AKD®

PROFINET RT Communication

Edition: Revision K, September 2016
Valid for firmware version 1.15
Part Number 903-200012-00
Original Documentation

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

Record of Document Revisions

Revision	Remarks
...	Table with lifecycle information of this document see "Record of Document Revisions" (➜ p. 67)
H, 12/2015	Added Signal numbers 104 to 107 to Telegram configuration (➜ p. 48). Added PN Parameters (➜ p. 59) chapter. Added PNU 28 to Supported PNU's (➜ p. 30). Added Standard telegram 400 (➜ p. 56).
J, 03/2016	Section Safety removed, chapter target group added, warning notes format updated
K, 09/2016	Updated instructions to Submode „Program mode“ (➜ p. 50).

Trademarks

- AKD is a registered trademark of Kollmorgen Corporation
- EnDat is a registered trademark of Dr. Johannes Heidenhain GmbH
- EtherCAT is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH
- Ethernet/IP is a registered trademark of ODVA, Inc.
- Ethernet/IP Communication Stack: copyright (c) 2009, Rockwell Automation
- sercos® is a registered trademark of sercos® international e.V.
- HIPERFACE is a registered trademark of Max Stegmann GmbH
- PROFINET is a registered trademark of PROFIBUS and PROFINET International (PI)
- SIMATIC is a registered trademark of SIEMENS AG
- Windows is a registered trademark of Microsoft Corporation

Current patents

- US Patent 5,162,798 (used in control card R/D)
- US Patent 5,646,496 (used in control card R/D and 1 Vp-p feedback interface)
- US Patent 6,118,241 (used in control card simple dynamic braking)
- US Patent 8,154,228 (Dynamic Braking For Electric Motors)
- US Patent 8,214,063 (Auto-tune of a Control System Based on Frequency Response)

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

Printed in the United States of America

1 Table of Contents

2 General

2.1 About this Manual

This manual, *AKD PROFINET RT Communication*, describes the installation, setup, range of functions, and software protocol for the PROFINET AKD product series. All AKD 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 your drive. Manual updates can be downloaded from the Kollmorgen website.

Related documents for the AKD series include:

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

Additional documentation:

- Profile-PROFIdrive (PI group, Profile-PROFIdrive_3172_v41_May06.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 Symbols Used

Symbol	Indication
DANGER	Indicates a hazardous situation which, if not avoided, will result in death or serious injury.
WARNING	Indicates a hazardous situation which, if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates situations which, if not avoided, could result in property damage.
NOTE	This symbol indicates important notes.
	Warning of a danger (general). The type of danger is specified by the text next to the symbol.
	Warning of danger from electricity and its effects.
	Warning of danger from suspended loads.
	Warning of danger from high temperature.
	Warning of danger from automatic start.

2.4 Abbreviations Used

Abbreviation	Meaning
Cat	Category
DO	Drive object
DU	Data Unit
GSD	Device description file
GSDML	GSD Markup Language
HMI	Human machine interface
ID	Identifier
I/O	Input / Output
IRT	Isochronous Real-Time
LED	Light emitting diode
PAP	Programm Ablauf Protokoll (program sequence protocol)
PLC	Programmable logic control
PNU	Parameter number
RT	Real-Time
STW	Control word
ZSW	Status word

3 Installation and Setup

3.1 Important Instructions

⚠DANGER

High Voltage up to 900V

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 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.

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:

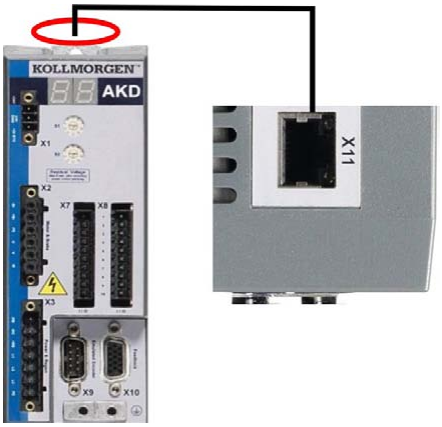
rotary	**linear**
at sinusoidal² commutation: 7500 rpm	at sinusoidal² commutation: 4 m/s
at trapezoidal commutation: 12000 rpm.	at trapezoidal commutation: 6.25 m/s

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 that were previously entered for position control and motion blocks must be adapted.

3.2 PROFINET Onboard

Connection to the PROFINET Network via X11.



Connect the service interface (X11) of the drive to an Ethernet interface on the PROFINET Master directly or via a network switch, **while the supply to the equipment is switched off.**

Confirm that the link LED on the AKD (the green LED on the RJ45 connector) and on your Master or Switch are both illuminated. If both lights are illuminated, then you have a good electrical connection.

PROFINET RT and WorkBench can operate simultaneously if a switch is used.

3.2.1 LED functions

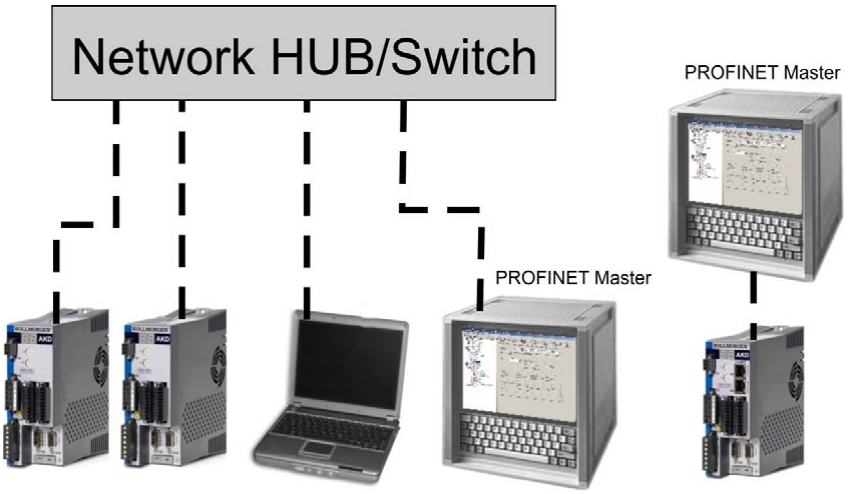
The communication status is indicated by the built-in LEDs.

Connector	LED#	Name	Function
X11	LED1	IN port Link	ON = active, OFF= not active
	LED2	RUN	ON = running, OFF = not running

3.2.2 Connection technology

You can connect to the PROFINET network using RJ-45 connectors. Use standard Cat. 5 Ethernet cables for either connection configuration.

3.2.3 Network Connection Examples



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.

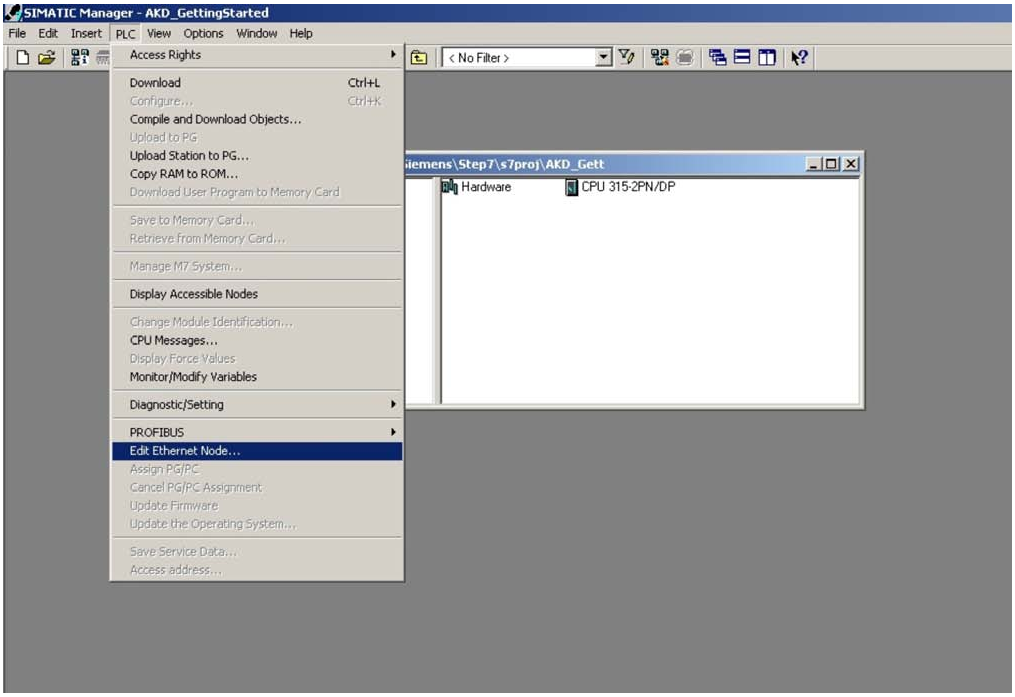
- Take appropriate measures to ensure that the operating and service personnel is aware of this danger.
- Implement appropriate protective measures to ensure that any unintended start-up of the machines cannot result in dangerous situations for personnel or machinery.
- Software limit-switches are not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

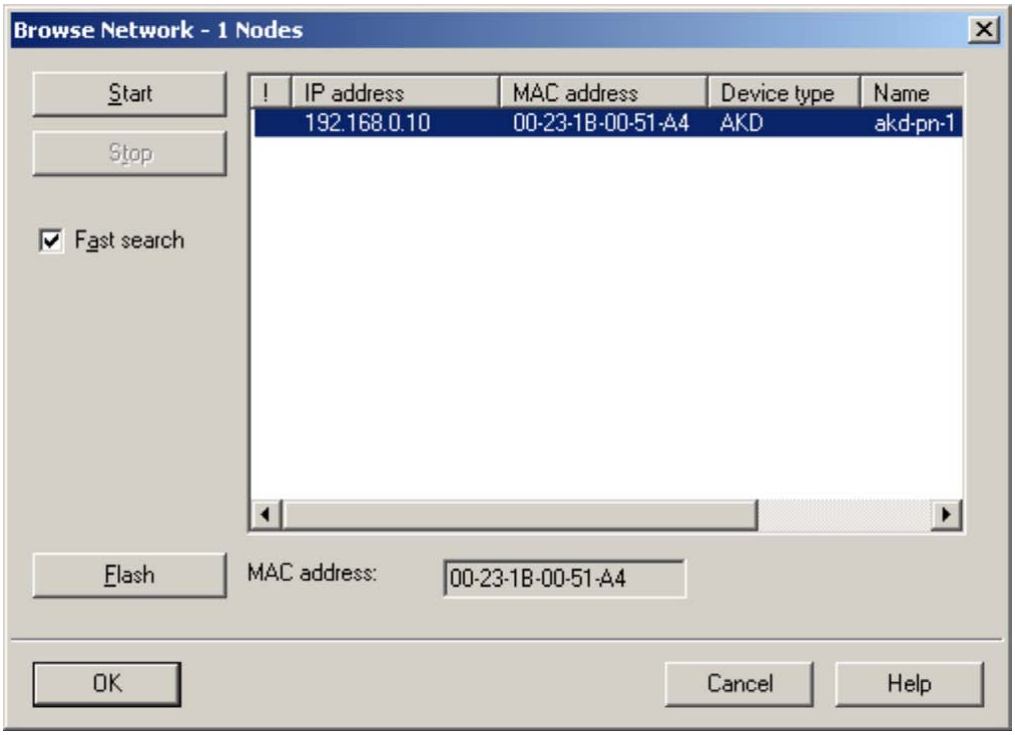
1. Check assembly/installation. Check that all the safety instructions in the product manual for the drive and this manual have been observed and implemented. Check station address and baud rate setting.
2. Connect PC,start WorkBench. Use the setup software WorkBench to set the parameters for the drive.
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 in the online help of the setup software.
4. Save parameters. When the parameters have been optimized, save them in the drive.

3.4 Configure IP Address parameters

Start the SIMATIC Manager. To assign a new IP address, go to PLC->Edit Ethernet Node:



In the next dialog window, click on browse in the Ethernet node group and look for all PROFINET devices in your network:



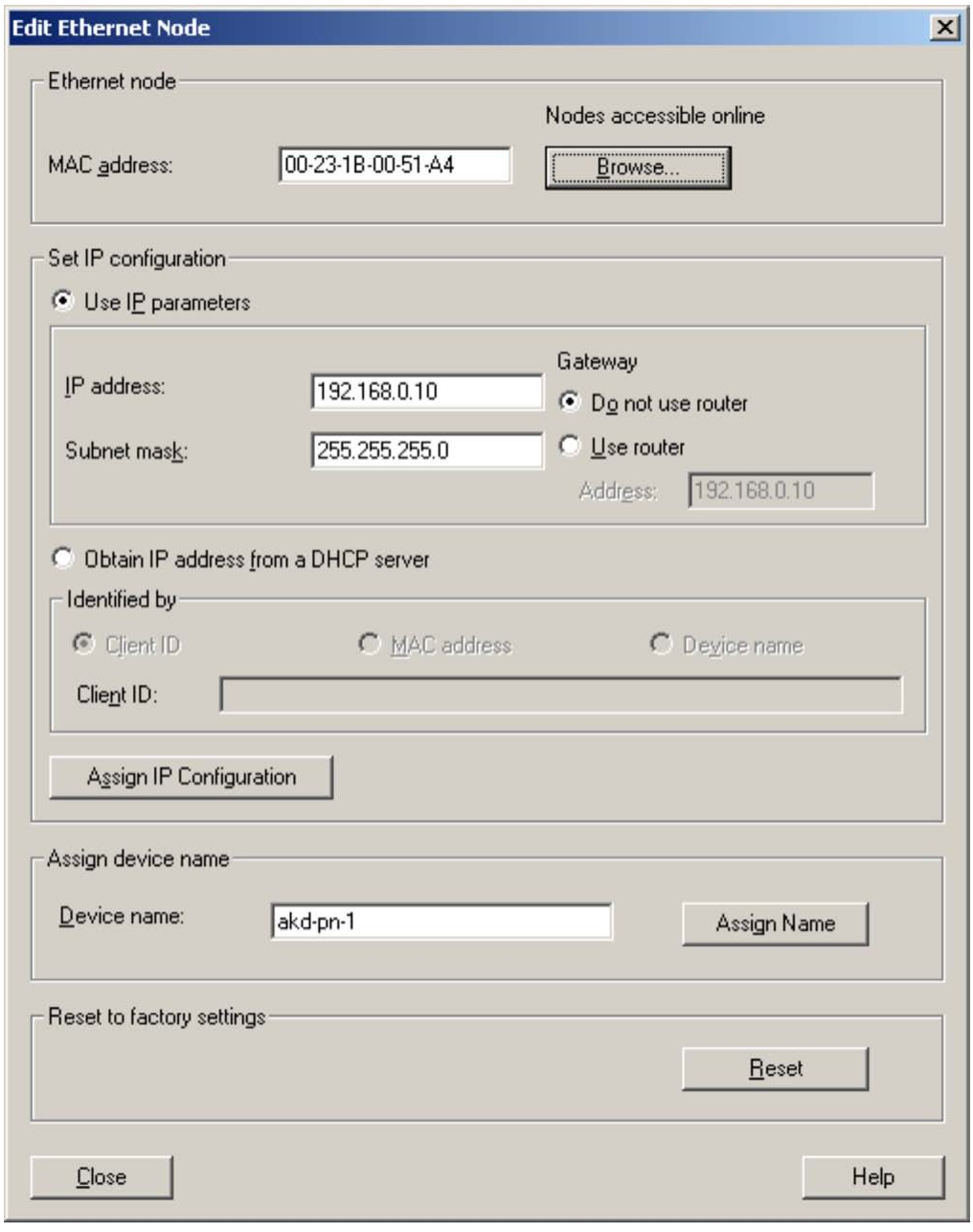
Select the AKD and click ok. If you have several AKD's in your network, you can also use the MAC address to filter one PROFINET device.

