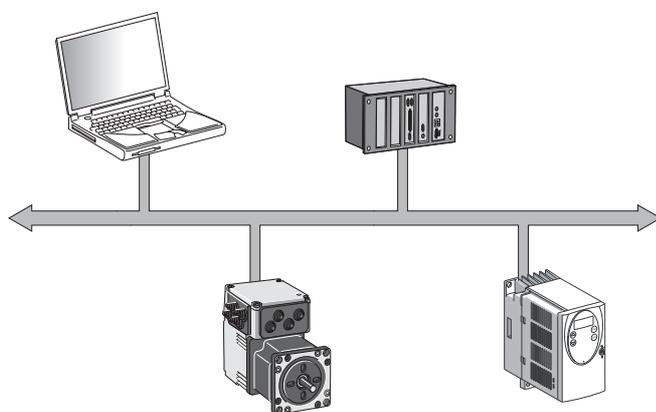


SD328B Profibus DPVO

Fieldbus interface

Fieldbus manual

V2.01, 11.2008



Important information

This manual is part of the product.

Carefully read this manual and observe all instructions.

Keep this manual for future reference.

Hand this manual and all other pertinent product documentation over to all users of the product.

Carefully read and observe all safety instructions and the chapter "Before you begin - safety information".

Some products are not available in all countries.

For information on the availability of products, please consult the catalog.

Subject to technical modifications without notice.

All details provided are technical data which do not constitute warranted qualities.

Most of the product designations are registered trademarks of their respective owners, even if this is not explicitly indicated.

Table of Contents

- Important information 2**
- Table of Contents 3**
- Writing conventions and symbols 5**
- 1 Introduction 7**
 - 1.1 Documentation and literature references 7
 - 1.2 Directives and standards 7
- 2 Before you begin - safety information 9**
- 3 Basics 11**
 - 3.1 Profibus technology 11
 - 3.1.1 Profibus transmission technology 11
 - 3.1.2 Network topology 11
 - 3.1.3 Access procedures 12
 - 3.1.4 Transmission technology in the network 12
 - 3.1.5 Device identification 12
 - 3.2 Fieldbus devices on the Profibus DP network 13
 - 3.3 Operating modes and functions 13
- 4 Installation 15**
- 5 Commissioning 17**
 - 5.1 Prerequisites for commissioning 17
 - 5.2 Starting operation on the network 18
- 6 Operation 19**
 - 6.1 Profibus parameters 19
 - 6.2 Profibus communication profile 20
 - 6.2.1 Profibus DP V0 communication 20
 - 6.2.2 Data structure 21
 - 6.2.3 Parameter channel and process data channel 21
 - 6.3 Parameter channel 22
 - 6.3.1 Overview 22
 - 6.3.2 Structure of the parameter channel 23
 - 6.4 Process data channel 26
 - 6.4.1 Overview 26
 - 6.4.2 Structure of the process data channel 26
 - 6.5 PLC as a fieldbus master 36
 - 6.5.1 Response in the case of special master commands 38

019844113724, V2.01, 11.2008

7	Examples	39
7.1	Overview of examples	39
7.2	Use of the parameter channel	40
7.2.1	Writing parameters	40
7.2.2	Reading a parameter	41
7.2.3	Synchronous errors	41
7.3	Operating states in the process data channel	42
7.3.1	Enabling and disabling the power stage	43
7.3.2	Triggering a Quick Stop	44
7.3.3	Resetting faults	44
7.4	Operating modes in the process data channel	45
7.4.1	Absolute positioning	47
7.4.2	Relative positioning	48
7.4.3	Profile Velocity	49
7.4.4	Homing by position setting	50
7.4.5	Reference movement	51
7.5	Error signaling in the process data channel	52
7.5.1	Synchronous errors	52
7.5.2	Asynchronous errors	54
8	Diagnostics and troubleshooting	55
8.1	Fieldbus communication error diagnostics	55
8.2	Error messages	56
8.2.1	Synchronous errors	56
8.2.2	Asynchronous errors	57
8.2.3	Errors during operating mode control	57
9	Glossary	59
9.1	Units and conversion tables	59
9.1.1	Length	59
9.1.2	Mass	59
9.1.3	Force	59
9.1.4	Power	59
9.1.5	Rotation	60
9.1.6	Torque	60
9.1.7	Moment of inertia	60
9.1.8	Temperature	60
9.1.9	Conductor cross section	60
9.2	Terms and Abbreviations	61
10	Index	63

Writing conventions and symbols

Work steps If work steps must be performed consecutively, this sequence of steps is represented as follows:

- Special prerequisites for the following work steps
- ▶ Step 1
- ◁ Specific response to this work step
- ▶ Step 2

If a response to a work step is indicated, this allows you to verify that the work step has been performed correctly.

Unless otherwise stated, the individual steps must be performed in the specified sequence.

Bulleted lists The items in bulleted lists are sorted alphanumerically or by priority. Bulleted lists are structured as follows:

- Item 1 of bulleted list
- Item 2 of bulleted list
 - Subitem for 2
 - Subitem for 2
- Item 3 of bulleted list

Making work easier Information on making work easier is highlighted by this symbol:



Sections highlighted this way provide supplementary information on making work easier.

SI units SI units are the original values. Converted units are shown in brackets behind the original value; they may be rounded.

Example:

Minimum conductor cross section: 1.5 mm² (AWG 14)

1 Introduction

Profibus is a serial fieldbus system in which products from different manufacturers can be networked without the need for special interface adaptation.

This manual describes online command processing for products in the Profibus DP V0 fieldbus network.

1.1 Documentation and literature references

Manuals In addition to this fieldbus manual, the following manuals also belong to the product:

- **Product manual**, describes the technical data, installation, commissioning and all operating modes and functions.
- **Motor manual**, describes the technical characteristics of the motors, including correct installation and commissioning.

Literature

- PROFIBUS Specification (FMS, DP, PA); Profibus User Organization
- Popp, M: Profibus DP/DPV1; Grundlagen, Tipps und Tricks für Anwender; ISBN 3-7785-2781-9, Hüthig-Verlag Heidelberg

1.2 Directives and standards

Directives, standards

- IEC 61158 Digital data communications for measurement and control, fieldbus standard

Profibus User Association Profibus-Nutzerorganisation e.V. (PNO)
Profibus User Association
Haid-und-Neu-Str. 7
D-76131 Karlsruhe

Profibus international on the Internet <http://www.profibus.com>

2 Before you begin - safety information

The information provided in this manual supplements the product manual. Carefully read the product manual before you begin.

3 Basics

3.1 Profibus technology

3.1.1 Profibus transmission technology

Three Profibus versions are available that can be used for time-critical and complex communication tasks:

- Profibus-FMS
- Profibus-PA
- Profibus DP

Profibus - FMS Profibus - FMS (FMS: Fieldbus Message Specification) is a universal, flexible solution for communication tasks in general automation technology. For example, FMS is used for communication between manufacturing cells.

Profibus - PA Profibus - PA (PA: Prozess-Automation) is primarily used in process technology, such as process automation. Profibus-PA networks are characterized by their ability to use sensors and actuators in hazardous locations (Ex areas, explosive atmospheres), and to provide data communication and power to devices via the bus.

Profibus - DP Profibus - DP (DP: Dezentrale Peripherie) is the fast Profibus version which is specially tailored for communication in production processes. Features include simple networking of new products in the bus and high transmission rates.

