACTION3.CONDITIONVALUE	40302	Float	Read/Write	
ACTION4.CONDITIONVALUE	40303	Float	Read/Write	
ACTION5.CONDITIONVALUE	40304	Float	Read/Write	
ACTION6.CONDITIONVALUE	40305	Float	Read/Write	
ACTION7.CONDITIONVALUE	40306	Float	Read/Write	
ACTION8.CONDITIONVALUE	40307	Float	Read/Write	
ACTION9.CONDITIONVALUE	40308	Float	Read/Write	
ACTION10.CONDITIONVALUE	40309	Float	Read/Write	
ACTION11.CONDITIONVALUE	40310	Float	Read/Write	
ACTION12.CONDITIONVALUE	40311	Float	Read/Write	
ACTION13.CONDITIONVALUE	40312	Float	Read/Write	
ACTION14.CONDITIONVALUE	40313	Float	Read/Write	
ACTION15.CONDITIONVALUE	40314	Float	Read/Write	
ACTION16.CONDITIONVALUE	40315	Float	Read/Write	
ACTION17.CONDITIONVALUE	40316	Float	Read/Write	
ACTION18.CONDITIONVALUE	40317	Float	Read/Write	
ACTION19.CONDITIONVALUE	40318	Float	Read/Write	
ACTION20.CONDITIONVALUE	40319	Float	Read/Write	
ACTION21.CONDITIONVALUE	40320	Float	Read/Write	
ACTION22.CONDITIONVALUE	40321	Float	Read/Write	
ACTION23.CONDITIONVALUE	40322	Float	Read/Write	
ACTION24.CONDITIONVALUE	40323	Float	Read/Write	
ACTION25.CONDITIONVALUE	40324	Float	Read/Write	
ACTION26.CONDITIONVALUE	40325	Float	Read/Write	
ACTION27.CONDITIONVALUE	40326	Float	Read/Write	
ACTION28.CONDITIONVALUE	40327	Float	Read/Write	
ACTION29.CONDITIONVALUE	40328	Float	Read/Write	
ACTION30.CONDITIONVALUE	40329	Float	Read/Write	
ACTION31.CONDITIONVALUE	40330	Float	Read/Write	
ACTION32.CONDITIONVALUE	40331	Float	Read/Write	
ACTION1.RUNCOUNT	40400	Unsigned32	Read Only	
ACTION2.RUNCOUNT	40401	Unsigned32	Read Only	