To be sure, that the intended device is selected, you can click the "Flash" Button in the dialog. The display of selected device will flash as long as this function is active.

NOTE The MAC address can be found on the label of the AKD.

Select the radio button *Use IP parameters*: then enter a new IP address and subnet mask to AKD. Click the *Assign IP Configuration* button for the change to take effect

Use the same popup to change the device name at this time by entering a name in the *Device Name* fields and clicking on the *Assign Name* button. Each device connected to the same IO connection must have a unique name. The PROFINET device name for AKD is derived from the AKD DRV.NAME parameter. The PLC, which acts as PROFINET IO-Controller. will use the *Device Name* as address and can change the IP address for each *Device Name*.



You will usually see a status message that indicates that the change was successful and the AKD display will show the new address. If you receive a failure message, make sure that no IO connection is currently running, then retry the address or name change.

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

3.4.1 Dependency Service channel (WorkBench) and PROFINET

WorkBench and PROFINET use the same IP communication channel to communicate with the drive so changing the IP address has implications for both interfaces.

There are several options for assigning the IP address for WorkBench and PROFINET as described in the AKD User Guide:

- DHCP, AutoIP
- Static IP addressing
  - Via rotary switches (Address area 192.168.0.xx)
  - Via ASCII commands IP.ADDRESS, IP.SUBNET. IP.GATEWAY
- PROFINET Devices (only): DCP

Once you have changed the IP address via DCP (e.g. through the steps above), the IP.* parameters stored in the drive are overwritten. Should you need to use DHCP or static IP addressing later, you must use WorkBench to set, the IP.MODE to something other than 1. See AKD User Guide for details on the mode you would like to use.

3.4.2 Reset of IP Address parameters

If the AKD can not be found in the network, as a last resort, reset all IP address parameters to their default values.

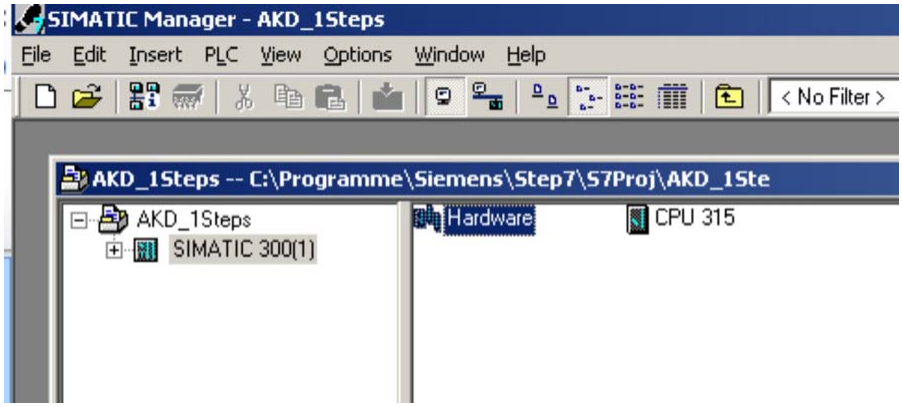
This is done by setting the rotary switches to 0 and holding the B1 button longer than 5 seconds. After this, the current IP address setting will be lost and the IP address is restored to default settings (after power cycle):

If rotary switches are zero, the DHCP and AutoIP is enabled.

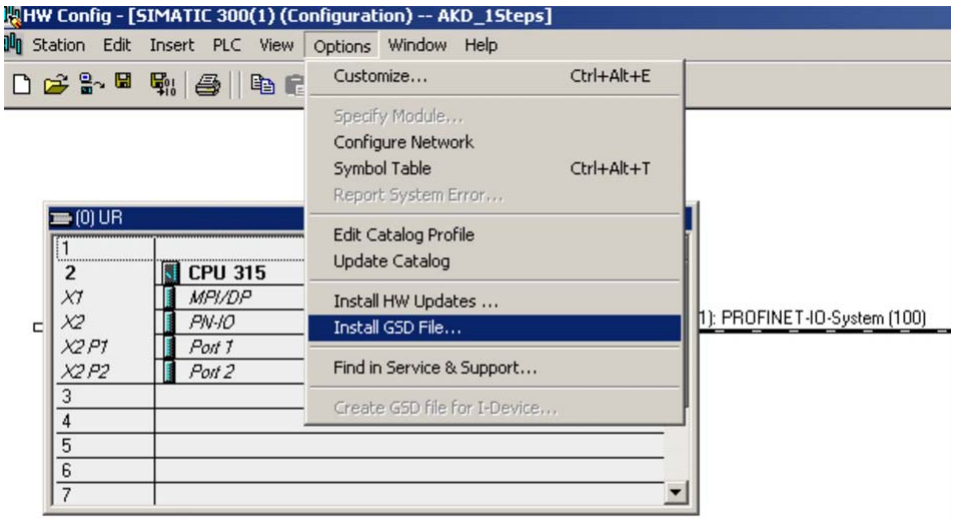
If rotary switches are not zero, the static IP address is 192.168.0.xx (xx rotary switches) and subnet mask 255.255.255.0.

3.5 Setup Step 7

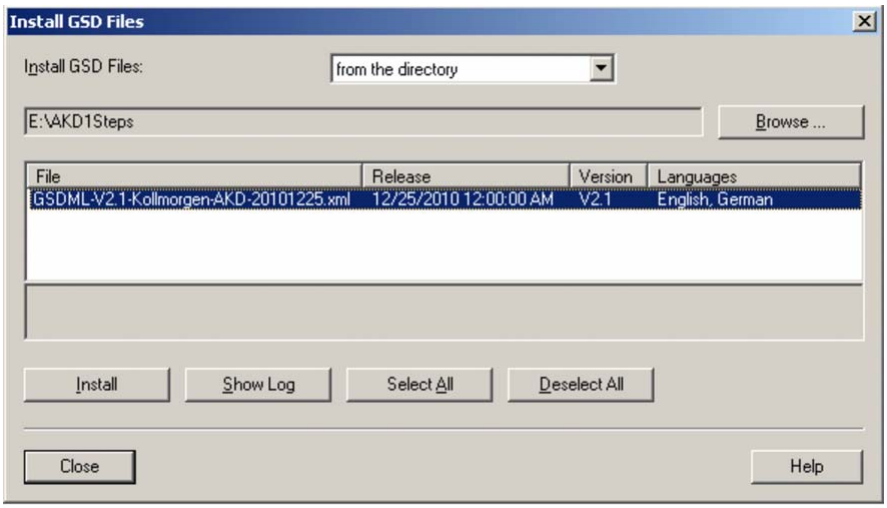
1. Start the SIMATIC Manager.
2. Open the hardware manager (double click on Hardware).



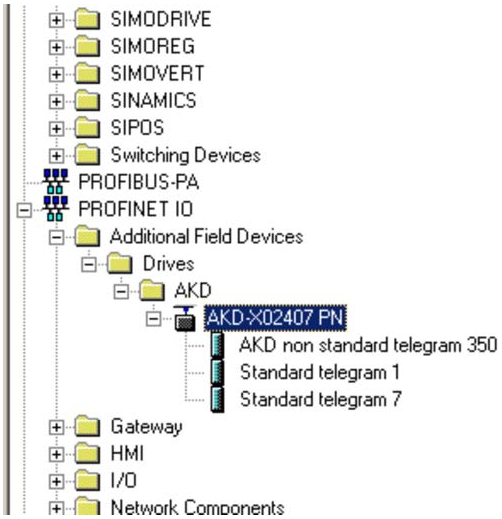
3. Go to Options and click "Install GSD Files". Here also the GSDML files for PROFINET devices can be installed.:



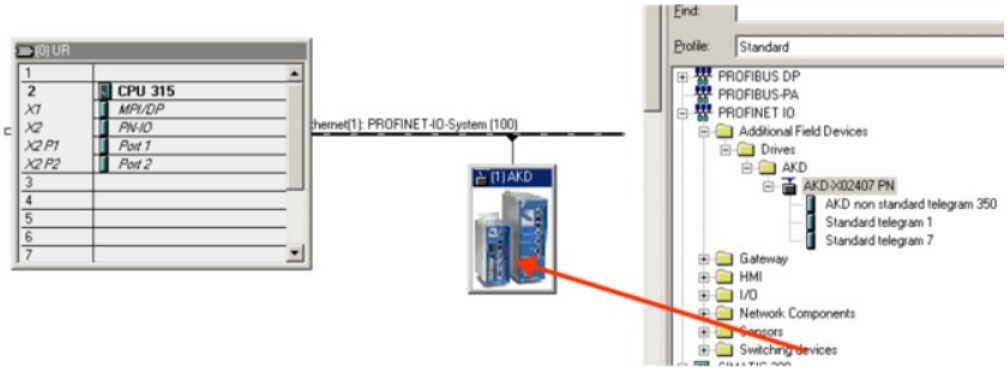
4. Browse for the latest AKD GSDML file and click on install:



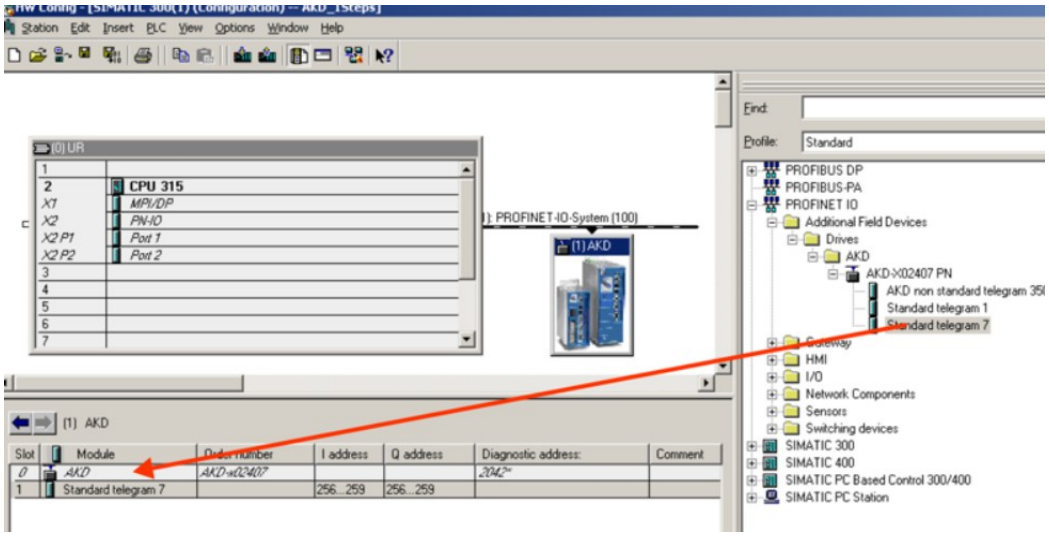
5. The AKD GSDML file is installed now and can be found in the SIMATIC hardware catalog.
   Open PROFINET I/O->Additional Fieldbus Devices->Drives->AKD



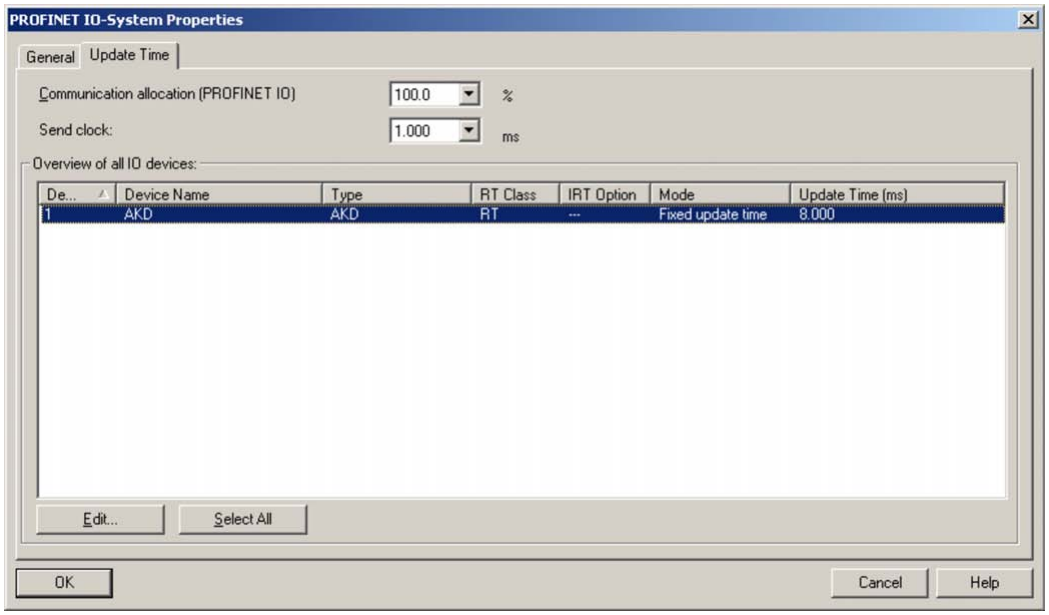
6. Click on the AKD device (not a telegram) and connect it to the PLC (drag and drop)



7. Now configure the telegram, for example telegram 7 for use in position mode.
   Drag and drop telegram 7 into slot 1.



8. Double click on the PROFINET network (line which connects PLC and AKD) and configure the update time. Click OK for closing this window.



9. Save and compile the hardware configuration.

3.6 Parameter Configuration with PROFIdrive over PROFINET IO

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

Within the AR, further addressing needs to done. PROFINET IO divides each device in so called slots and subslots. Sub 0 refers to the device itself and returns all generic data like vendor name, software and hardware version. The subslots within the device can be used with different real and virtual modules. Each module a functional component, which for example can be a digital I/O or Telegram with Position values.

AKD provides several virtual modules, which can be used in Slot 1 and are used for the real time data exchange.