The product with Profibus DP described in this manual supports various parameterization telegrams as per Profibus DPVO specification.

3.1.2 Network topology

A Profibus DP network consists of one or more masters (active bus devices) and slaves (passive bus devices). All bus devices are connected via the Profibus DP network cable.

Master The master controls the data traffic on the network. Examples of masters:

- Automation devices, e.g. PLCs
- PCs
- Programming devices.

Slave Slaves receive control commands and supply data to the master. Examples of slaves:

- Input/output modules
- Drive systems
- Sensors and actuators.

3.1.3 Access procedures

There are two possible access procedures resulting from the arrangement of network devices on the bus:

- Token-Passing method
- Master-Slave method

Token-Passing method The Token-Passing method is used between multiple master in a Profibus DP network. The masters form a logical token ring in which every master receives transmission authorization for a certain period of time.

Master-Slave method Data is exchanged with the product with the master-slave method. The slave has a transmit and receive buffer which it uses to transmit and receive data. The master reserves a memory area with a transmit and receive buffer for each slave.

Data exchange between master and slave is cyclical. The master sends commands to the slave and receives data back from the slave in the next cycle. The bus cycle is extended for repeated transmission of telegrams only in the case of errors.

The products are networked as slaves, therefore, they do not use the token passing.

3.1.4 Transmission technology in the network

Profibus DP networks can be set up with optical fiber cables or with RS-485 technology.

The products work with RS-485 technology and are connected to a Profibus DP network with two-wire cables.

RS-485 technology RS-485 is a simple method of transmission via two-wire twisted-pair cables. This technology allows for transmission rates between 9.6 kBit/s and 12 MBit/s.

3.1.5 Device identification

Generic Station Description GSD The specific features of a Profibus product are described in the Generic Station Description file (GSD file). The manufacturer supplies this file along with the product ; the file must be read by the network configuration program.

The GSD file contains all information on the operation of the product in the Profibus DP network, such as vendor-specific information and product designation, supported baud rate, levels and meaning of bus connector signals, time intervals for monitoring times and product-specific values for network devices such as settings for inputs/outputs. The GSD file for this product is available for download from the Internet.

Ident number A master device uses the Ident number to identify the device class of the connected slave. The Ident number is a unique number assigned to a specific device class by the Profibus user organization.

Slave address Each device on the network must be assigned a unique address between 1 and 126; slaves normally use the address range 3 ... 126. The master (normally address 0 ... 2) can communicate with each slave via this address. Details on setting the address for the product described here can be found in the product manual.

3.2 Fieldbus devices on the Profibus DP network

Different fieldbus products can be operated in the same fieldbus segment. Profibus DP provides a common basis for interchanging commands and data between the network devices.

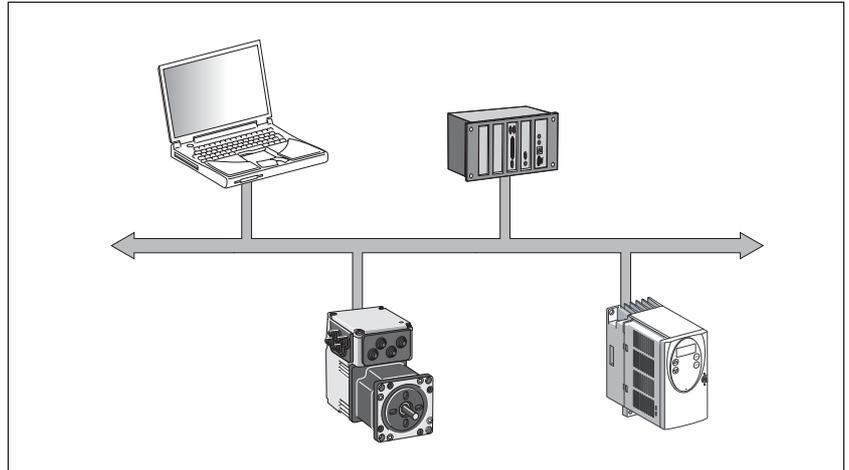


Figure 3.1 Fieldbus products on the network

3.3 Operating modes and functions

This manual only describes the protocol for fieldbus operation. Descriptions of operating modes, functions and all parameters can be found in the product manual.

Settings The following settings can be made via the fieldbus:

- Reading and writing parameters
- Monitoring inputs and outputs
- Diagnostics and error monitoring functions

4 Installation

▲ WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are EMERGENCY STOP, overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical functions.
- System control paths may include communication links. Consideration must be given to the implication of unanticipated transmission delays or failures of the link.
- Observe the accident prevention regulations and local safety guidelines.¹⁾
- Each implementation of the product must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death or serious injury.

1) For USA: Additional information, refer to NEMA ICS 1.1 (latest edition), Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control and to NEMA ICS 7.1 (latest edition), Safety Standards for Construction and Guide for Selection, Installation for Construction and Operation of Adjustable-Speed Drive Systems.

▲ WARNING

SIGNAL AND DEVICE INTERFERENCE

Signal interference can cause unexpected responses of device.

- Install the wiring in accordance with the EMC requirements.
- Verify compliance with the EMC requirements.

Failure to follow these instructions can result in death, serious injury or equipment damage.

For information on installation of the device and connecting the device to the fieldbus see the product manual.

5 Commissioning

⚠ DANGER

UNINTENDED CONSEQUENCES OF EQUIPMENT OPERATION

When the system is started, the drives are usually out of the operator's view and cannot be visually monitored.

- Only start the system if there are no persons in the hazardous area.

Failure to follow these instructions will result in death or serious injury.

⚠ WARNING

UNINTENDED OPERATION

- Do not write values to reserved parameters.
- Do not write values to parameters unless you fully understand the function. For more information see the product manual.
- Run initial tests without coupled loads.
- Verify that the system is free and ready for the movement before changing parameters.
- Verify the use of the bits with fieldbus communication: bit 0 is far right (least significant). Bit 15 is far left (most significant).
- Verify the use of the word sequence with fieldbus communication.
- Do not establish a fieldbus connection unless you have fully understood all communications principles.

Failure to follow these instructions can result in death, serious injury or equipment damage.

5.1 Prerequisites for commissioning

The following components are required for commissioning:

- Product with fieldbus interface
- GSD file (can be downloaded from the internet)
- Product manual and fieldbus manual
- ▶ Carefully read and understand the manuals before commissioning and take particular note of the safety instructions!



Using the library considerably facilitates controlling the device. The library is available for download from the Internet.

<http://www.schneider-electric.com>

5.2 Starting operation on the network

Operation on the network is started via a master. This master can be a PLC or a PC with the appropriate application software that allows you to send commands and receive data.

⚠ WARNING

LOSS OF CONTROL

Even if data link monitoring is active, it may not be possible to stop a running movement due to an error in the master controller.

- Suitable redundant control paths must be provided for hazardous functions.

Failure to follow these instructions can result in death, serious injury or equipment damage.

Activating the watchdog at the master for all slaves starts monitoring of the data links for Profibus. In special cases, the monitoring function can be deactivated for the drive system only by means of the following parameter:

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
PBSafeState	Safe state	-	UINT16	Modbus 6154
-	0 / NoError: No response	0	UINT16	Profibus 6154
-	1 / ErrorClass2: Error of class 2, drive switches to FAULT if the power stage was active	1	R/W per.	-
	Response of the drive in state 'Clear' of the ProfibusDP master and response to termination of the watchdog.	1		

Steps for troubleshooting Test all functions that are important for your system. Run the function tests first without coupled load. Also check the operating temperature during normal operation and the response of the system to power outage.