Parameter	PNU	Data Type	Access	Units
ACTION3.RUNCOUNT	40402	Unsigned32	Read Only	
ACTION4.RUNCOUNT	40403	Unsigned32	Read Only	
ACTION5.RUNCOUNT	40404	Unsigned32	Read Only	
ACTION6.RUNCOUNT	40405	Unsigned32	Read Only	
ACTION7.RUNCOUNT	40406	Unsigned32	Read Only	
ACTION8.RUNCOUNT	40407	Unsigned32	Read Only	
ACTION9.RUNCOUNT	40408	Unsigned32	Read Only	
ACTION10.RUNCOUNT	40409	Unsigned32	Read Only	
ACTION11.RUNCOUNT	40410	Unsigned32	Read Only	
ACTION12.RUNCOUNT	40411	Unsigned32	Read Only	
ACTION13.RUNCOUNT	40412	Unsigned32	Read Only	
ACTION14.RUNCOUNT	40413	Unsigned32	Read Only	
ACTION15.RUNCOUNT	40414	Unsigned32	Read Only	
ACTION16.RUNCOUNT	40415	Unsigned32	Read Only	
ACTION17.RUNCOUNT	40416	Unsigned32	Read Only	
ACTION18.RUNCOUNT	40417	Unsigned32	Read Only	
ACTION19.RUNCOUNT	40418	Unsigned32	Read Only	
ACTION20.RUNCOUNT	40419	Unsigned32	Read Only	
ACTION21.RUNCOUNT	40420	Unsigned32	Read Only	
ACTION22.RUNCOUNT	40421	Unsigned32	Read Only	
ACTION23.RUNCOUNT	40422	Unsigned32	Read Only	
ACTION24.RUNCOUNT	40423	Unsigned32	Read Only	
ACTION25.RUNCOUNT	40424	Unsigned32	Read Only	
ACTION26.RUNCOUNT	40425	Unsigned32	Read Only	
ACTION27.RUNCOUNT	40426	Unsigned32	Read Only	
ACTION28.RUNCOUNT	40427	Unsigned32	Read Only	
ACTION29.RUNCOUNT	40428	Unsigned32	Read Only	
ACTION30.RUNCOUNT	40429	Unsigned32	Read Only	
ACTION31.RUNCOUNT	40430	Unsigned32	Read Only	
ACTION32.RUNCOUNT	40431	Unsigned32	Read Only	
ACTION1.SOURCE	40500	Unsigned8	Read/Write	
ACTION2.SOURCE	40501	Unsigned8	Read/Write	
ACTION3.SOURCE	40502	Unsigned8	Read/Write	
ACTION4.SOURCE	40503	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
ACTION5.SOURCE	40504	Unsigned8	Read/Write	
ACTION6.SOURCE	40505	Unsigned8	Read/Write	
ACTION7.SOURCE	40506	Unsigned8	Read/Write	
ACTION8.SOURCE	40507	Unsigned8	Read/Write	
ACTION9.SOURCE	40508	Unsigned8	Read/Write	
ACTION10.SOURCE	40509	Unsigned8	Read/Write	
ACTION11.SOURCE	40510	Unsigned8	Read/Write	
ACTION12.SOURCE	40511	Unsigned8	Read/Write	
ACTION13.SOURCE	40512	Unsigned8	Read/Write	
ACTION14.SOURCE	40513	Unsigned8	Read/Write	
ACTION15.SOURCE	40514	Unsigned8	Read/Write	
ACTION16.SOURCE	40515	Unsigned8	Read/Write	
ACTION17.SOURCE	40516	Unsigned8	Read/Write	
ACTION18.SOURCE	40517	Unsigned8	Read/Write	
ACTION19.SOURCE	40518	Unsigned8	Read/Write	
ACTION20.SOURCE	40519	Unsigned8	Read/Write	
ACTION21.SOURCE	40520	Unsigned8	Read/Write	
ACTION22.SOURCE	40521	Unsigned8	Read/Write	
ACTION23.SOURCE	40522	Unsigned8	Read/Write	
ACTION24.SOURCE	40523	Unsigned8	Read/Write	
ACTION25.SOURCE	40524	Unsigned8	Read/Write	
ACTION26.SOURCE	40525	Unsigned8	Read/Write	
ACTION27.SOURCE	40526	Unsigned8	Read/Write	
ACTION28.SOURCE	40527	Unsigned8	Read/Write	
ACTION29.SOURCE	40528	Unsigned8	Read/Write	
ACTION30.SOURCE	40529	Unsigned8	Read/Write	
ACTION31.SOURCE	40530	Unsigned8	Read/Write	
ACTION32.SOURCE	40531	Unsigned8	Read/Write	
ACTION1.SOURCEID	40600	Unsigned8	Read/Write	
ACTION2.SOURCEID	40601	Unsigned8	Read/Write	
ACTION3.SOURCEID	40602	Unsigned8	Read/Write	
ACTION4.SOURCEID	40603	Unsigned8	Read/Write	
ACTION5.SOURCEID	40604	Unsigned8	Read/Write	
ACTION6.SOURCEID	40605	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
ACTION7.SOURCEID	40606	Unsigned8	Read/Write	
ACTION8.SOURCEID	40607	Unsigned8	Read/Write	
ACTION9.SOURCEID	40608	Unsigned8	Read/Write	
ACTION10.SOURCEID	40609	Unsigned8	Read/Write	
ACTION11.SOURCEID	40610	Unsigned8	Read/Write	
ACTION12.SOURCEID	40611	Unsigned8	Read/Write	
ACTION13.SOURCEID	40612	Unsigned8	Read/Write	
ACTION14.SOURCEID	40613	Unsigned8	Read/Write	
ACTION15.SOURCEID	40614	Unsigned8	Read/Write	
ACTION16.SOURCEID	40615	Unsigned8	Read/Write	
ACTION17.SOURCEID	40616	Unsigned8	Read/Write	
ACTION18.SOURCEID	40617	Unsigned8	Read/Write	
ACTION19.SOURCEID	40618	Unsigned8	Read/Write	
ACTION20.SOURCEID	40619	Unsigned8	Read/Write	
ACTION21.SOURCEID	40620	Unsigned8	Read/Write	
ACTION22.SOURCEID	40621	Unsigned8	Read/Write	
ACTION23.SOURCEID	40622	Unsigned8	Read/Write	
ACTION24.SOURCEID	40623	Unsigned8	Read/Write	
ACTION25.SOURCEID	40624	Unsigned8	Read/Write	
ACTION26.SOURCEID	40625	Unsigned8	Read/Write	
ACTION27.SOURCEID	40626	Unsigned8	Read/Write	
ACTION28.SOURCEID	40627	Unsigned8	Read/Write	
ACTION29.SOURCEID	40628	Unsigned8	Read/Write	
ACTION30.SOURCEID	40629	Unsigned8	Read/Write	
ACTION31.SOURCEID	40630	Unsigned8	Read/Write	
ACTION32.SOURCEID	40631	Unsigned8	Read/Write	
ACTION1.SOURCEPARAM	40700	Float	Read/Write	
ACTION2.SOURCEPARAM	40701	Float	Read/Write	
ACTION3.SOURCEPARAM	40702	Float	Read/Write	
ACTION4.SOURCEPARAM	40703	Float	Read/Write	
ACTION5.SOURCEPARAM	40704	Float	Read/Write	
ACTION6.SOURCEPARAM	40705	Float	Read/Write	
ACTION7.SOURCEPARAM	40706	Float	Read/Write	
ACTION8.SOURCEPARAM	40707	Float	Read/Write	