For read or write parameters to or from the AKD, the global base mode parameter access can be used (see PROFIdrive chapter 8.6). The parameter manager is accessed through Slot 1 and a non real time channel needs to be used for this purpose. The AKD supports the record data 47, which is used to address the Parameter numbers (PNUs).
Base mode parameter access shows 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) (only for request „Change parameter“)	Format	No. of Values	4 + 6 x n
	Values		
	...		
nth Parameter Value(s)	...		
			4 + 6 x n +...+ (Format_n x Qty_n)

The following PROFIdrive services are supported:

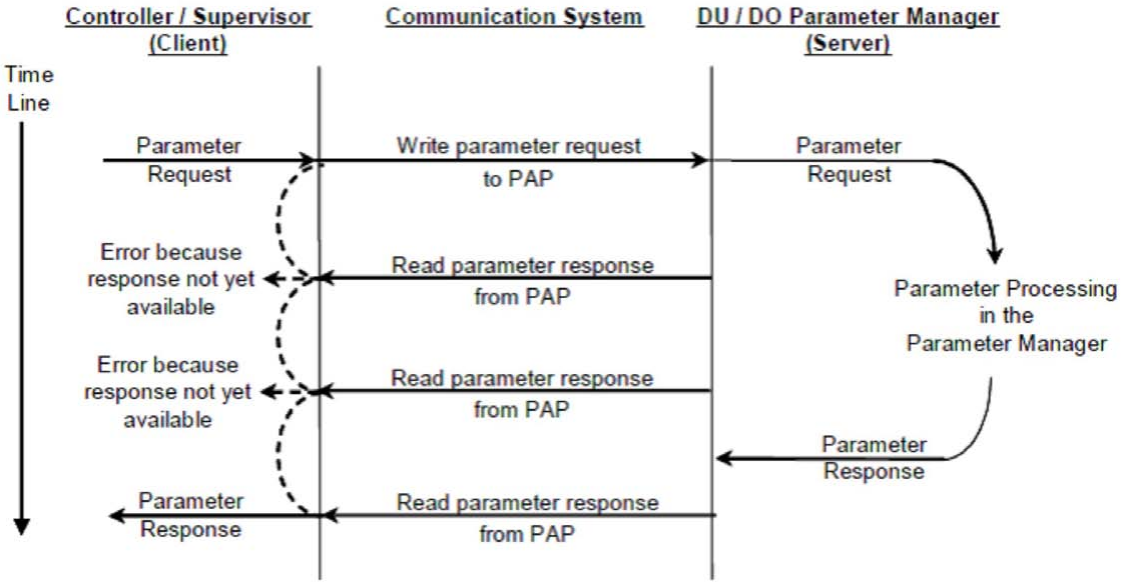
- Single parameter value request
- Multiple parameter value request
- Single parameter change request
- Multiple parameter change request

Record data fields

The table shows the structure and the supported fields in the AKD 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	0x00	one Axis
No. of Parameters	Unsigned8	0x01.. 0x27	
Attribute	Unsigned8	0x00 reserved 0x10 Value 0x20 Description	
No. of Elements	Unsigned8	0x01.. 0xEA Quantity	
Parameter number	Unsigned16	0x0001 .. 0xFFFF PNU	
Subindex	Unsigned16	0x0000 .. 0xFFFE	

3.6.1 Parameter configuration



3.6.2 Example for writing the operation mode

For writing the operation mode an acyclic change parameter value request needs to be send from the IO-Controller/Supervisor to the AKD.

If the user wants to write e.g. the operation mode to position mode (DRV.OPMODE 2) over PROFINET, the PNU 930 needs to be written with value 0x0002. The PROFIdrive base parameter access (see "Position Units" (➜ p. 57)) describes the procedure.

Change parameter request (Operation mode):

Byte (dec)	Value (hex)	Description
0	0x05	Request reference: e.g. 5
1	0x02	Request ID: Change parameters
2	0x00	Axis: 0 (the AKD parameter manager)
3	0x01	No of Parameter: 1
4	0x10	Attribute: Value
5	0x01	No. of Elements
6	0x03	PNU: 930 Operation mode
7	0xA2	
8	0x00	Subindex: 0
9	0x00	
10	0x42	Format: Word
11	0x01	No of Values: 1
12	0x00	Operation mode
13	0x02	

The AKD answers with a positive response without values:

Byte (dec)	Value (hex)	Description
0	0x05	Request Ref. mirrored: e.g. 5
1	0x02	Response ID: Change parameters
2	0x00	Axis: 0 (the AKD parameter manager)
3	0x01	No of Parameter: 1
4	0x00	Format
5	0x00	No. of Values 0

4 PROFINET IO

4.1 Introduction

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

A PROFINET IO network consists of following devices:

- IO controller: This is typically the PLC, which controls the whole application.
- IO device: a decentralized IO device (e.g. drive, encoder, sensor), which is controlled by the IO controller.
- IO supervisor: HMI (human machine interface) or PC for diagnostic purposes or commissioning.

The real time channel (RT) is used for IO data and alarm mechanism. In PROFINET IO RT (conformance class A and B), the RT data is transferred via 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 special hardware for IO Devices and switches are required.

All diagnostic and configuration data is transferred via the non real time channel (NRT). The well known UDP protocol is used for this purpose. Anyhow, no timing determinism can be guaranteed and typical the cycle times can be > 100ms.

4.2 Restrictions and requirements

4.2.1 Conformance Classes

AKD support Conformance Classes A and B. This means PROFIdrive parameters can be configured over the PROFINET network, fault can be delivered and cyclic data channel functions. However, the synchronization between axes can not take place since it is a part of Conformance Class C.

4.2.2 Cycle time of RT data

AKD fastest cycle time for the PROFINET cyclic data is 16 milliseconds.

4.2.3 Connector

PROFINET network connector in the AKD is the same RJ45 connector used for the service functions. This connector is numbered as X11 on the AKD's top panel.

4.2.4 Network topology

AKD can be connected as an I/O device on the PROFINET network in two manners:

1. As the last node in the network (since AKD has only one connector) in a line topology
2. As another node on the network in star topology (using a switch)

4.2.5 Modbus

Modbus is not supported on AKD Profinet drives.

5 PROFIDRIVE over PROFINET IO

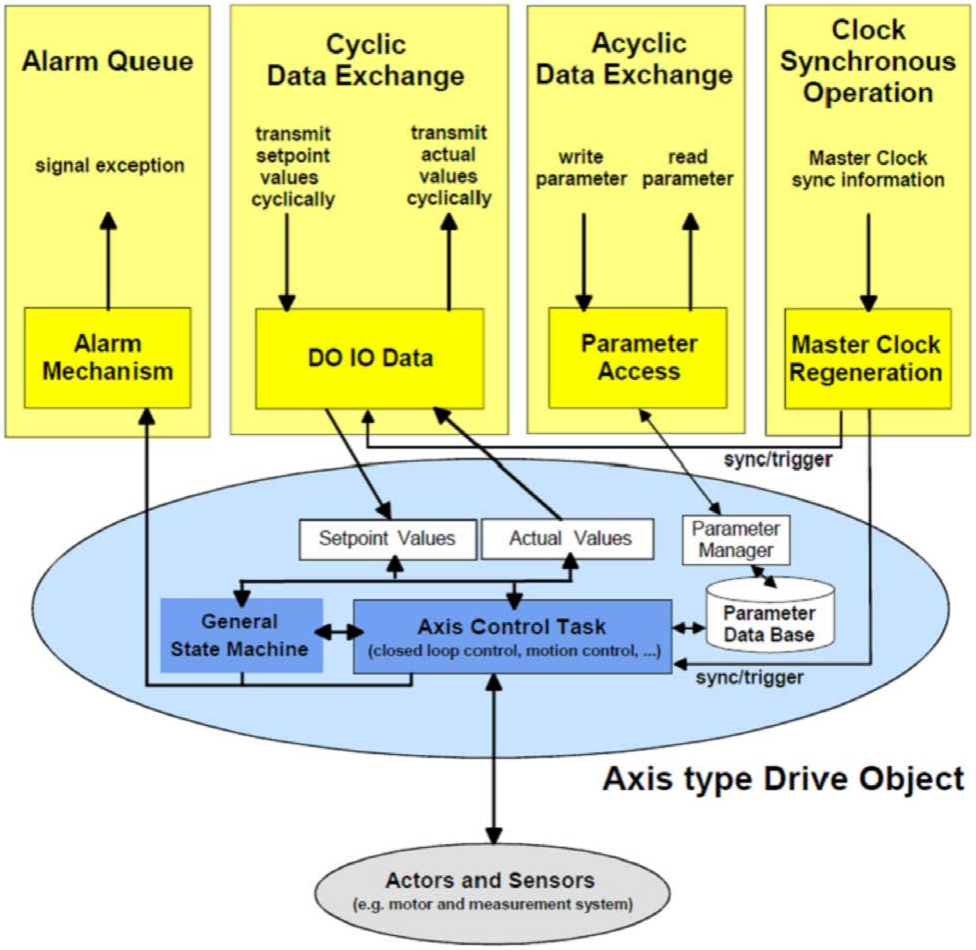
5.1 Introduction

The AKD supports the PROFIdrive profile for accessing and configuring standard and manufacture parameters via PROFINET IO to start/stop/configuring motion control tasks.

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

Note that the AKD supports all mandatory functionality of the PROFIdrive profile, but naturally not all optional functionality. This chapter describes the supported optional elements.

5.2 AKD as Drive Object (DO)



The drive object contains the following items:

- General state machine
- Axis control task
- Parameter manager with parameter data base

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

- Cyclic data exchange
- Acyclic data exchange
- Alarm queue (currently not supported)
- Clock synchronous operation (currently not supported)

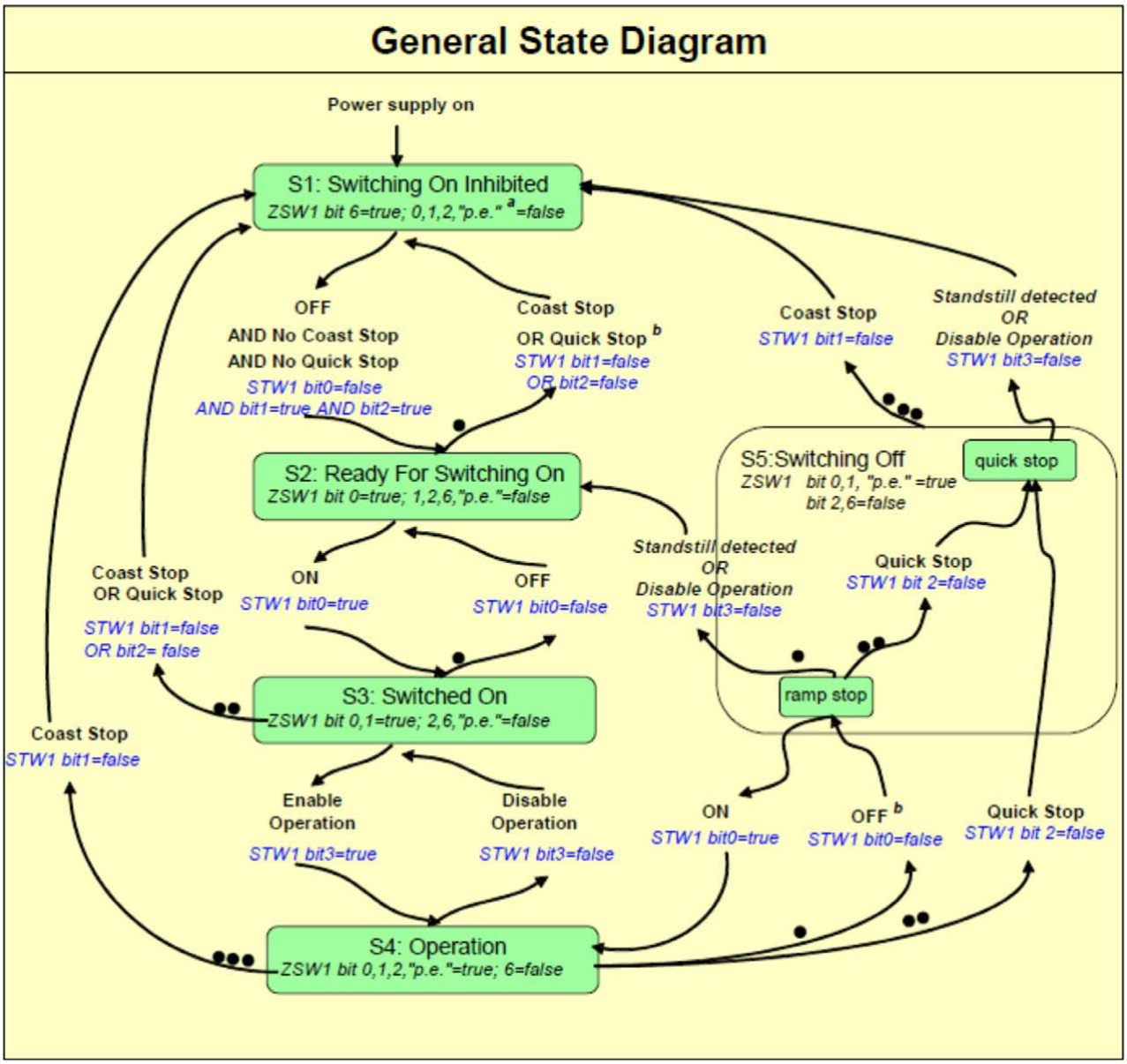
The cyclic data exchange includes the transmission/reception of data values like set point values (e.g. Position set point, velocity set point or control word) and actual values (actual position value, actual velocity or status word) between the master and the drive object. These values are called IO 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 an 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 an exception situations, which are generated through the state machine or the axis control task itself (not supported in AKD).

The clock synchronous operation requires PROFINET IRT (conformance class C), which is currently not supported by the AKD.

5.3 General State Machine



5.4 Control word bits (STW1)

The S7 application must set the bits in control word 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.

The control word (STW1) defines the following general functions:

General Control Word Bits		
Bit Number	**Description**	**Comment**
0	STW1 on/off	ON / OFF.
1	STW1 no coast stop	The drive will not coast stop if this bit is set.
2	STW1 no quick stop	The drive will not execute quick stop if this bit is set.
3	STW1 enable operation	The drive will enable and execute command if all preconditions are set.
7	Fault acknowledge	Set this bit to reset faults in the drive.
10	Control by PLC	When not set no command will be accepted from the PLC.

In velocity mode:

STW1 Special bits (Velocity mode)		
Bit Number	**Description**	**Comment**
4	Enable ramp generator of the drive	Use DRV.ACC and DRV.DEC.
5	Unfreeze the ramp generator in the drive	If frozen, the drive stays at current velocity without continuing to ramp up or down.
6	Enable set point	The drive accepts set point from the master. If this bit is not set, the velocity will be 0.
8	Jog 1 on/off	The drive runs up/brakes along the ramp to jogging setpoint 1/standstill. Prerequisite: Operation is enabled, drive is in standstill and STW1 bit 4, 5, 6 = 0.
9	Jog 2 on/off	The drive runs up/brakes along the ramp to jogging setpoint 2/standstill. Prerequisite: Operation is enabled, drive is in standstill and STW1 bit 4, 5, 6 = 0.
11-15	Device specific	Not implemented.

Control word 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 drive to be controlled by the PLC.

Bit 7 is used to acknowledge fault. The AKD will clear the fault and go automatically to S1 state after a fault is cleared.

The optional jog bits 8 and 9 can be used for the jogging functionility in velocity mode. PNU 1004 and 1005 define the jogging setpoints 1 and 2.

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 drive runs up/brakes along the ramp to jogging setpoint 1/standstill. Prerequisite: Operation is enabled, drive is in standstill and STW1 bit 4, 5, 6 = 0.
9	Jog 2 on/off	The drive runs up/brakes along the ramp to jogging setpoint 2/standstill. Prerequisite: Operation is enabled, drive is in standstill and STW1 bit 4, 5, 6 = 0.
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 functionility in position mode. PNU 1004 and 1005 define the jogging setpoints 1 and 2.

5.5 Status word bits (ZSW1)

All status word1 bits are implemented according to the PROFIdrive standard.

For application class 1 (speed control) and 3 (position mode) all mandatory bits are implemented.