If the slave does not send a response, check the following settings:

- Did you switch on the product and start the master for operation on the network?
- Are all cable connections ok (electrically and mechanically)?
- Did you set the correct address and baud rate?

For more information on the cause of errors and on troubleshooting see the product manual.

6 Operation

▲ WARNING

UNINTENDED OPERATION

- Do not write values to reserved parameters.
- Do not write values to parameters unless you fully understand the function. For more information see the product manual.
- Run initial tests without coupled loads.
- Verify that the system is free and ready for the movement before changing parameters.
- Verify the use of the bits with fieldbus communication: bit 0 is far right (least significant). Bit 15 is far left (most significant).
- Verify the use of the word sequence with fieldbus communication.
- Do not establish a fieldbus connection unless you have fully understood all communications principles.

Failure to follow these instructions can result in death, serious injury or equipment damage.

The chapter "Operation" describes the basic operating states, operating modes and functions of the product.



Using the library considerably facilitates controlling the device. The library is available for download from the Internet.

<http://www.schneider-electric.com>

6.1 Profibus parameters

Working with the Profibus protocol involves a number of specific parameters that are explained in this fieldbus manual. Please consult the product manual for an overview of all parameters including explanations and examples.

Parameters	Meaning, reference	Page
PBMapIn	Mapping of PZD5 and PZD6, drive ---> master	31
PBMapOut	Mapping of PZD5 and PZD6, master ---> drive	31
PBPkInhibit	Inhibit time for read requests	23
PBSafeState	Response of the drive when the Profibus master is in "Clear" state	38
PBFltPpo	Response to incorrect processing of the process data channel	38

6.2 Profibus communication profile

6.2.1 Profibus DP V0 communication

Profibus DP V0 provides the basic functionality of DP. This includes cyclic data exchange, station-, module- and channel-specific diagnostics and various alarm types for diagnostics and process alarms for hot-plugging bus devices.

Master-slave relationship

The master cyclically writes transmit data to the slaves and cyclically reads the receive data provided by the slaves. Receive and transmit data for one slave are transmitted as one unit in one message cycle.

Command processing: Transmit data and receive data

The master sends a command to the product (slave) to start operating modes and functions, execute a motion command or request information from the slave. The slave executes the command and acknowledges it with a confirmation.

The exchange of data follows a fixed pattern:

- Transmit data to the slave: The master places a command in the transmit data memory. From there, it is transmitted to the slave and executed.
- Receive data from the slave: the slave acknowledges the execution status of the command in the receive data. If the master device receives an acknowledgement without an error message, the command was correctly executed.

The master device can send new commands as soon as it has received acknowledgement concerning the current command. Acknowledgement information and error messages are included in the transmitted data in bit-coded form.

Due to the cyclical fieldbus transmission, the master device automatically receives current receive data from the slave device with every cycle. The acknowledgement information allows the master to detect whether the receive data was status information from the slave or the response to a previously transmitted command.

Commands

The master transmits control commands and action commands with the transmit data.

- After a control command the master receives a feedback message from the slave indicating whether the command could be successfully executed and processing be completed.
- In the case of an action command, the slave only signals whether it was possible to successfully start an action or a motion command.

The master must then continuously monitor for completion of the processing command by evaluating the receive data of the slave.

6.2.2 Data structure

In addition to command and control information, transmit and receive data also contain network administration data. The administration data is provided by the application program of the master.

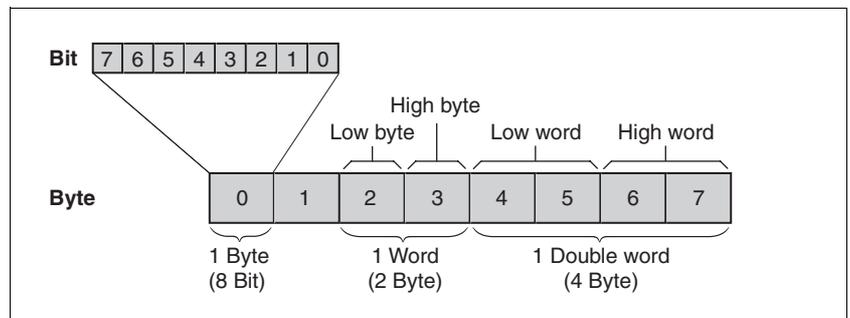


Figure 6.1 General data structure from the bit to the double word

Index and subindex in the descriptions of the parameters are shown in decimal notation. The data telegram with transmit data and receive data as well as all byte, word and double word values are shown in hexadecimal notation. Hexadecimal values are indicated as such by means of an "h" behind the numerical value, e.g. "31_h", decimal values have no special identification.

NOTE: Note the different counting format of bits (0 ... 7, right to left) and bytes (1-xx, left to right).

Word order: Big Endian format

Profibus data is transmitted in Big Endian format, i.e. numerical values over one byte are treated as in the decimal system. Example: the index value is transmitted in bytes 3 and 4, the index 102_h is thus represented as 0102_h.

6.2.3 Parameter channel and process data channel

Communication between master and slave is based on the PPO type 2 data telegram of the Profidrive profile of the PNO user organization. The content of the data does not correspond to the Profidrive profile! The data telegram consists of 20 bytes. The first 8 bytes are used for parameter transmission, the following 12 bytes (bytes 9 ... 20) transmit the process data. They are interpreted depending on the operating mode.

Parameter channel				Process data channel					
Bytes 1-2	Bytes 3-4	Bytes 5-6	Bytes 7-8	Bytes 9-10	Bytes 11-12	Bytes 13-14	Bytes 15-16	Bytes 17-18	Bytes 19-20
PKE	IDX	PWE	PWE	PZD1	PZD2	PZD3	PZD4	PZD5	PZD6

Meaning of the abbreviations

PKE	Parameter identifier	2 bytes
IDX	Index of parameter (parameter number)	2 bytes
PWE	Parameter value	4 bytes
PZD	Process data	12 bytes

Index, subindex The index uses bytes 3 and 4 of the data telegram; the subindex corresponds to byte 2.

The parameters are addressed using a 16 bit index and an 8 bit subindex. A data field consists of the value of a subindex entry. The individual data fields of a parameter are shown in decimal notation and may have to be converted into hexadecimal format. Hexadecimal values are shown with a subscript "h". In the case of the family of devices described, the subindex is 00. The index corresponds to the parameter address. The following example shows index and subindex entries for jog configuration.

Index	Subindex	Parameters	Meaning
10504	00	JOGn_slow	Speed of rotation slow jog
10506	00	JOGn_fast	Speed of rotation fast jog
10510	00	JOGstepusr	Jog distance prior to continuous run
10512	00	JOGtime	Waiting time prior to continuous run

Table 6.1 Examples of index and subindex entries

See the product manual for a list of all parameters.



See the parameter group overview in the chapter "Parameters" for the number format of the parameter values in a fieldbus command.

6.3 Parameter channel

6.3.1 Overview

The master can request a parameter value from the slave or change a parameter value via the parameter channel (the first 8 bytes of the 20 bytes data telegram). Each parameter can be uniquely addressed via the index and subindex.