Parameter	PNU	Data Type	Access	Units
ACTION9.SOURCEPARAM	40708	Float	Read/Write	
ACTION10.SOURCEPARAM	40709	Float	Read/Write	
ACTION11.SOURCEPARAM	40710	Float	Read/Write	
ACTION12.SOURCEPARAM	40711	Float	Read/Write	
ACTION13.SOURCEPARAM	40712	Float	Read/Write	
ACTION14.SOURCEPARAM	40713	Float	Read/Write	
ACTION15.SOURCEPARAM	40714	Float	Read/Write	
ACTION16.SOURCEPARAM	40715	Float	Read/Write	
ACTION17.SOURCEPARAM	40716	Float	Read/Write	
ACTION18.SOURCEPARAM	40717	Float	Read/Write	
ACTION19.SOURCEPARAM	40718	Float	Read/Write	
ACTION20.SOURCEPARAM	40719	Float	Read/Write	
ACTION21.SOURCEPARAM	40720	Float	Read/Write	
ACTION22.SOURCEPARAM	40721	Float	Read/Write	
ACTION23.SOURCEPARAM	40722	Float	Read/Write	
ACTION24.SOURCEPARAM	40723	Float	Read/Write	
ACTION25.SOURCEPARAM	40724	Float	Read/Write	
ACTION26.SOURCEPARAM	40725	Float	Read/Write	
ACTION27.SOURCEPARAM	40726	Float	Read/Write	
ACTION28.SOURCEPARAM	40727	Float	Read/Write	
ACTION29.SOURCEPARAM	40728	Float	Read/Write	
ACTION30.SOURCEPARAM	40729	Float	Read/Write	
ACTION31.SOURCEPARAM	40730	Float	Read/Write	
ACTION32.SOURCEPARAM	40731	Float	Read/Write	
ACTION1.TASK	40800	Unsigned8	Read/Write	
ACTION2.TASK	40801	Unsigned8	Read/Write	
ACTION3.TASK	40802	Unsigned8	Read/Write	
ACTION4.TASK	40803	Unsigned8	Read/Write	
ACTION5.TASK	40804	Unsigned8	Read/Write	
ACTION6.TASK	40805	Unsigned8	Read/Write	
ACTION7.TASK	40806	Unsigned8	Read/Write	
ACTION8.TASK	40807	Unsigned8	Read/Write	
ACTION9.TASK	40808	Unsigned8	Read/Write	
ACTION10.TASK	40809	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
ACTION11.TASK	40810	Unsigned8	Read/Write	
ACTION12.TASK	40811	Unsigned8	Read/Write	
ACTION13.TASK	40812	Unsigned8	Read/Write	
ACTION14.TASK	40813	Unsigned8	Read/Write	
ACTION15.TASK	40814	Unsigned8	Read/Write	
ACTION16.TASK	40815	Unsigned8	Read/Write	
ACTION17.TASK	40816	Unsigned8	Read/Write	
ACTION18.TASK	40817	Unsigned8	Read/Write	
ACTION19.TASK	40818	Unsigned8	Read/Write	
ACTION20.TASK	40819	Unsigned8	Read/Write	
ACTION21.TASK	40820	Unsigned8	Read/Write	
ACTION22.TASK	40821	Unsigned8	Read/Write	
ACTION23.TASK	40822	Unsigned8	Read/Write	
ACTION24.TASK	40823	Unsigned8	Read/Write	
ACTION25.TASK	40824	Unsigned8	Read/Write	
ACTION26.TASK	40825	Unsigned8	Read/Write	
ACTION27.TASK	40826	Unsigned8	Read/Write	
ACTION28.TASK	40827	Unsigned8	Read/Write	
ACTION29.TASK	40828	Unsigned8	Read/Write	
ACTION30.TASK	40829	Unsigned8	Read/Write	
ACTION31.TASK	40830	Unsigned8	Read/Write	
ACTION32.TASK	40831	Unsigned8	Read/Write	
ACTION1.TASKID	40900	Unsigned8	Read/Write	
ACTION2.TASKID	40901	Unsigned8	Read/Write	
ACTION3.TASKID	40902	Unsigned8	Read/Write	
ACTION4.TASKID	40903	Unsigned8	Read/Write	
ACTION5.TASKID	40904	Unsigned8	Read/Write	
ACTION6.TASKID	40905	Unsigned8	Read/Write	
ACTION7.TASKID	40906	Unsigned8	Read/Write	
ACTION8.TASKID	40907	Unsigned8	Read/Write	
ACTION9.TASKID	40908	Unsigned8	Read/Write	
ACTION10.TASKID	40909	Unsigned8	Read/Write	
ACTION11.TASKID	40910	Unsigned8	Read/Write	
ACTION12.TASKID	40911	Unsigned8	Read/Write	