The status word (ZSW1) defines the following functions:

General Status Word 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 set point) / Operation Disabled.
3	Fault present	A fault is present in the drive.
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	

In velocity mode:

ZSW1 Special bits (Velocity mode)		
Bit Number	**Description**	**Comment**
8	Velocity error within range	
10	Target velocity reached	
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 (PL.ERR and PL.ERRWTHRESH).
10	Target position reached	DRV.MOTIONSTAT Bit 11(Motion task target position has been reached).
11	Home position set	DRV.MOTIONSTAT Bit 1 & 2 (Homing finished).
12	Traversing Task acknowledgment	On positive edge, traversing task acknowledged or set point accepted.
13	Drive stopped	Axis is not moving
14	Motion task active	DRV.MOTIONSTAT Bit 0 (Motion task active/inactive)
15	Device specific	Not implemented

5.6 Supported PNU's

List of all supported PROFIdrive PNU's

The table mentions all supported PROFIdrive specific parameters. The access needs to be done via base mode parameter access described in "Parameter Configuration with PROFIdrive over PROFINET IO" (➜ p. 18).

PNU	Name	Data type	Description
915	DO IO Data configuring (set point telegram)	Array of U16	
916	DO IO Data configuring (actual value telegram)	Array of U16	
922	Telegram selection	U16	The PROFIdrive telegram used for the IO connection can be configured.
923	List of all parameters for signals	Array of U16	All supported signals and their corresponding PNU's.
930	Operating mode	U16	
944	Fault message counter	U16	
947	Fault number	Array of U16	All active faults.
964	Drive Unit Identification	Array of U16	Indices 0 – 4
965	Profile identification number		
975	DO identification		
980 to 989	Number list of defined parameter	Array of U16	
1002	No. of singleturn bits	U16	Scaling of singleturn part in signal MDI_TARPOS
1004	Jog v1	S16	Jogging set point 1
1005	Jog v2	S16	Sogging set point 2
1006	Jog Acc	U16	Jogging acceleration. Uses the acceleration ramp of velocity loop.
1007	Jog Dec	U16	Jogging deceleration. Uses the deceleration ramp of velocity loop.
1008	Acceleration scaling	U16	Scaling factor for acceleration and deceleration.

List of all manufacturer specific PNU's

The table shows all manufacture specific signals. All supported PROFIdrive and manufacture specific signals can be mapped into telegram 0 (dynamic telegram configuration).

PNU	Name	Data type	Description
1	STW1	U16	I/O control word
2	ZSW1	U16	I/O status word
5	NSOLL_A	S16	Velocity set point value
6	NIST_A	S16	Velocity actual value
28	XIST_A	U32	Actual feedback position
32	SATZANW	U16	Motion task selection
33	AKTSATZ	U16	Actual motion task running
52	ITIST_GLATT	U16	Active Current (torque)

Supported Formats:

Format	Data Type
0x41	Byte
0x42	Word
0x43	Dword

Supported AKD PNUs

Parameter	Index	Attributes
AIN.CUTOFF	2000	DWord,
AIN.DEADBAND	2001	Word,
AIN.ISCALE	2002	DWord,
AIN.OFFSET	2003	Word, Signed,
AIN.PSCALE	2004	DWord, Signed,
AIN.VALUE	2006	Word,
AIN.VSCALE_32	2007	DWord,
AIN.ZERO	2008	Command,
AOUT.ISCALE	2009	DWord,
AOUT.MODE	2010	Word,
AOUT.OFFSET	2011	Word, Signed,
AOUT.PSCALE	2012	DWord, Signed,
AOUT.VALUE	2014	DWord, Signed,
AOUT.VALUEU	2016	DWord, Signed,
AOUT.VSCALE_32	2018	DWord,
BODE.EXCITEGAP	2019	Byte,
BODE.FREQ	2020	DWord,
BODE.IAMP	2021	DWord, Signed,
BODE.INJECTPOINT	2022	Byte,
BODE.MODE	2023	Byte,
BODE.MODETIMER	2024	DWord,
BODE.PRBDEPTH	2025	Byte,
BODE.VAMP_32	2026	DWord, Signed,
CAP0.EDGE	2027	Byte,
CAP0.EN	2028	Byte,
CAP0.EVENT	2029	Byte,
CAP0.FILTER	2030	Byte,
CAP0.MODE	2031	Byte,
CAP0.PLFB	2032	DWord, Signed,
CAP0.PREEDGE	2034	Byte,
CAP0.PREFILTER	2035	Byte,
CAP0.PRESELECT	2036	Byte,
CAP0.STATE	2037	Byte,
CAP0.T	2038	DWord,
CAP0.TRIGGER	2039	Byte,
CAP1.EDGE	2040	Byte,
CAP1.EN	2041	Byte,
CAP1.EVENT	2042	Byte,
CAP1.FILTER	2043	Byte,

Parameter	Index	Attributes
CAP1.MODE	2044	Byte,
CAP1.PLFB	2045	DWord, Signed,
CAP1.PREEDGE	2047	Byte,
CAP1.PREFILTER	2048	Byte,
CAP1.PRESELECT	2049	Byte,
CAP1.STATE	2050	Byte,
CAP1.T	2051	DWord,
CAP1.TRIGGER	2052	Byte,
CS.DEC	2053	DWord,
CS.STATE	2055	Byte,
CS.TO	2056	DWord,
CS.VTHRESH_32	2057	DWord,
DIN.ROTARY	2058	Byte,
DIN1.INV	2060	Byte,
DIN1.MODE	2061	Word,
DIN1.PARAM	2062	DWord, Signed,
DIN1.STATE	2064	Byte,
DIN2.INV	2065	Byte,
DIN2.MODE	2066	Word,
DIN2.PARAM	2067	DWord, Signed,
DIN2.STATE	2069	Byte,
DIN3.INV	2070	Byte,
DIN3.MODE	2071	Word,
DIN3.PARAM	2072	DWord, Signed,
DIN3.STATE	2074	Byte,
DIN4.INV	2075	Byte,
DIN4.MODE	2076	Word,
DIN4.PARAM	2077	DWord, Signed,
DIN4.STATE	2079	Byte,
DIN5.INV	2080	Byte,
DIN5.MODE	2081	Word,
DIN5.PARAM	2082	DWord, Signed,
DIN5.STATE	2084	Byte,
DIN6.INV	2085	Byte,
DIN6.MODE	2086	Word,
DIN6.PARAM	2087	DWord, Signed,
DIN6.STATE	2089	Byte,
DIN7.INV	2090	Byte,
DIN7.MODE	2091	Word,
DIN7.PARAM	2092	DWord, Signed,
DIN7.STATE	2094	Byte,
DOUT.CTRL	2095	Byte,
DOUT.RELAYMODE	2096	Byte,
DOUT1.MODE	2098	Byte,
DOUT1.PARAM	2099	DWord, Signed,

Parameter	Index	Attributes
DOUT1.STATE	2101	Byte,
DOUT1.STATEU	2102	Byte,
DOUT2.MODE	2103	Byte,
DOUT2.PARAM	2104	DWord, Signed,
DOUT2.STATE	2106	Byte,
DOUT2.STATEU	2107	Byte,
DRV.ACC	2108	DWord,
DRV.ACTIVE	2110	Byte,
DRV.CLRFAULTHIST	2111	Command,
DRV.CLRFAULTS	2112	Command,
DRV.CMDSOURCE	2113	Byte,
DRV.DBILIMIT	2114	DWord,
DRV.DEC	2115	DWord,
DRV.DIR	2117	Byte,
DRV.DIS	2118	Command,
DRV.DISMODE	2119	Byte,
DRV.DISSOURCES	2120	Word,
DRV.DISTO	2121	DWord,
DRV.EMUEDIR	2122	Byte,
DRV.EMUEMODE	2123	Word,
DRV.EMUEMTURN	2124	DWord,
DRV.EMUERES	2125	DWord,
DRV.EMUEZOFFSET	2126	Word,
DRV.EN	2127	Command,
DRV.ENDEFAULT	2128	Byte,
DRV.HANDWHEEL	2129	DWord,
DRV.HWENMODE	2130	Byte,
DRV.ICONT	2131	DWord, Signed,
DRV.IPEAK	2132	DWord, Signed,
DRV.IZERO	2133	DWord,
DRV.MOTIONSTAT	2134	DWord,
DRV.OPMODE	2135	Byte,
DRV.RSTVAR	2136	Command,
DRV.STOP	2137	Command,
DRV.TYPE	2138	Byte,
DRV.ZERO	2139	Byte,
FB1.BISSBITS	2140	Byte,
FB1.ENCRES	2141	DWord,
FB1.IDENTIFIED	2142	Byte,
FB1.INITSIGNED	2143	Byte, Signed,
FB1.MECHPOS	2144	DWord,
FB1.OFFSET	2145	DWord, Signed,
FB1.ORIGIN	2147	DWord,
FB1.PFIND	2149	Byte,
FB1.PFINDCMDU	2150	DWord,

Parameter	Index	Attributes
FB1.POLES	2151	Word,
FB1.PSCALE	2152	Byte,
FB1.RESKTR	2153	Word,
FB1.RESREFPHASE	2154	DWord, Signed,
FB1.SELECT	2155	Byte, Signed,
FB1.TRACKINGCAL	2156	Byte,
FBUS.PARAM01	2157	DWord,
FBUS.PARAM02	2158	DWord,
FBUS.PARAM03	2159	DWord,
FBUS.PARAM04	2160	DWord,
FBUS.PARAM05	2161	DWord,
FBUS.PARAM06	2162	DWord,
FBUS.PARAM07	2163	DWord,
FBUS.PLLTHRESH	2177	Word,
FBUS.SAMPLEPERIOD	2178	Byte,
FBUS.SYNCACT	2179	DWord,
FBUS.SYNCDIST	2180	DWord,
FBUS.SYNCWND	2181	DWord,
FBUS.TYPE	2182	Byte,
GEAR.ACCMAX	2183	DWord,
GEAR.DECMAX	2185	DWord,
GEAR.IN	2187	Word,
GEAR.MODE	2188	Word,
GEAR.MOVE	2189	Command,
GEAR.OUT	2190	Word, Signed,
GEAR.VMAX_32	2191	DWord,
HOME.ACC	2192	DWord,
HOME.AUTOMOVE	2194	Byte,
HOME.DEC	2195	DWord,
HOME.DIR	2197	Word,
HOME.DIST	2198	DWord, Signed,
HOME.FEEDRATE	2200	Word,
HOME.IPEAK	2201	DWord, Signed,
HOME.MODE	2203	Word,
HOME.MOVE	2204	Command,
HOME.P	2205	DWord, Signed,
HOME.PERRTHRESH	2207	DWord, Signed,
HOME.SET	2209	Command,
HOME.V_32	2210	DWord,
HWLS.NEGSTATE	2211	Byte,
HWLS.POSSTATE	2212	Byte,
IL.BUSFF	2213	DWord, Signed,
IL.CMD	2214	DWord, Signed,
IL.CMDU	2215	DWord, Signed,
IL.FB	2216	DWord, Signed,

Parameter	Index	Attributes
IL.FF	2217	DWord,
IL.FOLDFTHRESH	2218	DWord,
IL.FOLDFTHRESHU	2219	DWord, Signed,
IL.FOLDWTHRESH	2220	DWord, Signed,
IL.FRICTION	2221	DWord,
IL.IFOLD	2222	DWord,
IL.IUFB	2223	DWord, Signed,
IL.IVFB	2224	DWord, Signed,
IL.KACCFF	2225	DWord, Signed,
IL.KBUSFF	2226	DWord,
IL.KP	2227	Word,
IL.KPDRATIO	2228	DWord,
IL.KVFF	2229	DWord, Signed,
IL.LIMITN	2230	DWord, Signed,
IL.LIMITP	2231	DWord, Signed,
IL.MFOLDD	2232	DWord,
IL.MFOLDR	2233	DWord,
IL.MFOLDT	2234	DWord,
IL.MIFOLD	2235	DWord,
IL.OFFSET	2236	DWord, Signed,
IL.VCMD	2237	Word, Signed,
IL.VUFB	2238	Word, Signed,
IL.VVFB	2239	Word, Signed,
MOTOR.AUTOSET	2240	Byte,
MOTOR.BRAKE	2241	Byte,
MOTOR.BRAKERLS	2242	Byte,
MOTOR.CTF0	2243	DWord,
MOTOR.ICONT	2244	DWord,
MOTOR.IDDATAVALID	2245	Byte,
MOTOR.INERTIA	2246	DWord,
MOTOR.IPEAK	2247	DWord,
MOTOR.KT	2248	DWord,
MOTOR.LQLL	2249	DWord,
MOTOR.PHASE	2250	Word,
MOTOR.PITCH	2251	DWord,
MOTOR.POLES	2252	Word,
MOTOR.R	2253	DWord,
MOTOR.RTYPE	2254	Byte,
MOTOR.TBRAKEAPP	2255	Word,
MOTOR.TBRAKERLS	2256	Word,
MOTOR.TEMP	2257	DWord,
MOTOR.TEMPFAULT	2258	DWord,
MOTOR.TEMPWARN	2259	DWord,
MOTOR.TYPE	2260	Byte,
MOTOR.VMAX	2261	Word,

Parameter	Index	Attributes
MOTOR.VOLTMAX	2262	Word,
MT.ACC	2263	DWord,
MT.CLEAR	2265	Word, Signed,
MT.CNTL	2266	DWord,
MT.CONTINUE	2267	Command,
MT.DEC	2268	DWord,
MT.EMERGMT	2270	Word, Signed,
MT.LOAD	2271	Command,
MT.MOVE	2272	Word,
MT.MTNEXT	2273	Byte,
MT.NUM	2274	Byte,
MT.P	2275	DWord, Signed,
MT.SET	2277	Command,
MT.TNEXT	2278	Word,
MT.TNUM	2279	Byte,
MT.TPOSWND	2280	DWord, Signed,
MT.TVELWND_32	2282	DWord,
MT.V_32	2283	DWord,
MT.VCMD_32	2284	DWord, Signed,
PL.CMD	2285	DWord,
PL.ERR	2287	DWord,
PL.ERRMODE	2289	Byte,
PL.ERRFTHRESH	2290	DWord,
PL.ERRWTHRESH	2292	DWord,
PL.FB	2294	DWord, Signed,
PL.FBSOURCE	2296	Byte,
PL.INTINMAX	2297	DWord,
PL.INTOUTMAX	2299	DWord,
PL.KI	2301	DWord,
PL.KP	2302	DWord,
PL.MODP1	2303	DWord, Signed,
PL.MODP2	2305	DWord, Signed,
PL.MODPDIR	2307	Byte,
PL.MODPEN	2308	Byte,
PLS.EN	2309	Word,
PLS.MODE	2310	Word,
PLS.P1	2311	DWord, Signed,
PLS.P2	2313	DWord, Signed,
PLS.P3	2315	DWord, Signed,
PLS.P4	2317	DWord, Signed,
PLS.P5	2319	DWord, Signed,
PLS.P6	2321	DWord, Signed,
PLS.P7	2323	DWord, Signed,
PLS.P8	2325	DWord, Signed,
PLS.RESET	2327	Word,