6.3.2 Structure of the parameter channel

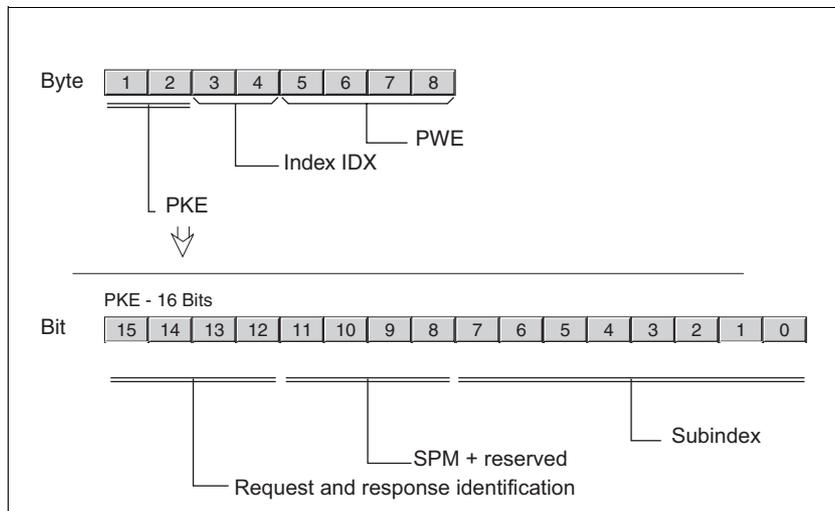


Figure 6.2 Parameter channel: Parameter identifier in bytes 1 and 2

The following abbreviations are used for the parameter channel:

Bytes 1 and 2: PKE as parameter identifier (request identifier and subindex)

Bytes 3 and 4: IDX for index of parameter (parameter number)

Bytes 5 ... 8: PWE for parameter value

Parameter identifier PKE

The first two bytes (parameter identifier PKE) contain the request identifier or the response identifier as well as the subindex of the parameter. In the case of this device family, the subindex is 0. Bytes 3 and 4 contain the index (IDX). The parameter value is contained in bytes 5 ... 8 (PWE).

The request identifier or response identifier (bits 12 ... 15) indicates the fields of the parameter channel to be evaluated.

Request identifier	Function	Response identifier	
		positive	negative
0	No request	0	7
1	Request parameter value (word)	1	7
1	Request parameter value (double word)	2	7
2	Change parameter value (word)	1	7
3	Change parameter value (double word)	2	7

Table 6.2 Request identifier and response identifier

NOTE: Write requests (change parameter value) are only executed by the slave if the value of the request identifier changes from 0 to 2 or 3.

Read requests are executed as often as the value of request identifier is equal to 1. In order to limit the system load, a cycle time between two read processes is defined with the parameter `PBPkInhibitbit`. The read request is not executed again until this cycle time has elapsed.

If the response identifier to a request is 0, the slave has not yet completed the request. The slave signals to the master that the request has been successfully executed via response identifier 1 or 2 (positive response identifier). The slave signals to the master that an error has occurred via response identifier 7 (negative response identifier). In the case of a negative response identifier, bytes 5 ... 8 (parameter value) contains the error number; bytes 5 and 6 contain the value 0 and bytes 7 and 8 a 16 bit value.

Only one request can be processed at a time. The slave provides the response until the master sends a new request. If a response includes parameter values, the slave responds with the current value in the case of a repetition (cyclic processing).

Bits 8 ... 11 (reserved) must be 0.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
PBPkInhibit	Inhibit time during read tasks in the parameter channel	ms 0 1000 65535	UINT16 UINT16 R/W per.	Modbus 6152 Profibus 6152
-	In the case of a static read task, the reader value is cyclically updated according to the wait time defined with this parameter. 0: No wait time >0: Wait time in ms		-	

Example: correct reading of a parameter

In the example, the program number of the Profibus software is to be read. The program number is stored in parameter `_prgNoDEV` (index 258; subindex 00). The master sends a read request to the slave. After processing, the slave provides the requested data in bytes 5 ... 8 (parameter value PWE). The parameter value read has the decimal value 8470 which corresponds to 2116_{16} . Because this is a word, the corresponding positive response identifier must be 1.

The master sends the following transmit data to the slave (values that are not relevant for the example are represented by x):

Transmit data: index: $258=102_{16}$, subindex: 00

Parameters	PKE, 1st byte request identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Tx $0102_{16}:00_{16}$	10_{16}	00_{16}	0102_{16}	xxxx xxxx	Reading the program number. The data has no significance.

The 4 data bytes have no significance for a read request.

Receive data:

Parameters	PKE, 1st byte response identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Rx $0102_{16}:00_{16}$	10_{16}	00_{16}	0102_{16}	$0000\ 2116_{16}$	Data 2116 correspond to the program number.

Based on the response identifier (2 or 1) in the PKE (bits 12 and 13), the product distinguishes between parameter values with 32 bit data and parameter values with 16 bit data (also described in the product manual as INT32 or UINT32 and INT16 or UINT16 data types). In the case of 16 bit data, the usable information is contained in the last two data bytes, the first two data bytes have no significance.

The message keeps being provided until the master sends the request identifier 0 to the slave prior to the next request.

Example: incorrect writing of a parameter

After the master has read the information from the above example, you must first reset the slave reset with the request identifier "No request" (PKE:00).

The slave is then ready to execute new request. In this example, the value of a non-existent parameter is to be changed. The value of the parameter with index 101 = 00h 65_h and subindex 00 is to be changed to 222 = DE_h.

Index: 101=0065_h

Subindex: 00 = 00_h

Value: 222 = 0000 00DE_h

Parameters	PKE, 1st byte request identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Tx 0065 _h :00 _h	30 _h	00 _h	0065 _h	0000 00DE _h	Writing a non-existent parameter

Because the slave cannot address the parameter, an error message is returned; the parameter value in this case is 0000 B30A_h. Error messages in the parameter channel are referred to as synchronous errors because they are processed during regular cyclic data exchange.

Parameters	PKE, 1st byte response identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Rx 0065 _h :00 _h	70 _h	00 _h	0065 _h	0000 B30A _h	Error message 0000B30A _h is returned = parameter does not exist

For information on synchronous errors and error numbers see page and page onwards in the respective product manual.

6.4 Process data channel

6.4.1 Overview

The process data channel is used for real-time data exchange of process data such as actual position and reference position, current operating state. Transmission is very fast because the data is sent without additional administration data and a response from the recipient is not required.

In addition, the master can control the operating state of the slave via the process data channel, e.g. enable and disable the power stage, trigger and reset a Quick Stop, reset errors and enable operating modes.

Changing operating states and activating operating modes must be executed separately. An operating mode can normally only be activated if the operating state is "Operation Enable".

NOTE: A new operating mode and a new acceleration can only be set if the motor is at a standstill. Acceleration values are accepted in the process data channel during motor movements, but the value is only set during the next motion command. All other information can be changed with an active operating mode.