Parameter	PNU	Data Type	Access	Units
ACTION13.TASKID	40912	Unsigned8	Read/Write	
ACTION14.TASKID	40913	Unsigned8	Read/Write	
ACTION15.TASKID	40914	Unsigned8	Read/Write	
ACTION16.TASKID	40915	Unsigned8	Read/Write	
ACTION17.TASKID	40916	Unsigned8	Read/Write	
ACTION18.TASKID	40917	Unsigned8	Read/Write	
ACTION19.TASKID	40918	Unsigned8	Read/Write	
ACTION20.TASKID	40919	Unsigned8	Read/Write	
ACTION21.TASKID	40920	Unsigned8	Read/Write	
ACTION22.TASKID	40921	Unsigned8	Read/Write	
ACTION23.TASKID	40922	Unsigned8	Read/Write	
ACTION24.TASKID	40923	Unsigned8	Read/Write	
ACTION25.TASKID	40924	Unsigned8	Read/Write	
ACTION26.TASKID	40925	Unsigned8	Read/Write	
ACTION27.TASKID	40926	Unsigned8	Read/Write	
ACTION28.TASKID	40927	Unsigned8	Read/Write	
ACTION29.TASKID	40928	Unsigned8	Read/Write	
ACTION30.TASKID	40929	Unsigned8	Read/Write	
ACTION31.TASKID	40930	Unsigned8	Read/Write	
ACTION32.TASKID	40931	Unsigned8	Read/Write	
ACTION1.TASKPARAM	41000	Float	Read/Write	
ACTION2.TASKPARAM	41001	Float	Read/Write	
ACTION3.TASKPARAM	41002	Float	Read/Write	
ACTION4.TASKPARAM	41003	Float	Read/Write	
ACTION5.TASKPARAM	41004	Float	Read/Write	
ACTION6.TASKPARAM	41005	Float	Read/Write	
ACTION7.TASKPARAM	41006	Float	Read/Write	
ACTION8.TASKPARAM	41007	Float	Read/Write	
ACTION9.TASKPARAM	41008	Float	Read/Write	
ACTION10.TASKPARAM	41009	Float	Read/Write	
ACTION11.TASKPARAM	41010	Float	Read/Write	
ACTION12.TASKPARAM	41011	Float	Read/Write	
ACTION13.TASKPARAM	41012	Float	Read/Write	
ACTION14.TASKPARAM	41013	Float	Read/Write	