Parameter	Index	Attributes
PLS.STATE	2328	Word,
PLS.T1	2329	Word,
PLS.T2	2330	Word,
PLS.T3	2331	Word,
PLS.T4	2332	Word,
PLS.T5	2333	Word,
PLS.T6	2334	Word,
PLS.T7	2335	Word,
PLS.T8	2336	Word,
PLS.UNITS	2337	Byte,
PLS.WIDTH1	2338	DWord, Signed,
PLS.WIDTH2	2340	DWord, Signed,
PLS.WIDTH3	2342	DWord, Signed,
PLS.WIDTH4	2344	DWord, Signed,
PLS.WIDTH5	2346	DWord, Signed,
PLS.WIDTH6	2348	DWord, Signed,
PLS.WIDTH7	2350	DWord, Signed,
PLS.WIDTH8	2352	DWord, Signed,
REC.ACTIVE	2354	Byte,
REC.DONE	2355	Byte,
REC.GAP	2356	Word,
REC.NUMPOINTS	2357	Word,
REC.OFF	2358	Command,
REC.STOPTYPE	2359	Byte,
REC.TRIG	2360	Command,
REC.TRIGPOS	2361	Byte,
REC.TRIGSLOPE	2363	Byte,
REC.TRIGTYPE	2364	Byte,
REC.TRIGVAL	2365	DWord, Signed,
REGEN.POWER	2367	DWord,
REGEN.REXT	2369	Word,
REGEN.TEXT	2370	DWord,
REGEN.TYPE	2371	Byte, Signed,
REGEN.WATTEXT	2372	Word,
SM.I1	2373	DWord, Signed,
SM.I2	2374	DWord, Signed,
SM.MODE	2375	Word,
SM.MOVE	2376	Command,
SM.T1	2377	Word,
SM.T2	2378	Word,
SM.V1_32	2379	DWord, Signed,
SM.V2_32	2380	DWord, Signed,
STO.STATE	2381	Byte,
SWLS.EN	2382	Word,
SWLS.LIMIT0	2383	DWord, Signed,

Parameter	Index	Attributes
SWLS.LIMIT1	2385	DWord, Signed,
SWLS.STATE	2387	Word,
UNIT.ACCLINEAR	2388	Byte,
UNIT.ACCROTARY	2389	Byte,
UNIT.PIN	2390	DWord,
UNIT.PLINEAR	2391	Byte,
UNIT.POUT	2392	DWord,
UNIT.PROTARY	2393	Byte,
UNIT.VLINEAR	2394	Byte,
UNIT.VROTARY	2395	Byte,
VBUS.OVFTHRESH	2397	Word,
VBUS.OVWTHRESH	2398	Word,
VBUS.RMSLIMIT	2399	Byte,
VBUS.UVFTHRESH	2400	Word,
VBUS.UVMODE	2401	Byte,
VBUS.UVWTHRESH	2402	Word,
VBUS.VALUE	2403	DWord, Signed,
VL.ARPF1	2404	DWord,
VL.ARPF2	2405	DWord,
VL.ARPF3	2406	DWord,
VL.ARPF4	2407	DWord,
VL.ARPQ1	2408	DWord,
VL.ARPQ2	2409	DWord,
VL.ARPQ3	2410	DWord,
VL.ARPQ4	2411	DWord,
VL.ARTYPE1	2412	Byte,
VL.ARTYPE2	2413	Byte,
VL.ARTYPE3	2414	Byte,
VL.ARTYPE4	2415	Byte,
VL.ARZF1	2416	DWord,
VL.ARZF2	2417	DWord,
VL.ARZF3	2418	DWord,
VL.ARZF4	2419	DWord,
VL.ARZQ1	2420	DWord,
VL.ARZQ2	2421	DWord,
VL.ARZQ3	2422	DWord,
VL.ARZQ4	2423	DWord,
VL.BUSFF_32	2424	DWord, Signed,
VL.CMD_32	2425	DWord, Signed,
VL.CMDU_32	2426	DWord, Signed,
VL.ERR_32	2427	DWord, Signed,
VL.FB_32	2428	DWord, Signed,
VL.FBFILTER_32	2429	DWord, Signed,
VL.FBSOURCE	2430	Byte,
VL.FF_32	2431	DWord, Signed,

Parameter	Index	Attributes
VL.GENMODE	2432	Word,
VL.KBUSFF	2433	DWord,
VL.KI	2434	DWord,
VL.KO	2435	DWord,
VL.KP	2436	DWord,
VL.KVFF	2437	DWord,
VL.LIMITN_32	2438	DWord, Signed,
VL.LIMITP_32	2439	DWord,
VL.LMJR	2440	DWord,
VL.MODEL_32	2441	DWord, Signed,
VL.OBSBW	2442	DWord,
VL.OBSMODE	2443	DWord,
VL.THRESH_32	2444	DWord, Signed,
WS.ARM	2445	Command,
WS.DISTMAX	2446	DWord, Signed,
WS.DISTMIN	2448	DWord, Signed,
WS.IMAX	2450	DWord, Signed,
WS.MODE	2451	Byte,
WS.NUMLOOPS	2452	Byte,
WS.STATE	2453	Byte,
WS.T	2454	Word,
WS.TDELAY1	2455	Word,
WS.TDELAY2	2456	Word,
WS.TDELAY3	2457	Word,
WS.VTHRESH_32	2458	DWord, Signed,
DIN1.FILTER	2459	Word,
DIN2.FILTER	2460	Word,
DIN3.FILTER	2461	Word,
DIN4.FILTER	2462	Word,
DIN5.FILTER	2463	Word,
DIN6.FILTER	2464	Word,
DIN7.FILTER	2465	Word,
FB1.HALLSTATEU	2466	Byte,
FB1.HALLSTATEV	2467	Byte,
FB1.HALLSTATEW	2468	Byte,
DRV.NVSAVE	2469	Command,
MODBUS.DIO	2470	DWord,
MODBUS.DRV	2471	DWord,
MODBUS.DRVSTAT	2472	DWord,
MODBUS.HOME	2473	DWord,
MODBUS.MOTOR	2474	DWord,
MODBUS.MT	2475	Word,
MODBUS.SM	2476	DWord,
DRV.FAULT1	2477	Word,
DRV.FAULT2	2478	Word,

Parameter	Index	Attributes
DRV.FAULT3	2479	Word,
DRV.FAULT4	2480	Word,
DRV.FAULT5	2481	Word,
DRV.FAULT6	2482	Word,
DRV.FAULT7	2483	Word,
DRV.FAULT8	2484	Word,
DRV.FAULT9	2485	Word,
DRV.FAULT10	2486	Word,
MODBUS.PIN	2487	DWord,
MODBUS.POUT	2488	DWord,
MODBUS.PSCALE	2489	Word,
FB2.ENCRES	2492	DWord,
FB2.MODE	2493	Word,
FB2.SOURCE	2494	Word,
MOTOR.TBRAKETO	2495	DWord, Signed,
MODBUS.MSGLOG	2496	Byte,
USER.INT1	2497	DWord, Signed,
USER.INT2	2498	DWord, Signed,
USER.INT3	2499	DWord, Signed,
USER.INT4	2500	DWord, Signed,
USER.INT5	2501	DWord, Signed,
USER.INT6	2502	DWord, Signed,
USER.INT7	2503	DWord, Signed,
USER.INT8	2504	DWord, Signed,
USER.INT9	2505	DWord, Signed,
USER.INT10	2506	DWord, Signed,
USER.INT11	2507	DWord, Signed,
USER.INT12	2508	DWord, Signed,
USER.INT13	2509	DWord, Signed,
USER.INT14	2510	DWord, Signed,
USER.INT15	2511	DWord, Signed,
USER.INT16	2512	DWord, Signed,
USER.INT17	2513	DWord, Signed,
USER.INT18	2514	DWord, Signed,
USER.INT19	2515	DWord, Signed,
USER.INT20	2516	DWord, Signed,
USER.INT21	2517	DWord, Signed,
USER.INT22	2518	DWord, Signed,
USER.INT23	2519	DWord, Signed,
USER.INT24	2520	DWord, Signed,
DRV.NVCHECK_32	2521	DWord,
FB3.MODE	2522	Word,
FB3.P_32	2523	DWord, Signed,
MODBUS.SCALING	2524	Byte,
DRV.EMUEPULSEWIDTH	2525	DWord,

Parameter	Index	Attributes
DRV.EMUECHECKSPEED	2526	Byte,
DRV.HWENABLE	2527	Byte,
IL.MI2T	2592	DWord,
AIN.DEADBANDMODE	2593	Word,
AIN.MODE	2594	Byte,
DIO10.DIR	2595	Byte,
DIO10.INV	2596	Byte,
DIO11.DIR	2597	Byte,
DIO11.INV	2598	Byte,
DIO9.DIR	2599	Byte,
DIO9.INV	2600	Byte,
FAULT130.ACTION	2601	Byte,
FAULT131.ACTION	2602	Byte,
FAULT132.ACTION	2603	Byte,
FAULT134.ACTION	2604	Byte,
FAULT702.ACTION	2605	Byte,
IP.MODE	2606	Word,
LOAD.INERTIA	2607	DWord,
MOTOR.KE	2608	DWord,
VBUS.HALFVOLT	2609	Byte,
FB2.DIR	2610	Byte,
DRV.HANDWHEELSRC	2611	Byte,
DRV.HWENDELAY	2612	Byte,
IL.KPLOOKUPINDEX	2613	Word,
IL.KPLOOKUPVALUE	2614	DWord,
FAULT451.ACTION	2615	Byte,
MOTOR.BRAKEIMM	2616	Byte,
AIN2.CUTOFF	2617	DWord,
AIN2.DEADBAND	2618	Word,
AIN2.DEADBANDMODE	2619	Word,
AIN2.ISCALE	2620	DWord,
AIN2.MODE	2621	Byte,
AIN2.OFFSET	2622	Word, Signed,
AIN2.PSCALE	2623	DWord,
AIN2.VALUE	2625	Word,
AIN2.VSCALE	2626	DWord,
AIN2.ZERO	2629	Command,
AOUT.CUTOFF	2635	DWord,
AOUT2.CUTOFF	2636	DWord,
AOUT2.ISCALE	2637	DWord,
AOUT2.MODE	2638	Word,
AOUT2.OFFSET	2639	Word, Signed,
AOUT2.PSCALE	2640	DWord,
AOUT2.VALUE	2642	DWord, Signed,
AOUT2.VALUEU	2644	DWord, Signed,

Parameter	Index	Attributes
AOUT2.VSCALE	2646	DWord,
BODE.IFLIMIT	2648	DWord, Signed,
BODE.IFTHRESH	2649	DWord, Signed,
BODE.VFLIMIT	2650	DWord, Signed,
BODE.VFTHRESH	2651	DWord, Signed,
DIN10.STATE	2653	Byte,
DIN11.STATE	2654	Byte,
DIN21.FILTER	2655	Word,
DIN21.INV	2656	Byte,
DIN21.MODE	2657	Word,
DIN21.PARAM	2658	DWord, Signed,
DIN21.STATE	2660	Byte,
DIN22.FILTER	2661	Word,
DIN22.INV	2662	Byte,
DIN22.MODE	2663	Word,
DIN22.PARAM	2664	DWord, Signed,
DIN22.STATE	2666	Byte,
DIN23.FILTER	2667	Word,
DIN23.INV	2668	Byte,
DIN23.MODE	2669	Word,
DIN23.PARAM	2670	DWord, Signed,
DIN23.STATE	2672	Byte,
DIN24.FILTER	2673	Word,
DIN24.INV	2674	Byte,
DIN24.MODE	2675	Word,
DIN24.PARAM	2676	DWord, Signed,
DIN24.STATE	2678	Byte,
DIN25.FILTER	2679	Word,
DIN25.INV	2680	Byte,
DIN25.MODE	2681	Word,
DIN25.PARAM	2682	DWord, Signed,
DIN25.STATE	2684	Byte,
DIN26.FILTER	2685	Word,
DIN26.INV	2686	Byte,
DIN26.MODE	2687	Word,
DIN26.PARAM	2688	DWord, Signed,
DIN26.STATE	2690	Byte,
DIN27.FILTER	2691	Word,
DIN27.INV	2692	Byte,
DIN27.MODE	2693	Word,
DIN27.PARAM	2694	DWord, Signed,
DIN27.STATE	2696	Byte,
DIN28.FILTER	2697	Word,
DIN28.INV	2698	Byte,
DIN28.MODE	2699	Word,

Parameter	Index	Attributes
DIN28.PARAM	2700	DWord, Signed,
DIN28.STATE	2702	Byte,
DIN29.FILTER	2703	Word,
DIN29.INV	2704	Byte,
DIN29.MODE	2705	Word,
DIN29.PARAM	2706	DWord, Signed,
DIN29.STATE	2708	Byte,
DIN30.FILTER	2709	Word,
DIN30.INV	2710	Byte,
DIN30.MODE	2711	Word,
DIN30.PARAM	2712	DWord, Signed,
DIN30.STATE	2714	Byte,
DIN31.FILTER	2715	Word,
DIN31.INV	2716	Byte,
DIN31.MODE	2717	Word,
DIN31.PARAM	2718	DWord, Signed,
DIN31.STATE	2720	Byte,
DIN32.FILTER	2721	Word,
DIN32.INV	2722	Byte,
DIN32.MODE	2723	Word,
DIN32.PARAM	2724	DWord, Signed,
DIN32.STATE	2726	Byte,
DIN9.STATE	2727	Byte,
DOUT10.STATE	2728	Byte,
DOUT10.STATEU	2729	Byte,
DOUT11.STATE	2730	Byte,
DOUT11.STATEU	2731	Byte,
DOUT21.MODE	2732	Byte,
DOUT21.PARAM	2733	DWord, Signed,
DOUT21.STATE	2735	Byte,
DOUT21.STATEU	2736	Byte,
DOUT22.MODE	2737	Byte,
DOUT22.PARAM	2738	DWord, Signed,
DOUT22.STATE	2740	Byte,
DOUT22.STATEU	2741	Byte,
DOUT23.MODE	2742	Byte,
DOUT23.PARAM	2743	DWord, Signed,
DOUT23.STATE	2745	Byte,
DOUT23.STATEU	2746	Byte,
DOUT24.MODE	2747	Byte,
DOUT24.PARAM	2748	DWord, Signed,
DOUT24.STATE	2750	Byte,
DOUT24.STATEU	2751	Byte,
DOUT25.MODE	2752	Byte,
DOUT25.PARAM	2753	DWord, Signed,