6.4.2 Structure of the process data channel

The 12 bytes of the process data channel have the following designation:

Bytes 9 ... 20: PZD1 ... 6, process data in word format

The data formats of transmit data (master to slave) and receive data (slave to master) differ as follows:

Transmit data format

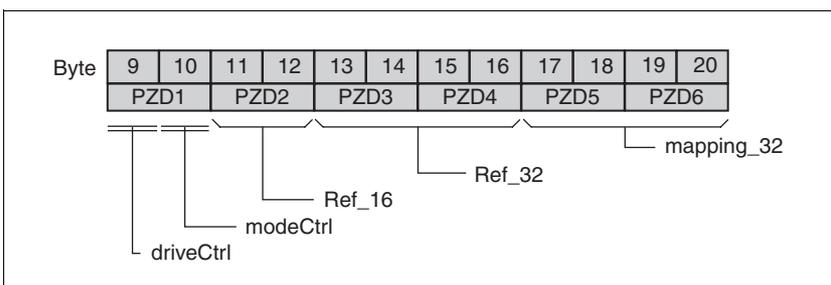


Figure 6.3 Transmit data in the process data channel: master to slave

Bytes 9 and 10: driveCtrl and modeCtrl for setting the operating state and the operating mode. The structure is explained below, see page .

Bytes 11 and 12: ref_16, 16 bit reference value, assignment depends on operating mode, for example for reference speed.

Bytes 13 ... 16: ref_32, 32 bit reference value, assignment depends on operating mode, e.g. Pos for position data (consisting of high word=Pos1 and low word=Pos2).

Bytes 17 ... 20: mapping, these bytes can be parameterized, the content is specified via index and subindex. The default value is the acceleration (32 bit): acc (consisting of high word=acc1 and low word=acc2).

The the parameters `PBMapOut` and `PBMapIn` are used to define the parameter to be mapped to the PZD5 and PZD6. During parameterization, a check is run as to whether the value is permissible. If mapping is deactivated, the data in bytes 17 ... 20 is not relevant.

Description `driveCtrl`:

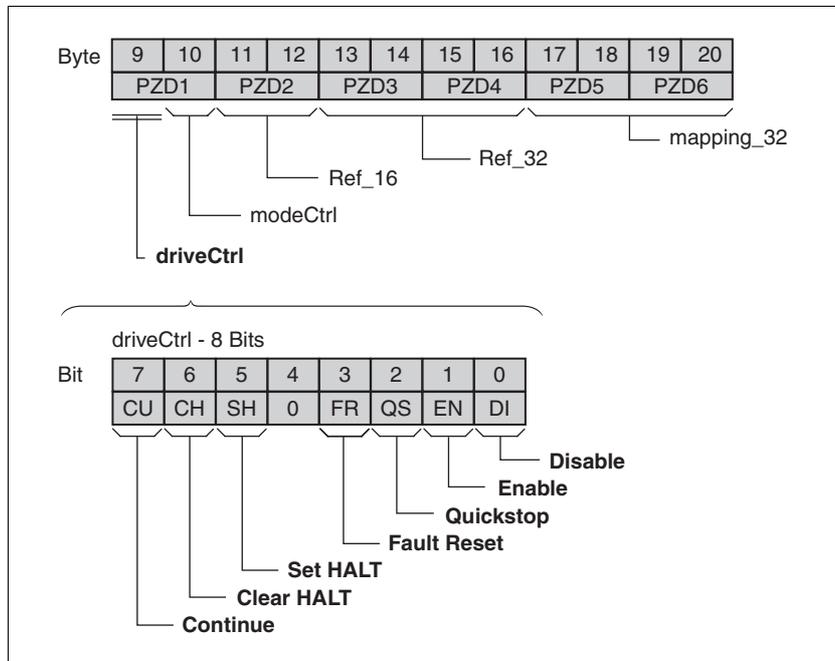


Figure 6.4 Transmit data in the process data channel: `driveCtrl`

The operating states are changed via the process data channel PZD1 `driveCtrl`, bits 0 ... 7 . The corresponding functions are executed in the case of a change from 0 to 1.



NOTE: The Enable bit must be set as long current is top be applied to the motor.

Change of operating state ^{1) 2)}	Effects on operating states ³⁾
Bit 0: Power stage Disable	6 - 3 - 4 (Operation enable ---> Switch on disable ---> Ready to switch on)
Bit 1: Power stage Enable	4 - 5 - 6 (Ready to switch on ---> Switched on ---> Operation Enable)
Bit 2: Quickstop	6 - 7 (Operation enable ---> QuickStop active)
Bit 3: Fault Reset	9 - 3 - 4 (Fault ---> Switch on disable --->Ready to switch on)
Bit 4: 0	Reserved
Bit 5: Set HALT	Set HALT
Bit 6: Clear HALT	Clear HALT
Bit 7: Continue	Resume operating mode interrupted by HALT

1) Process data channel: Processing is started when 0 changes to 1.
 2) Parameter channel: Processing is started if bit value = 1
 3) The operating states and the state transitions are described in the product manual

Table 6.3 Change of operating state (`driveCtrl`)

The value Zero is a special case: If during transmission all bits 0 ... 7 are zero, the product interprets this as `Disable` command and disable the power stage.

Handling of errors If requests for changing the operating state cannot be executed, these requests are ignored. There is no error response. The assignment of errors to error classes can be parameterized.

Ambivalent bit combinations are treated in accordance with the following list:

- Bit 0 (power stage Disable) has priority over bit 1 (power stage Enable)
- Bit 2 (Quick Stop) has priority over bit 3 (Fault Reset))
- Bit 5 (Set Halt) has priority over bit 6 (Clear HALT) and bit 7 (Continue)

Description modeCtrl:

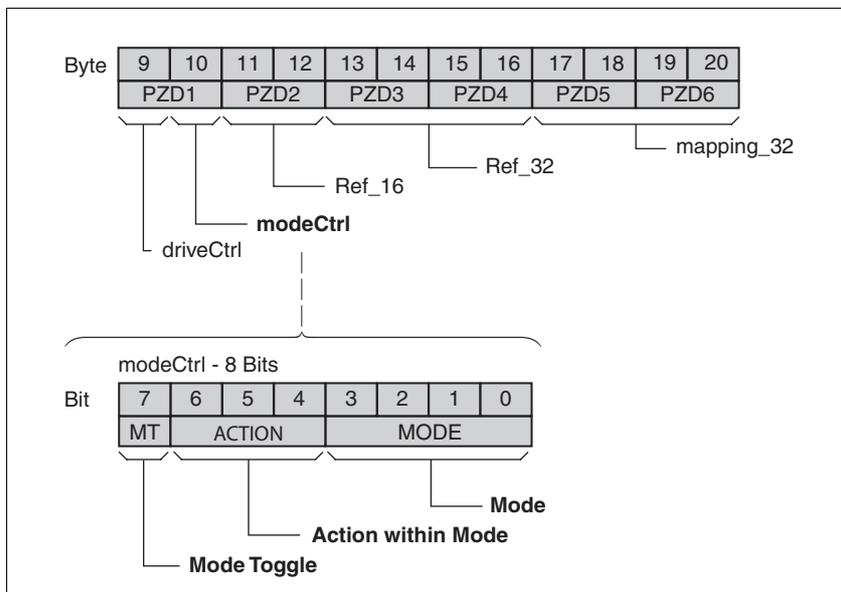


Figure 6.5 Transmit data in the process data channel: `modeCtrl`

The operating modes are controlled by means of `modeCtrl`. The master must enter the following values to activate an operating mode or to change reference values:

- Reference values in fields PZD2, PZD3 and PZD4
- Select operating mode with `modeCtrl`, bits 0 ... 2 (MODE). Bit 3 must contain 0!
- Select action for this operating mode with `modeCtrl`, bits 4 ... 6 (ACTION)
- Toggle `modeCtrl`, bit 7 (MT)

The table below shows the possible operating modes, functions and the corresponding reference values.