Parameter	PNU	Data Type	Access	Units
ACTION15.TASKPARAM	41014	Float	Read/Write	
ACTION16.TASKPARAM	41015	Float	Read/Write	
ACTION17.TASKPARAM	41016	Float	Read/Write	
ACTION18.TASKPARAM	41017	Float	Read/Write	
ACTION19.TASKPARAM	41018	Float	Read/Write	
ACTION20.TASKPARAM	41019	Float	Read/Write	
ACTION21.TASKPARAM	41020	Float	Read/Write	
ACTION22.TASKPARAM	41021	Float	Read/Write	
ACTION23.TASKPARAM	41022	Float	Read/Write	
ACTION24.TASKPARAM	41023	Float	Read/Write	
ACTION25.TASKPARAM	41024	Float	Read/Write	
ACTION26.TASKPARAM	41025	Float	Read/Write	
ACTION27.TASKPARAM	41026	Float	Read/Write	
ACTION28.TASKPARAM	41027	Float	Read/Write	
ACTION29.TASKPARAM	41028	Float	Read/Write	
ACTION30.TASKPARAM	41029	Float	Read/Write	
ACTION31.TASKPARAM	41030	Float	Read/Write	
ACTION32.TASKPARAM	41031	Float	Read/Write	
ACTION1.TASKTEXT	41100	String	Read/Write	
ACTION2.TASKTEXT	41101	String	Read/Write	
ACTION3.TASKTEXT	41102	String	Read/Write	
ACTION4.TASKTEXT	41103	String	Read/Write	
ACTION5.TASKTEXT	41104	String	Read/Write	
ACTION6.TASKTEXT	41105	String	Read/Write	
ACTION7.TASKTEXT	41106	String	Read/Write	
ACTION8.TASKTEXT	41107	String	Read/Write	
ACTION9.TASKTEXT	41108	String	Read/Write	
ACTION10.TASKTEXT	41109	String	Read/Write	
ACTION11.TASKTEXT	41110	String	Read/Write	
ACTION12.TASKTEXT	41111	String	Read/Write	
ACTION13.TASKTEXT	41112	String	Read/Write	
ACTION14.TASKTEXT	41113	String	Read/Write	
ACTION15.TASKTEXT	41114	String	Read/Write	
ACTION16.TASKTEXT	41115	String	Read/Write	

Parameter	PNU	Data Type	Access	Units
ACTION17.TASKTEXT	41116	String	Read/Write	
ACTION18.TASKTEXT	41117	String	Read/Write	
ACTION19.TASKTEXT	41118	String	Read/Write	
ACTION20.TASKTEXT	41119	String	Read/Write	
ACTION21.TASKTEXT	41120	String	Read/Write	
ACTION22.TASKTEXT	41121	String	Read/Write	
ACTION23.TASKTEXT	41122	String	Read/Write	
ACTION24.TASKTEXT	41123	String	Read/Write	
ACTION25.TASKTEXT	41124	String	Read/Write	
ACTION26.TASKTEXT	41125	String	Read/Write	
ACTION27.TASKTEXT	41126	String	Read/Write	
ACTION28.TASKTEXT	41127	String	Read/Write	
ACTION29.TASKTEXT	41128	String	Read/Write	
ACTION30.TASKTEXT	41129	String	Read/Write	
ACTION31.TASKTEXT	41130	String	Read/Write	
ACTION32.TASKTEXT	41131	String	Read/Write	
AXIS#.PROFINET.REFV	60000	Float	Read/Write	Velocity

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.com
Mail:	support@kollmorgen.com
Tel.:	+1 - 540 - 633 - 3545
Fax:	+1 - 540 - 639 - 4162

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

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
Fax:	+1 - 540 - 639 - 4162

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