Parameter	Index	Attributes
DOUT25.STATE	2755	Byte,
DOUT25.STATEU	2756	Byte,
DOUT26.MODE	2757	Byte,
DOUT26.PARAM	2758	DWord, Signed,
DOUT26.STATE	2760	Byte,
DOUT26.STATEU	2761	Byte,
DOUT27.MODE	2762	Byte,
DOUT27.PARAM	2763	DWord, Signed,
DOUT27.STATE	2765	Byte,
DOUT27.STATEU	2766	Byte,
DOUT28.MODE	2767	Byte,
DOUT28.PARAM	2768	DWord, Signed,
DOUT28.STATE	2770	Byte,
DOUT28.STATEU	2771	Byte,
DOUT29.MODE	2772	Byte,
DOUT29.PARAM	2773	DWord, Signed,
DOUT29.STATE	2775	Byte,
DOUT29.STATEU	2776	Byte,
DOUT30.MODE	2777	Byte,
DOUT30.PARAM	2778	DWord, Signed,
DOUT30.STATE	2780	Byte,
DOUT30.STATEU	2781	Byte,
DOUT9.STATE	2782	Byte,
DOUT9.STATEU	2783	Byte,
DRV.BLINKDISPLAY	2784	Command,
DRV.CLRCRASHDUMP	2785	Command,
DRV.NVLOAD	2788	Command,
DRV.SETUPREQBITS	2790	DWord,
DRV.WARNING1	2791	DWord,
DRV.WARNING2	2792	DWord,
DRV.WARNING3	2793	DWord,
FAULT139.ACTION	2797	Byte,
FB1.CALTHRESH	2802	DWord,
FB1.P	2805	DWord, Signed,
FB1.PDIR	2807	Byte,
FB1.PIN	2808	DWord,
FB1.POFFSET	2809	DWord, Signed,
FB1.POUT	2811	DWord,
FB1.PUNIT	2812	DWord,
FB1.USERBYTE	2813	Byte,
FB1.USERDWORD	2814	DWord,
FB1.USERWORD	2815	Word,
FB2.P	2816	DWord, Signed,
FB2.PIN	2818	DWord,
FB2.POFFSET	2819	DWord, Signed,

Parameter	Index	Attributes
FB2.POUT	2821	DWord,
FB2.PUNIT	2822	DWord,
FB3.P	2823	DWord, Signed,
FB3.PDIR	2825	Byte,
FB3.PIN	2826	DWord,
FB3.POFFSET	2827	DWord, Signed,
FB3.POUT	2829	DWord,
FB3.PUNIT	2830	DWord,
HOME.MAXDIST	2831	DWord, Signed,
IL.DIFOLD	2833	DWord,
IL.MI2TWTHRESH	2834	Byte,
IL.MIMODE	2835	Byte,
IP.RESET	2836	Command,
MOTOR.VOLTMIN	2837	Word,
MOTOR.VOLTRATED	2838	Word,
MOTOR.VRATED	2839	DWord, Signed,
SD.LOAD	2842	Command,
SD.SAVE	2843	Command,
SD.STATUS	2844	Byte,
VL.FBUNFILTERED	2845	DWord, Signed,
WS.DISARM	2847	Command,
WS.FREQ	2848	DWord,
WS.TDELAY4	2849	Word,
WS.CHECKT	2850	Word,
WS.CHECKV	2851	DWord, Signed,
AOUT.VSCALE	2858	DWord,
WS.TSTANDSTILL	2860	Word,
WS.TIRAMP	2861	Word,
FB1.EXTENDEDMULTITURN	2862	Byte,
MOTOR.IMTR	2864	Word,
IL.FBSOURCE	2865	Byte,
MOTOR.IMID	2866	DWord,
WS.CHECKMODE	2867	Byte,
REGEN.POWERFILTERED	2868	DWord,
FBUS.PROTECTION	2871	Byte,
FBUS.BLOCKING	2872	Byte,
FBUS.STATE	2873	Byte, Signed,
TEMP.CONTROL	2874	Word, Signed,
TEMP.POWER	2875	Word, Signed,
TEMP.POWER	2876	Word, Signed,
TEMP.POWER	2877	Word, Signed,
MODBUS.ERRORMODE	2878	Byte,
MODBUS.CLRERRORS	2879	Command,
IL.CMDACC	2880	DWord, Signed,
DRV.DOWNLOADALLOWED	2882	DWord,

Parameter	Index	Attributes
CAP0.FBSOURCE	2883	Byte,
CAP1.FBSOURCE	2884	Byte,
FB1.INITPSAVED	2885	DWord, Signed,
FB1.INITPWINDOW	2887	DWord,
FB1.INITPSTATUS	2889	Byte,
FB1.LASTIDENTIFIED	2890	Byte,
DRV.MOTIONDISSOURCES	2891	Word,
MOTOR.LDLL	2892	DWord,
MOTOR.LISAT	2893	DWord,
MOTOR.IDMAX	2894	DWord,
MOTOR.PHSADVK1	2895	DWord, Signed,
MOTOR.PHSADVK2	2896	DWord, Signed,
MOTOR.TEMPC	2900	Word,
VL.VFTHRESH	2901	DWord, Signed,
IL.PWMFREQ	2903	Word,
IL.DEADBAND	2904	Word,
DRV.POWERBOARDID	2905	Byte,
DRV.EMUESTEPCMD	2906	DWord, Signed,
DRV.EMUESTEPMODE	2907	Word,
CMP0.MODE	2910	Byte,
CMP0.SOURCE	2911	Word, Signed,
CMP1.SOURCE	2912	Word, Signed,
CMP1.MODE	2913	Byte,
CMP0.ARM	2914	Byte,
CMP1.ARM	2915	Byte,
CMP0.OUTMASK	2916	DWord,
CMP0.SETPOINT	2917	DWord, Signed,
CMP0.STATE	2919	Byte,
CMP0.WIDTH	2920	DWord,
CMP0.WIDTHTYPE	2922	Byte,
CMP1.OUTMASK	2923	DWord,
CMP1.SETPOINT	2924	DWord, Signed,
CMP1.STATE	2926	Byte,
CMP1.WIDTH	2927	DWord,
CMP1.WIDTHTYPE	2929	Byte,
CMP0.MODBOUND1	2930	DWord, Signed,
CMP0.MODBOUND2	2932	DWord, Signed,
CMP0.MODEN	2934	Byte,
CMP0.MODVALUE	2935	DWord, Signed,
CMP1.MODBOUND1	2937	DWord, Signed,
CMP1.MODBOUND2	2939	DWord, Signed,
CMP1.MODEN	2941	Byte,
CMP1.MODVALUE	2942	DWord, Signed,
FB3.DIR	2944	Byte,
CMP0.ADVANCE	2945	DWord, Signed,

Parameter	Index	Attributes
CMP1.ADVANCE	2946	DWord, Signed,
SFD.DIAGMODE	2947	Byte,
SFD.ADDR	2948	DWord,
SFD.WRITEENABLE	2950	Byte,
SFD.SECTORERASE	2951	Command,
DOUT9.MODE	2952	Byte,
DOUT9.PARAM	2953	DWord, Signed,
DOUT10.MODE	2955	Byte,
DOUT10.PARAM	2956	DWord, Signed,
DOUT11.MODE	2958	Byte,
DOUT11.PARAM	2959	DWord, Signed,
CMP0.SOURCEVALUE	2961	DWord, Signed,
CMP1.SOURCEVALUE	2963	DWord, Signed,
FAULT570.ACTION	2966	Byte,
DRV.FAULTDISPLAYMODE	2967	Byte,
FB1.MOTORPHASE	2968	Word,
FB1.MOTORPOLES	2969	Word,
FB2.MOTORPHASE	2970	Word,
FB2.MOTORPOLES	2971	Word,
FB3.MOTORPHASE	2972	Word,
FB3.MOTORPOLES	2973	Word,
PL.PDELAY	2981	DWord,
VL.FFDELAY	2982	DWord,
MOTOR.FIELDWEAKENING	2983	Byte,
SM.ACC	3009	DWord,
SM.DEC	3011	DWord,
DRV.REBOOT	3013	DWord,
PN.POSSCALE	3014	DWord,
AIN.UVFTHRESH	3015	Word, Signed,
AIN.UVWTHRESH	3016	Word, Signed,
AIN.OVFTHRESH	3017	Word, Signed,
AIN.OVWTHRESH	3018	Word, Signed,

5.7 Signals

MDI_MOD	
Bit	**Description**
0	0: Relative Positioning 1: Absolute Positioning
1 to 15	Reserved

5.8 Telegram configuration

The telegram configuration is made according to the PROFIdrive standard. The PROFIdrive parameters used in the configuration are: P922, P923, P915, P916 (see PROFIdrive profile, page 110). The following PROFIdrive signals are changing the corresponding AKD signals:

Signal No.	Signal name	Signal format	PROFIdrive signal name	AKD signal
1	Control word 1	WORD	STW1	Control word 1
2	Status word 1	WORD	ZSW1	Status word 1
5	Speed A	WORD	NSOLL_A	VL.CMD
6	Speed actual value	WORD	NIST_A	VL.FB
32	Traversing block selection	WORD	SATZANW	MT.MOVE
33	Actual traversing block	WORD	AKTSATZ	MT.PARAMS
34	MDI target position	DWORD	MDI_TARPOS	MT.P*
35	MDI velocity	DWORD	MDI_VELOCITY	MT.V*
36	MDI acceleration	WORD	MDI_ACC	MT.ACC*
37	MDI deceleration	WORD	MDI_DEC	MT.DEC*
38	MDI mode	WORD	MDI_MOD	MT.CNTL*
52	Active current (torque)	WORD	ITIST_GLATT	IL.FB
100	Homing distance	DWORD	HOME_DIST	HOME.DIST
101	Motion task feedrate	WORD	MT_FEEDRATE	MT.FEEDRATE
102	Current Loop	WORD	IL_CMD_ FIELDBUS	IL.CMD*
103	Homing Mode	WORD	HOME_MODE	HOME.MODE
104	Position Loop Error	DWORD	XIST_ERROR	PL.ERR*
105	Drive Motion Status	DWORD	DRIVE_MOTION_ STATUS	DRV.MOTIONSTAT
106	Drive Inputs	WORD	DRIVE_INPUTS	See Drive Inputs Signal (➜ p. 49)
107	Reserved	WORD	-	-

*Attention: The PROFIdrive signals are not mapped 1:1 to the AKD signals. The unit conversion for PROFIdrive needs to be used.

Either the predefined standard telegrams can be used for accessing the signals or a free mapping can be used with telegram 0.

The signals are also available in the PNU list. Each signal can be read/write with the same PNU number. For instance, signal “Speed actual value” is also available with PNU 6.

5.8.1 Drive Inputs Signal

The DRIVE_INPUTS signal can be used to map and combine several AKD commands into one signal.

Bit number	Description	AKD command
0	Negative software limit switch	SWLS.STATE Bit 0
1	Positive software limit switch	SWLS.STATE Bit 1
2	Negative hardware limit switch	HWLS.NEGSTATE
3	Positive hardware limit switch	HWLS.POSSTATE
4	Hardware disable	DRV.DISSOURCES Bit 2
5	Controlled stop input	DRV.DISSOURCES Bit 5
6	STO state	STO.STATE
7		
8 to 16	Reserved	-

5.9 Velocity Mode (Application class 1)

In this mode, the drive is controlled via a primary set point (speed set point). The speed control is completely in the drive controller.

The field bus is merely the transmission medium between the automation system and the drive controller. The Cyclic Data Exchange Communication Service is used.

Example

This example demonstrates enabling the drive and executing motion in velocity mode using standard telegram 1. This means that the master needs to send 32 bits (16 control word and 16 velocity command) and read back 32 bits (16 status word and 16 velocity feedback)

1. Send control word 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)
2. Send control word 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)
3. Send control word 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)
4. Send control word 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 (1 rps)

5.10 Position Mode (Application class 3)

In this application class the Drive Object (DO) provides a closed position control loop with its own position interpolation. The motion tasks, which are configured by MT parameters in AKD, can be accessed.

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

Furthermore the general state machine of the drive Axis Object is extended to start/-configure/stop a motion task.

“ONLY” in state S4 („Operational“), the extended state machine can be accessed.

5.10.1 Submode „Program mode“

The „Program mode“ can be used to start/switch to a specific predefined motion task via I/O messaging. Telegram 7 ("Standard telegram 7" (➜ p. 54)) is used for this purpose. For addressing the motion task signal „SATZANW“ is used. With signal „AKTSATZ“ the actual running motion task number can be read.

Requirements:

- Drive axis state machine needs to be in S4 („Operational“)
- Operation mode needs to be „Position mode“
- Standard telegram 7 needs to be configured
- Axis needs to be homed (ZSW1 Bit 11 set, See also Status word bits (ZSW1) (➜ p. 29))
- Motion task needs to be configured

Start a motion task:

- Set SATZANW to the motion task number, which shall 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 will be activated
- ZSW1 Bit 13 will be change 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:

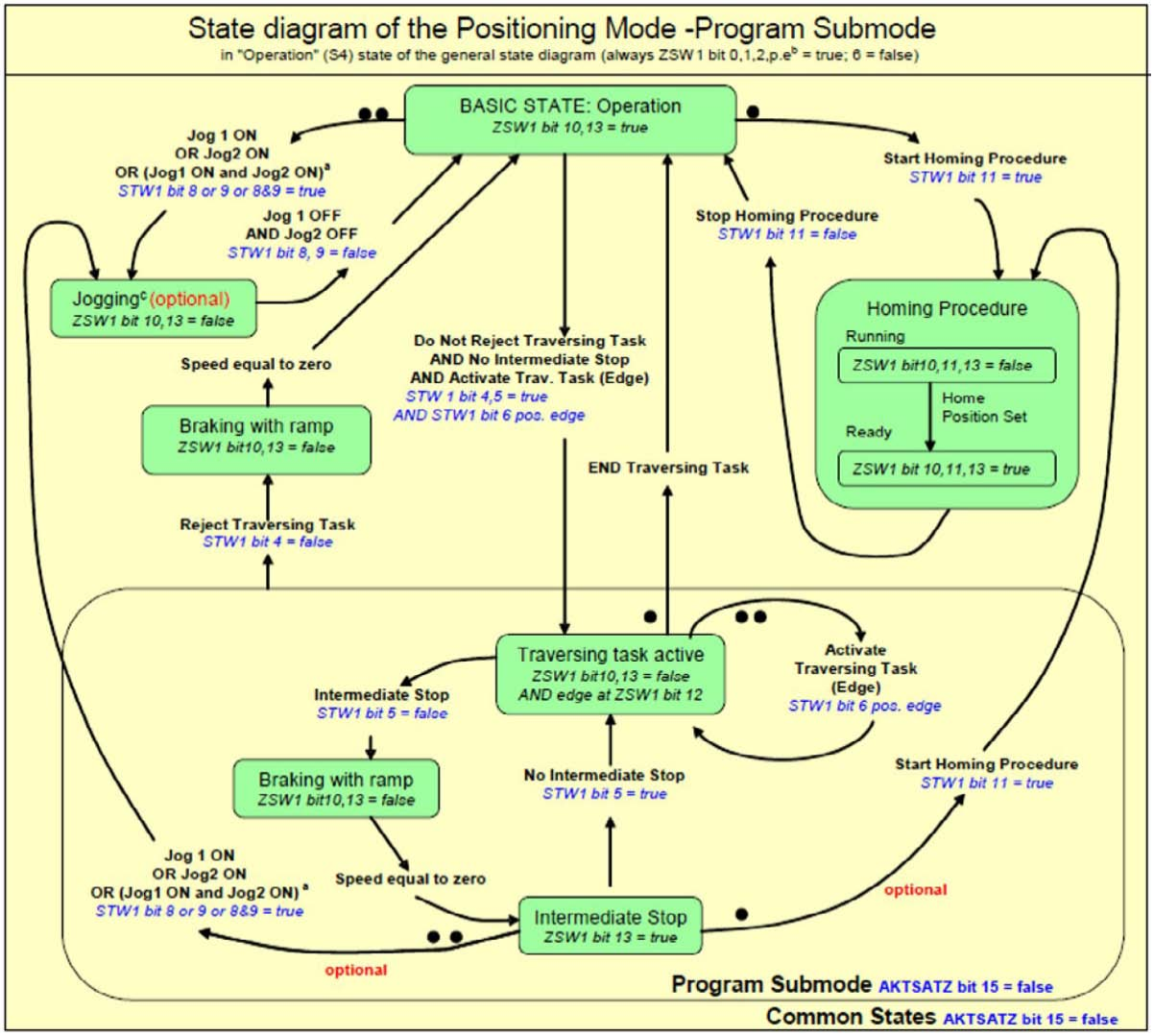
- case of warning, ZSW1 Bit 7 is set (See also Status word bits (ZSW1) (➜ p. 29))
- case of fault, ZSW1 Bit 3 is set (See also Status word bits (ZSW1) (➜ p. 29))

The figure on the next page shows the extension of the general state diagram of DO. Also the optional jog functionality is supported. The homing procedure can be achieved through bit STW1 Bit 11 (See also Homing (➜ p. 53)). After an intermediate stop, the motion task can be activated again through STW1 Bit 5 set.