Operating mode	modeCtrl ¹⁾	Description	Reference value ref_16, PZD2	Reference value ref_32, PZD3 and 4
Jog	01h	Jog - classical jog	Operation (selection of direction of rotation and speed of rotation) As JOGactivate	-
Homing	02h	Position setting	-	Position for position setting As HMp_absusr
	12h	Reference movement	Type of reference movement As HMmethod	-
Profile position	03h	Absolute positioning	Reference speed As PPn_target	Reference position As PPp_absusr
	13h	Relative positioning with reference to the currently set target position	Reference speed As PPn_target	Reference position As PPp_relprefusr
	23h	Relative positioning with reference to the current motor position	Reference speed As PPn_target	Reference position As PPp_relpactusr
Profile Velocity	04h	Profile Velocity	Reference speed As PVn_target (16 bit only!)	
Electronic gear	05h	Electronic gear, immediate synchronization	Denominator of gear ratio As GEARdenom (16 bit only!)	Numerator of gear ratio As GEARnum
	15h	Electronic gear, synchronization with compensation movement	Denominator of gear ratio As GEARdenom (16 bit only!)	Numerator of gear ratio As GEARnum
Speed control	17h	Speed control	Reference speed As SPEEDn_target	-

1) Column corresponds to the value to be entered in byte modeCtrl, however, without ModeToggle (Bit 7)

Table 6.4 Setting operating modes via modeCtrl

Reference positions are entered in increments in [usr], reference speeds in [min⁻¹].

▲ WARNING

UNINTENDED OPERATION

- Note that any changes to the values of these parameters are executed by the drive controller immediately on receipt of the data set.
- Verify that the system is free and ready for movement before changing these parameters.

Failure to follow these instructions can result in death, serious injury or equipment damage.

If operating mode, reference position and reference speed are simultaneously transmitted via the process data channel, the data must be consistent. Therefore, operating mode data are only evaluated if MT (Bit 7) has been toggled. Toggling means that since the last transmission a change from 0 to 1 or from 1 to 0 has been detected for this bit.

MT (Bit 7) is mirrored in the receive data set which allows the master to determine that the data has been accepted by the slave.

For more information on the toggle flag see page 45.

Mapping The following parameters can be entered in PZD5 and PZD6: Temperatur, Spannung, Fehlernummer, Strom. . The error number (matches status word) and the IO word can be set consistently. The parameters- `PBMapIn` (drive \Rightarrow master) and `PBMapOut` (master \Rightarrow drive) let you define the parameter to be mapped to PZD5 and PZD6 for the transmit data and/or the receive data. During parameterization, a check is run as to whether the value is permissible. If mapping has been disabled, there are no defined values in bytes 17 ... 20.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
PBMapIn	Mapping of PZD5+6 to master	-	UINT32	Modbus 6150
-	A parameter with a data length of 32 bits can be used for mapping the PZD5+6 from the drive to the master.	0	UINT32	Profibus 6150
-		0	R/W	
-		65535	per.	
	The following parameters can be used: 0: No mapping active 7178: Error number of last cause of interruption 2050: Digital inputs/outputs 7200: Temperature power stage 7198: DC bus voltage of power stage supply 7720: Current motor current 7176: Action word		-	
PBMapOut	Mapping of PZD5+6 to drive	-	UINT32	Modbus 6148
-	A parameter with a data length of 32 bits can be used for mapping the PZD5+6 from the master to the drive.	0	UINT32	Profibus 6148
-		0	R/W	
-		65535	per.	
	The following parameters can be used: 0: No mapping active 1556: Acceleration of profile generator 1558: Deceleration of profile generator 1538: Symmetrical ramp		-	

Example of transmit data and response in process data channel: Positioning Relative positioning by 20000usr (0000 4E20_h) is to be carried out, current position=0. The reference speed is to be 1000 min⁻¹ (03E8_h). Start position is 0 usr.

Master <--> Slave						
Triggering positioning	Transmit data	--->	driveCtrl	modeCtrl	ref_16	Ref_32
			02 _h	93 _h	03E8 _h	0000 4E20 _h
Positioning running x_err = 0, x_end = 0	Receive data <---		driveStat	modeStat		32Bit act. pos.
			0006 _h	83 _h		xxxx xxxx _h
	Transmit data	--->	driveCtrl	modeCtrl	ref_16	Ref_32
			02 _h	93 _h	03E8 _h	0000 4 E20 _h
Positioning finished x_err = 0, x_end = 1, x_info = 1	Receive data <---		driveStat	modeStat		32Bit act. pos.
			6006 _h	83 _h		0000 4E20 _h

Table 6.5 Relative positioning

The entry is also made in Big Endian format. The acceleration can only be set via the process data channel if the corresponding parameter has been mapped to PZD5 and PZD6.

019844113724, V2.01, 11.2008

Example of transmit data and response in process data channel: Profile Velocity

The motor is to rotate in Profile Velocity mode at a reference speed of 1000 min⁻¹ (03E8_h). The master must send the following data to the slave:

		Master <---> Slave					
Starting Profile Velocity with 1000 min ⁻¹	Transmit data	--->	driveCtrl 02 _h	modeCtrl 84 _h	ref_16 03E8 _h	ref_32 xxxxxxxx _h	
Motor accelerates	Receive data <---		driveStat 0006 _h	modeStat 84 _h			32Bit act. pos. xxxx xxxx _h
	Transmit data	--->	driveCtrl 02 _h	modeCtrl 84 _h	ref_16 03E8 _h	ref_32 xxxxxxxx	
Reference speed reached x_err = 0, x_end = 0, x_info = 1	Receive data <---		driveStat 2006 _h	modeStat 84 _h			32Bit act. pos. xxxx xxxx _h

Table 6.6 Profile Velocity

The response data contain the current position of the motor and not the speed in "32 bit actual position". The mode bit in the status word indicates whether the set reference speed has been reached.

Receive data format, detailed description

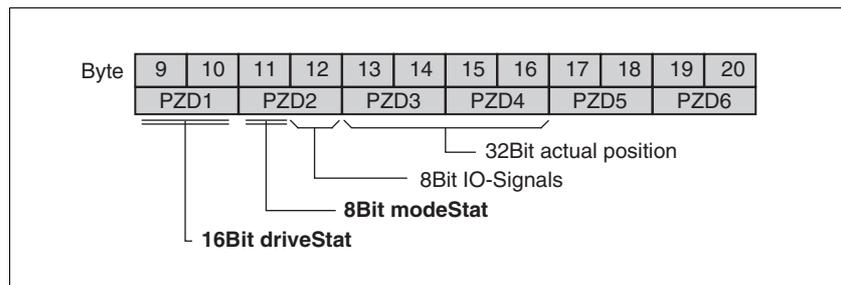


Figure 6.6 Receive data in the process data channel: slave to master

Bytes 9 and 10: driveStat, contains the current operating state, warning bits and error bits and the status of the current operating mode as a field-bus status word.

Byte 11: "modeStat", returns the current operating

Byte 12: "ioSignals", status of the input signals

Bytes 13 ... 16: "32 bit actual position", current position data

Bytes 17...20: these bytes can be parameterized, the content is specified via index and subindex. They do not indicate any time consistency with bytes 9...16.

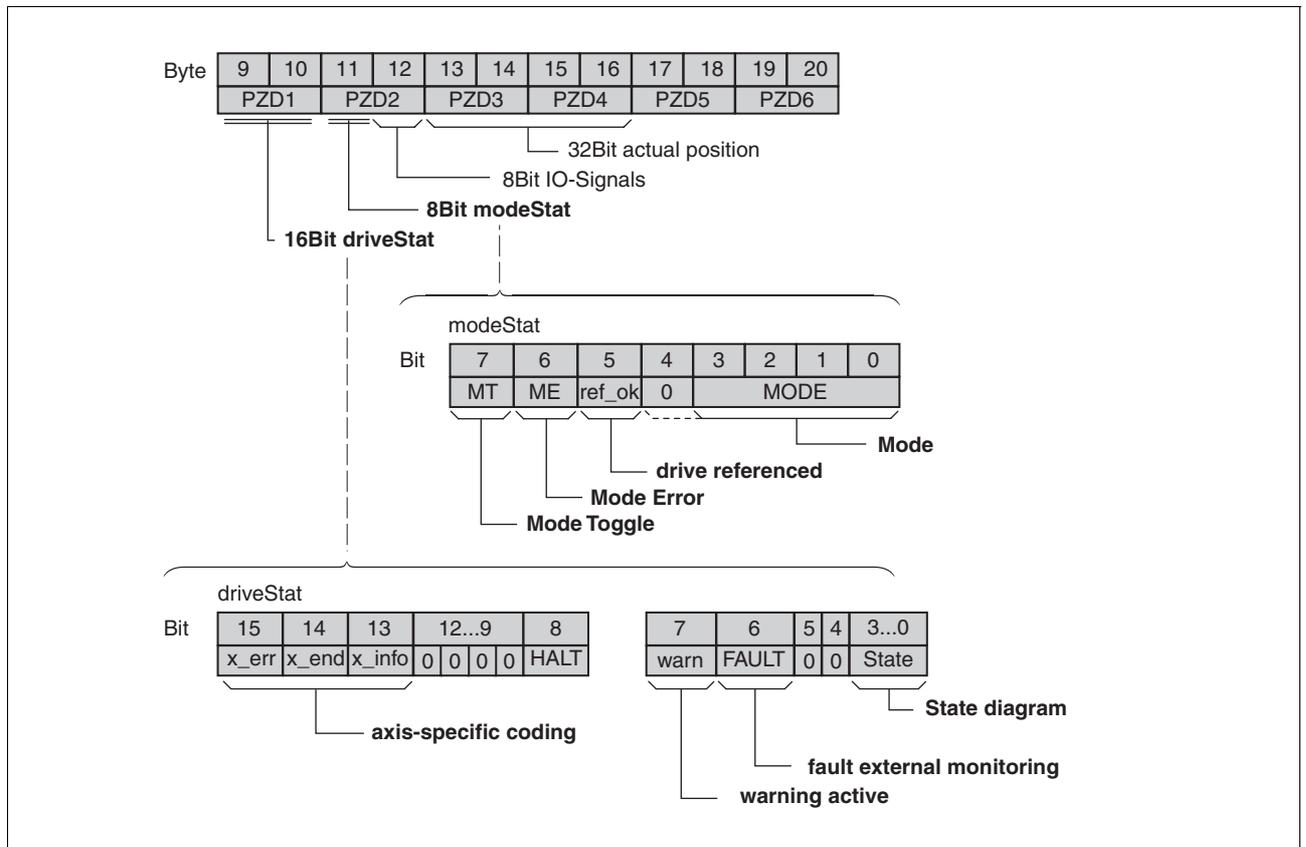


Figure 6.7 Structure of receive data in the process data channel

Description of driveStat The driveStat status word has the following structure

Bit	Name	Description
0 ... 3	State	State diagram, state
6	FAULT	Fault detection, fault occurred
7	warn	A warning was generated
8	HALT	A HALT request is active
13	x_info	Additional information on the operating mode
14	x_end	End identifier for operating mode processing
15	x_err	Operating mode error

Table 6.7 Bit description of driveStat

Description of modeStat modeStat indicates the current processing state of the operating modes:

Bit	Name	Description
0 ... 3	Mode	Currently selected operating mode, see
5	ref_ok	Is set if homing of the product by means of a reference movement or position setting was successful.
6	ME, ModeError	Set if a request of the master via transmit data was rejected.
7	MT, ModeToggle	Mirrored bit 7 (Mode Toggle) of transmit data, this acknowledges acceptance of transmit data. Data may only be evaluated if the MT sent from the master is equal to the MT of the slave.

Table 6.8 Bit description of modeStat

Synchronized processing is possible with the transmit data modeStat, Bit 7 (ModeToggle – MT) and the receive data, Bit 6 and 7 (ModeError – ME and ModeToggle – MT). Synchronized processing means that the master waits for feedback messages from the slave and responds to them.

Description of I/O signals (byte 12)

The current physical states of the signals are represented in a bit-coded way. If the bit is "0", there is no voltage. If the bit is "1", there is voltage.

Bit	Assignment
Bit 0	Reference switch $\overline{\text{REF}}$
Bit 1	Negative limit switch $\overline{\text{LIMN}}$
Bit 2	Positive limit switch $\overline{\text{LIMP}}$
Bit 3	Function $\overline{\text{HALT}}$
Bit 4	Safety function STO $\overline{\text{STO_B}}$ ($\overline{\text{PWRR_B}}$)
Bit 5	Safety function STO $\overline{\text{STO_A}}$ ($\overline{\text{PWRR_A}}$)
Bit 6	Not assigned
Bit 7	Not assigned

Example of receive data in the process data channel

Absolute positioning by 12000usr (0000 2EE0_h) is to be carried out, current position=0. The reference speed is to be 500 min⁻¹ (01F4_h). Start position is 0 usr.

		Master <---> Slave				
Triggering positioning	Transmit data	--->	driveCtrl 02 _h	modeCtrl 83 _h	ref_16 01F4 _h	Ref_32 0000 2EE0 _h
Positioning running x_err = 0, x_end = 0	Receive data	<---	driveStat 0006 _h	modeStat 83 _h		32Bit act. pos. xxxx xxxx _h
	Transmit data	--->	driveCtrl 02 _h	modeCtrl 83 _h	ref_16 01F4 _h	Ref_32 0000 2 EE0 _h
Positioning finished x_err = 0, x_end = 1, x_info = 1	Receive data	<---	driveStat 6006 _h	modeStat 83 _h		32Bit act. pos. 0000 2EE0 _h

Table 6.9 Relative positioning

Special case very short distance positioning

In the case of very short distance positioning, the reference position may already have been reached when the status of the receive data is returned to the master. In this case, the MT (PZD1, modeCtrl, Bit 7) sent from the master is equal to the MT (PZD2, modeStat, Bit 7) returned by the slave, and Bit `x_end = 1` already set. Therefore, `x_end = 0` does not exist for the master. If no error has occurred, the positioning command was correctly executed all the same.