If the general state machine of the DO is in “Operational” and the standard telegram 7 is used to configure a motion task, the following sequence can be used to start a motion task:

- Configure a motion task
- Change the general state machine to S4 (Drive is enabled)
- Set SATZANW to the motion task number, which needs to be started
- Used STW1 Bit 4,5 and 6 to start the motiontask. BIT 6 needs to be an edge

The extension of the general state diagram of DO:



5.10.2 Submode „Manual data input (MDI)“

The "manual data input" mode can be used to run a motion task directly configured through IO 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 can be activated by setting bit 15 in signal “SATZANW”.

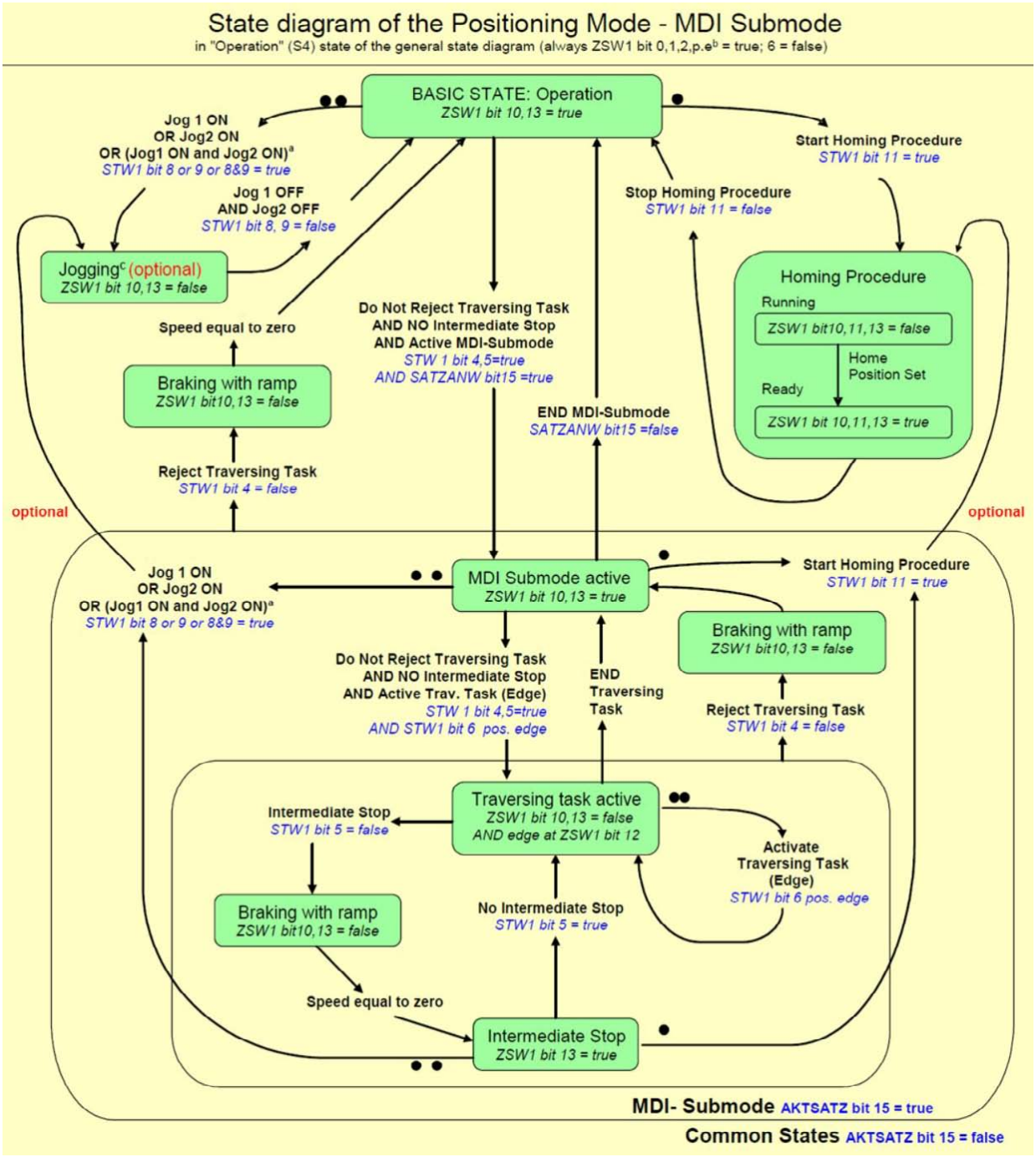
Requirements:

- Drive axis state machine needs to be in S4 („Operational“).
- Operation mode needs to be „Position mode“.
- Standard telegram 9 needs to be configured.
- Axis needs to be homed (ZSW1 Bit 11 set, See also Status word bits (ZSW1) (➜ p. 29).

Run a motion task:

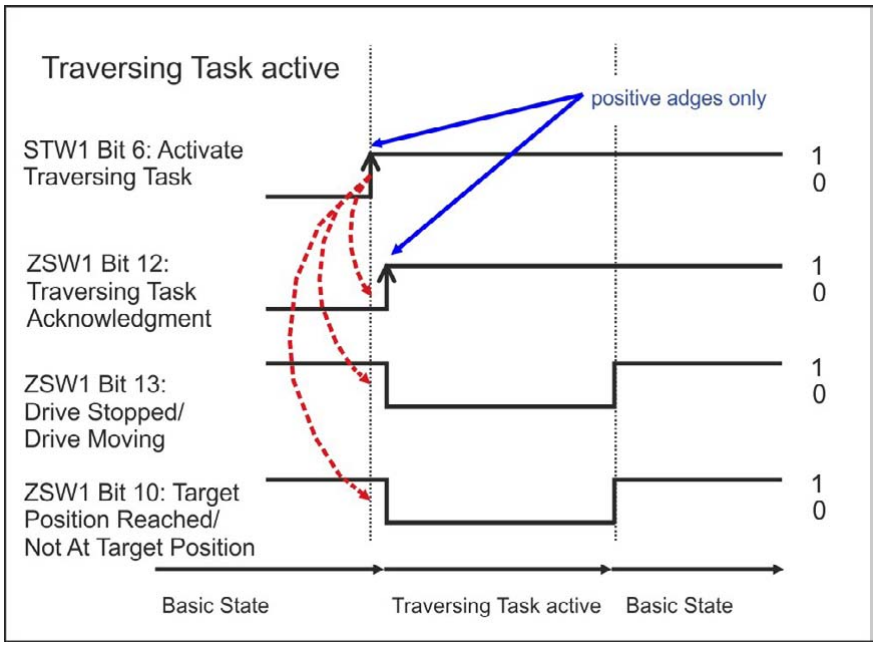
- Set bit 15 in SATZANW to 1
- Configure all setpoint value in telegram 9 like MDI_ACC, MDI_DEC, MDI_MOD etc..
- Set STW1 Bits 4 and 5 to true (Do not reject traversing task and no intermediate stop).
- Set STW1 Bit 6 from zero to one, the motion task will be loaded.
- ZSW1 Bit 12 is set to one, when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

The extension of the state diagram for mdi mode:



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 shows the behavior:



If another motion task shall be executed before the current motion task has finished ("Change on the fly"), the "Activate traversing task" Bit in STW1 can be directly cleared (after the setting of Bit 12 in ZSW1 was detected). In addition, the drive will clear Bit 12 in ZSW1. After this another motion task can be loaded. If an error occurs or the configured motion task can not be executed, bit 12 in ZSW1 will not be set.

5.10.3 Homing

Requirements:

- Drive axis state machine needs to be in S4 („Operational")
- The homing mode needs to be configured via HOME.MODE 1001 (available also via PNU 1001).
- No motion task is active
- Operation mode needs to be „Position mode"

Home procedure:

- STW1 Bit 11 set to one
- ZSW1 Bit 10, 11 ,13 will be set to false if homing is running
- ZSW1 Bit 10,11,13 will be 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, 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 AKD specific commands:

An activation of the homing procedure via STW1 Bit 11 corresponds to the AKD specific command HOME.MOVE. When the homing procedure is finished, the AKD set the bits 2 and bit 4 in DRV.MOTIONSTAT. „Only" if these two bits are set, the PROFIdrive specific homing flag ZSW1 Bit 11 (home position) is set.

5.11 I/O Telegrams

5.11.1 Telegram 0

Telegram 0 is used for the free mapping of PROFIdrive signals into the PROFINET I/O data. With PNU 922 the telegram can be configured. PNU915 defines then set point signals and PNU 916 the actual value signals.
Limitations: The number and kind of signals, which can be mapped, are depending on the configuration of your PROFINET master. The length for input and output values in the IO communication is given through telegram configuration in slot 1.

5.11.2 Standard telegram 1

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

IO Data Number	Set point	Actual values
1	STW1	ZSW1
2	NSOLL_A	NIST_A

5.11.3 Standard telegram 7

Typically used for application class 3 (position mode). Predefined motion tasks can be selected directly via IO data.

IO Data Number	Set point	Actual values
1	STW1	ZSW1
2	SATZANW	AKTSATZ

5.11.4 Standard telegram 9

Typically used for application class 3 (position mode). A motion task can be directly configured via IO data.

IO Data Number	Set point	Actual values
1	STW1	ZSW1
2	SATZANW	AKTSATZ
3	STW2	ZSW2
4	MDI_TARPOS	XIST_A
5		XIST_A
6	MDI_VELOCITY	
7		
8	MDI_ACC	
9	MDI_DEC	
10	MDI_MOD	

5.11.5 Manufacturer specific telegram 350

Telegram 350 is typically used for application class 1 (velocity mode). Additonally to telegram 1 the actual current value can be monitored in the IO data.

IO Data number	Set point	Actual values
1	STW1	ZSW1
2	NSOLL_A	NIST_A
3		ITIST_GLATT

5.11.6 Manufacturer specific telegram 351

Telegram 351 is typically used for application class 1 (velocity mode). Additonally to telegram 1 the actual current value and position value can be monitored in the IO data.

IO Data number	Set point	Actual values
1	STW1	ZSW1
2	NSOLL_A	NIST_A
3		ITIST_GLATT
4, 5		XIST_A

5.11.7 Manufacturer specific telegram 352

IO Data Number	Set point	Actual values
1	STW1	ZSW1
2	SATZANW	AKTSATZ
3	STW2	ZSW2
4	MDI_TARPOS	XIST_A
5		XIST_A
6	MDI_VELOCITY	
7		
8	MDI_ACC	
9	MDI_DEC	
10	MDI_MOD	
11	HOME_DIST	
12		

5.11.8 Manufacturer specific telegram 353

IO Data Number	Set point	Actual values
1	STW1	ZSW1
2	SATZANW	AKTSATZ
3	STW2	ZSW2
4	MDI_TARPOS	XIST_A
5		XIST_A
6	MDI_VELOCITY	
7		
8	MDI_ACC	
9	MDI_DEC	
10	MDI_MOD	
11	HOME_DIST	
12		
13	HOME_MODE	

5.11.9 Standard telegram 400

This telegram can be used for the free mapping. 24 Byte are available for the input and output of the I/O data. PNU 915 and 916 can be used to change the mapping. The default “mapping” is similar to telegram 9.

IO Data Number	Set point	Actual values
1	Process data word 1	Process data word 1
2	Process data word 2	Process data word 2
3	...	...
12	Process data word 12	Process data word 12

5.12 Units

5.12.1 Velocity units

Velocity units are normalized according to X2 data normalization of PROFIdrive (hence "set normalization bit x" => 2^x = 100% velocity).

In velocity mode:
The AKD uses x=15 and 100% is the maximum velocity of the AKD hence 12000 rpm. Thus the velocity units are 2^15 = 12000 rpm.

E.g. if the S7 wants to set a velocity of 60 rpm in the cyclic channel in needs to convert:

(60 / 12000) * 2^15 = 163

In position mode:
For signal MDI_VELOCITY x = 32 and 100% is the maximum velocity of the AKD hence 12000 rpm. Thus a value of 2^32 for MDI_VELOCITY is equal 12000 rpm.

5.12.2 Position Units

The signal MDI_TARPOS a 32 signed position value. In the default configuration, the resolution per revolution is 2^16 (65536) counts. The number of singleturn bits (default 16) can be changed through PNU 1002.

5.12.3 Acc-/Deceleration Units

The acceleration signal MDI_ACC and MDI_DEC are normalized in the X2 format (x = 16 is equal to 100% and means 50,000,000 rpm /sec). Acceleration and deceleration can be scaled using PNU1008 as follows

PNU1008 value	rpm/sec
1	763 rpm/sec
2	1536 rpm/sec
...	...
65536	50,000,000 rpm /sec

5.12.4 Current units

Current units are normalized according to N2 data normalization of PROFIdrive (2^14 = 100%). The AKD's 100% is the maximum current of the AKD hence DRV.IPEAK.

Thus current units are 2^14 = DRV.IPEAK.

Example: for 3 A AKD, if the S7 reads a current value of -182 Arms in the cyclic channel from the AKD, it will need to execute the conversion
(9 / 2^14) * (-182) = -0.1 Arms

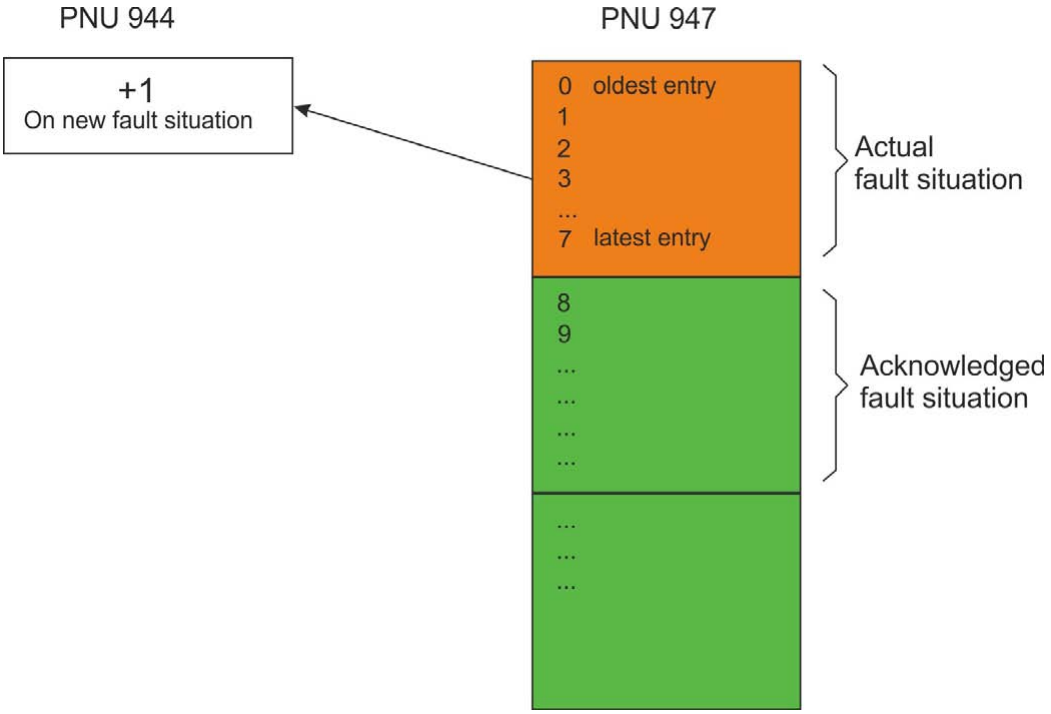
5.13 Alarms

Not implemented yet.

5.14 Fault

All actual faults can be described as a so called “Fault situation”. This fault situation can be read from the fault buffer with PNU 947. A fault situation can have up to 8 different faults. An unacknowledged is indicated by ZSW1 bit 3 (fault present). With acknowledging the actual fault situation, the fault numbers are shifted to the acknowledged faults section (shift of 8 positions in the fault buffer (PNU 947)). Each time a fault situation is generated, a fault message counter is incremented (PNU 944). With reading the value of the fault message counter before and after the read process of the Fault buffer (947), the user can be sure that no other fault situation has been occurred in the meanwhile if these values are equal.

The fault buffer PNU 947 store up to 8 fault situation. Each fault situation can have up to 8 fault numbers.



5.15 ASCII configuration

Use the following AKD parameters to configure behavior.

FBUS.PARAM01: Reads the configured telegram (PNU922 in Supported PNU's (➜ p. 30)).

FBUS.PARAM02 Bit 0:

- 1: Fault 702 (Fieldbus communication lost) is triggered if the PROFINET master changes states from RUN to STOP.
- 0: No fault is triggered if the PROFINET master changes states from RUN to STOP.

PN.STW1: Returns the control word (signal STW1) written by the PROFINET master.

PN.ZSW1: Returns the status word (signal ZSW1) sent by the AKD to the PROFINET master.

6 PN Parameters

This section describes thePN (Profinet) parameters.

6.1 PN.POSSCALE

General Information	
Type	NV Parameter
Description	Sets scaling for PNUs in Profinet which have an index greater than 2000, have the data type float, and are dependent on the position scaling parameters UNIT.PLINEAR or UNIT.PROTARY.
Units	N/A
Range	1 to 1,000,000
Default Value	1000
Data Type	Integer
Start Version	M_01-13-08-000

Variants Supported

Variant	Supported
AKD Base	No
AKD with Position Indexer	No
AKD EtherCAT	No
AKD CANopen	No
AKD BASIC	No
AKD SynqNet	No
AKD EtherNet/IP	No
AKD Profinet	Yes
AKD sercos® III	No
AKD-N	No
AKD-C	No

Fieldbus Information

Fieldbus	Address	Attributes	Signed?
Profinet	3014	DWord	No

Description

This parameter sets scaling for PNUs in Profinet which have an index greater than 2000, have the data type float, and are dependent on the position scaling parameters UNIT.PLINEAR or UNIT.PROTARY.

PNU values are scaled by dividing the parameter value by the value of PN.POSSCALE:

PNU value = parameter value (such as PL.FB or HOME.P) / PN.POSSCALE

6.2 PN.TIMEOUTFTHRESH

General Information	
Type	R/O Parameter
Description	Reads the Timeout Fault Threshold for the Profinet Communication Lost Fault (F702)
Units	N/A
Range	N/A

General Information	
Default Value	N/A
Data Type	Integer
Start Version	M_01-13-11-000

Variants Supported

Variant	Supported
AKD Base	No
AKD with Position Indexer	No
AKD EtherCAT	No
AKD CANopen	No
AKD BASIC	No
AKD SynqNet	No
AKD EtherNet/IP	No
AKD Profinet	Yes
AKD sercos® III	No
AKD-N	No
AKD-C	No

Fieldbus Information

Fieldbus	Address	Attributes	Signed?
Profinet	3022	DWord	No

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	2860	No	32 Bit	No	M_01-13-11-000

Description

Reads the timeout fault threshold for the Profinet Communication Lost Fault (F702). This is the time that is assigned to the drive via the Profinet master. To change the value, set the “watchdog time” or “watchdog factor” through the master. Be advised that setting a high timeout value can lead to the drive not recognizing a lost connection.

7 Sample Projects

7.1 Sample S7 Project

7.1.1 Introduction

On our website www.kollmorgen.com, you can find an STEP 7 sample project which provides a PROFINET network with an IO-controller and the AKD as IO-device.

The sample project can help you to learn:

- how to enable the drive
- how to write/read a parameter via the acyclic channel
- how the cyclic data exchange is done

The sample project is based on a CPU-315 controller, which easily can be changed to another PROFINET supporting controller.

NOTE Kollmorgen does not guarantee correctness.

7.1.2 Project description

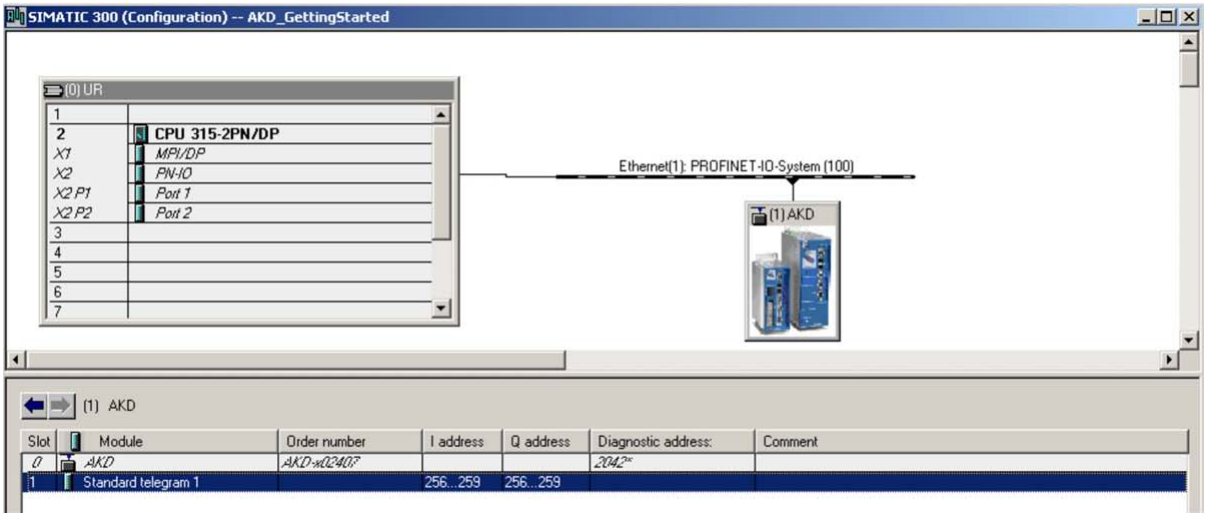
You will find in the STEP 7 program three organization blocks that need to be implemented.

- OB1, which is used for the main program and is a cyclic process.
- OB40, which is used for any process alarm (needs to be implemented the CPU from STOP to RUN).
- OB82, which catches the diagnostic alarms.

Two variable tables are included

- TG1_IO_DATA can be used to control easily the IO data between the plc and the AKD
- PARAMETER_ACCESS table is used for read/write PNU's via PROFINET

In the hardware manager, you can see the following setup :



7.1.3 Getting started

1. To use this example project, open the SIMATIC manager and retrieve the project zip file (SimaticManager->File->Retrieve).
2. After the project is loaded, go to the hardware manager and check the communication setup. If no AKD GSDML file is installed, the hardware manager will install it from the project. If there is already an installation of the GSDML file, this step is not required.
3. Check the communication setup for your system in the hardware manager and adapt it to your settings.
   The initial setup for AKD is a static IP Address and an IO cycle time of 128 ms. Change the IP addresses for AKD and the PLC to your specific setup.
4. Verify the hardware configuration in the hardware manager and click the “save and compile” button in your configuration.
   Verify the process-image input and output area of your plc and verify that is greater than 256 byte or change the input/output start address of the AKD telegram you choose. By default, the input is copied to start address 256 and the output to start address 256.
5. In this example, the AKD shall be used in velocity mode. Therefore the “Standard Telegram 1” (PROFIdrive) is chosen. The signals control word, speed set point as well as status word and speed actual value are mapped to the IO-Data.
   The AKD has to be set to DRV.OPMODE 1 for this operation.

.

7.1.4 Enable the drive and run in velocity mode

The general state machine of the PROFIdrive ("General State Machine" (➜ p. 26)) needs to be toggled to enable the drive. You can find in the variable table “TG1_IO_DATA” all necessary input and output parameters described bit-wise.

To enable the drive, write the following sequence in the control world:

1. QW 2#0000_0100_0000_0000 -> Remote control over field bus
2. QW 2#0000_0100_0000_0110 -> Go to S2 (Switch on inhibited)
3. QW 2#0000_0100_0000_1110 -> Go to S3 (Switched on)
4. QW 2#0000_0100_0000_1111 -> Go to S4 (Operational)

The drive is enabled, if the corresponding bits in the status word are set. TIP: If WorkBench is connected, you will see the drive is enabled.

TG1_IO_DATA -- @AKD_GettingStarted\SIMATIC 300\CPU 315-2PN/DP\S7-Programm(1) ONLINE

	Address	Symbol	Display format	Status value	Modify value
1	I 257.0	"SW_Bit0_ReadyToSwitchOn"	BOOL	true	
2	I 257.1	"SW_Bit1_ReadyToOperate"	BOOL	true	
3	I 257.2	"SW_Bit2_OperationEnabled"	BOOL	true	
4	I 257.3	"SW_Bit3_FaultPresent"	BOOL	false	
5	I 257.4	"SW_Bit4_CoastStopNotAct"	BOOL	true	
6	I 257.5	"SW_Bit5_QuickStopnotAct"	BOOL	true	
7	I 257.6	"SW_Bit6_SwitchOnInhibted"	BOOL	false	
8	I 257.7	"SW_Bit7_WarningPresent"	BOOL	false	
9	I 256.0	"SW_Bit8_SpeedErrorInRang"	BOOL	false	
10	I 256.1	"SW_Bit9_ControlRequested"	BOOL	true	
11	I 256.2	"SW_Bit10_FornReached"	BOOL	false	
12	I 256.3	"SW_Bit11_DeviceSpecific"	BOOL	false	
13	I 256.4	"SW_Bit12_DeviceSpecific"	BOOL	false	
14	I 256.5	"SW_Bit13_DeviceSpecific"	BOOL	false	
15					
16	Q 257.0	"CW_Bit0_SwichtedOn"	BOOL	true	
17	Q 257.1	"CW_Bit1_NoCoastStop"	BOOL	true	
18	Q 257.2	"CW_Bit2_NoQuickStop"	BOOL	true	
19	Q 257.3	"CW_Bit3_EnableOperation"	BOOL	true	
20	Q 257.4	"CW_Bit4_EnableRampGen"	BOOL	true	
21	Q 257.5	"CW_Bit5_UnfreezeRamp"	BOOL	true	
22	Q 257.6	"CW_Bit6_EnableSetpoint"	BOOL	true	
23	Q 257.7	"CW_Bit7_FaultAck"	BOOL	false	
24	Q 256.0	"CW_Bit8_JogOn"	BOOL	false	
25	Q 256.1	"CW_Bit9_Jog2On"	BOOL	false	
26	Q 256.2	"CW_Bit10_ControlByPlc"	BOOL	true	
27	Q 256.3	"CW_Bit11_StartHoming"	BOOL	false	
28	// control word				
29	QW 256		BIN	2#0000_0100_0111_1111	2#0000_0100_0111_1111
30	// status word				
31	IW 256		HEX	W#16#0237	
32	// actual velocity				
33	IW 258		HEX	W#16#009E	
34	// set point velocity				
35	QW 258		HEX	W#16#00A3	W#16#00A3
36					

Now the set point velocity can be set. In the example (see variable table) the value is QW 258 is 0xA3 (60 rpm ("Units" (➜ p. 57))). To start the motion, set Bit 4, 5 and 6 (enable ramp generator, unfreeze ramp generator and enable new set point).

8 Troubleshooting

8.1 AKD Triggers 702 Communication Fault

The root cause could be:

1. High traffic on the Profinet network
2. Overloaded PLC, which cannot send the I/O packages within the watchdog time
3. Overload of the AKD from too many communication requests (I/O data, PNU access and WorkBench connection)

Possible solutions:

1. Create a subnet for your Profinet network and use a Profinet switch
2. Reduce the runtime of your PLC program
3. Increase the watchdog time for a communication timeout in the AKD Profinet slave. Or if you have WorkBench running, disable device discovery to decrease the communication load.

9 Record of Document Revisions

Revision	Remarks
A, 10/2011	Launch version
B, 03/2012	Jog mode added, mdi submode description extended
C, 08/2012	Manufacturer specific telegram 351
D, 05/2013	Corrections, formatting acc. to 82079
E, 12/2013	Added bit 14 for position mode in Status word bits (ZSW1) (➜ p. 29).
F, 05/2014	Added Signal No. 100 and 101 to Telegram configuration (➜ p. 48). Added Manufacturer specific telegram 352 to I/O Telegrams (➜ p. 54).
G, 12/2014	Manufacturer specific telegram 353 (➜ p. 56) added. Signal number 102 and 103 added to Telegram configuration (➜ p. 48). Acc-/Deceleration Units (➜ p. 57) scaling information added.
H, 12/2015	Added Signal numbers 104 to 107 to Telegram configuration (➜ p. 48). Added PN Parameters (➜ p. 59) chapter. Added PNU 28 to Supported PNU's (➜ p. 30). Added Standard telegram 400 (➜ p. 56).
J, 03/2016	Section Safety removed, chapter target group added, warning notes format updated
K, 09/2016	Updated instructions to Submode „Program mode“ (➜ p. 50).

This page intentionally left blank.

10 Index

This page intentionally left blank.

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
203A 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

South America
KOLLMORGEN
Avenida Tamboré, 1077 Tamboré -
Barueri - SP Brasil
CEP: 06460-000

Web:	www.kollmorgen.com
Tel.:	+55 11 4191-4771

Europe
KOLLMORGEN Europe GmbH
Pempelfurtstraße 1
40880 Ratingen
Germany

Web:	www.kollmorgen.com
Mail:	technik@kollmorgen.com
Tel.:	+49 - 2102 - 9394 - 0
Fax:	+49 - 2102 - 9394 - 3155

China and SEA
KOLLMORGEN
Room 202, Building 3, Lane 168,
Lin Hong Road, Changning District
Shanghai

Web:	www.kollmorgen.cn
Mail:	sales.china@kollmorgen.com
Tel.:	+86 - 400 661 2802