Handling of errors

If the master toggles Bit 7 (MT), this is a request to the slave to start an operating mode or to change data of the current operating mode. If the request cannot be processed, the slave signals this to the master by the following actions:

- In `modeStat` in the receive data, Bit 6 (ModeError) is set.
This bit remains set until MT (`mode-Stat, Bit 7`) (ModeToggle) is toggled again in the receive data. The master can read the corresponding error code by a read access to parameter `ModeError`.

- Continuation of the current operating mode

The active operating mode is not influenced and there is no state transition.

Possible reasons for an unsuccessful operating mode request:

- Reference values outside the value range
- Change of operating mode during processing (impossible)
- Invalid operating mode requested
- The product is not in state 6 (`Operation Enable`) of the state diagram. See the product manual for additional information.

6.5 PLC as a fieldbus master

The fieldbus master provides each connected slave its own memory for transmit data and receive data. Data can be exchanged between PLC memory and fieldbus master via the peripheral equipment range or the process image range.

Fieldbus transmission and the application program's read and write accesses to transmit and receive data is asynchronous. Therefore, it is possible that data is read by the fieldbus master from the PLC memory before the PLC was able to completely update the data.

⚠ CAUTION

INCORRECT CONTROL COMMANDS

If a PLC is used as the master device, the exchange of data can lead to inconsistent transmit data since fieldbus and PLC cycles do not operate synchronously.

- Observe all notes concerning operation with a PLC.

Failure to follow these instructions can result in injury or equipment damage.

The following must be observed for operation of a master PLC:

- Copy data from high to low addresses.
- Toggle MT (`modeCtrl`, Bit 7) last
- During data exchange via the process image, the transmit data must be copied from the memory for the process image to the memory of the fieldbus master. This copy process must not create inconsistent data on the fieldbus.

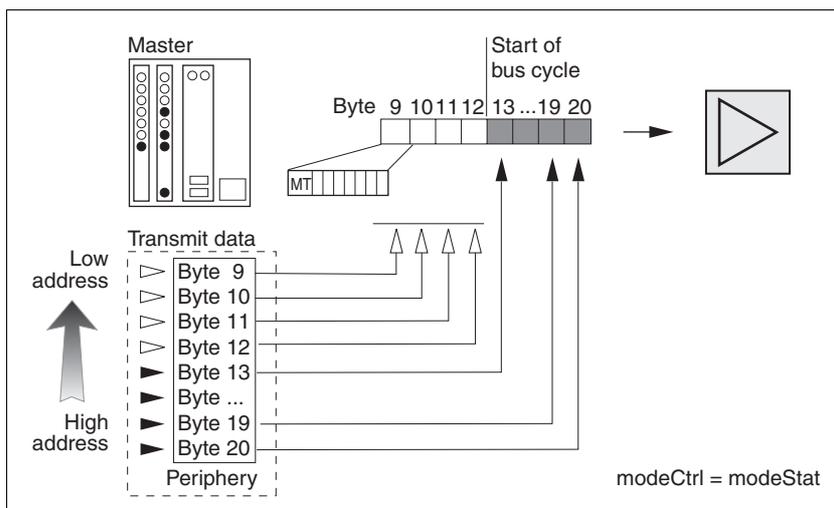


Figure 6.8 Data consistency, byte 9 (Bit 7) is copied last

Data exchange via the peripheral memory

In the case of data exchange via the peripheral memory, the data is consistent if MT (modeCtrl, Bit 7) is entered last. The controller ignores the transmitted data as long as this bit is equal to the MT (modeStat, Bit 7).

The following example shows the problems of a lack of data consistency:

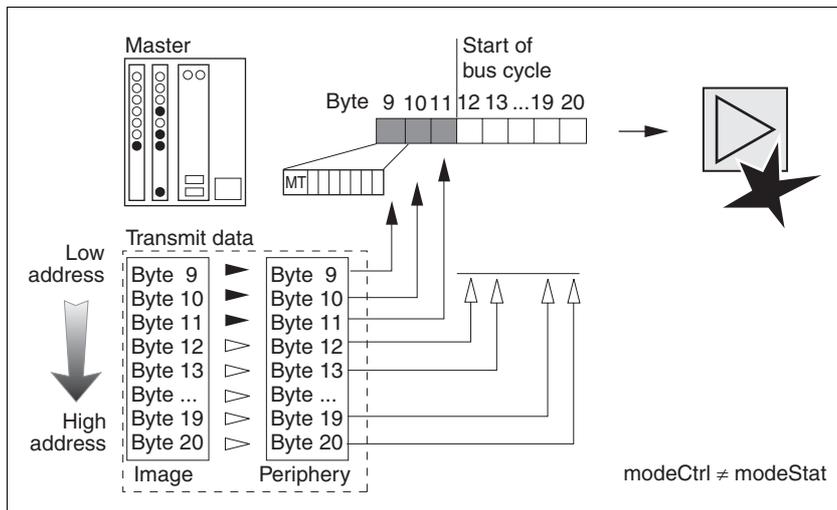


Figure 6.9 Lack of data consistency: undefined movement!!

Data exchange via process image memory

Data consistency during data exchange via the process image memory can only be achieved if there is no bus access to the data in the peripheral memory during the copy process between image and peripheral memory in the direction from a low to a high address.

Inconsistent data is generated if bit 7 has already been transmitted from byte 9 (MT) via the bus before the slave has received the remaining correct data. As soon as MT is transmitted, the slave detects the state transition when checking the bit and interprets this as a new command, which is executed immediately.

6.5.1 Response in the case of special master commands

Behavior with "Clear" command of master The response to the Global Control Command "Clear" from a PLC as a Profibus master can be parameterized. The error response to a "Clear" command may be parameterized as a class 0 or a class 1 error.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
PBSafeState	Safe state	-	UINT16	Modbus 6154
-	0 / NoError: No response	0	UINT16	Profibus 6154
-	1 / ErrorClass2: Error of class 2, drive switches to FAULT if the power stage was active	1	R/W per.	-
	Response of the drive in state 'Clear' of the ProfibusDP master and response to termination of the watchdog.			

Response to incorrect processing of the process data channel The error class for errors that occur during processing of the process data channel can be parameterized. The error response may be parameterized as a class 0 or a class 1 error.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
PBFItPpo	Error response to process data channel processing fault	-	UINT16	Modbus 6158
-	0 / none: No error response (Mode Error)	0	UINT16	Profibus 6158
-	1 / ErrorClass1: Error class 1	1	R/W per.	-

7 Examples

7.1 Overview of examples

The programming examples show hands-on applications for network operation. There are two access methods via the Profibus fieldbus: via the parameter channel and via the process data channel.

Use of the parameter channel

An access is a write or read access to a single parameter. The available parameters are described in the product manual. This chapter describes the use of the parameter channel on the basis of just a small number of parameters since this type of communication can be used with all available parameters.

Use of the process data channel

The process data channel is recommended for positioning because the information is transmitted much more effectively. The chapter provides various hands-on examples of the application of the protocols supported by the product and describes the general procedures

Structure of the examples

The examples show the following:

- Task
- Initial conditions
- Required commands in the transmit telegram
- Response in the receive telegram
- Possible restrictions for command execution

You should be familiar with the following to be able to understand the examples:

- Operating concept and functionality of the drive system
For more information see the product manual
- Fieldbus protocol and connection to the master
- Functionality of the fieldbus profile.

Product manual

The examples supplement the descriptions of the function principles and the operating modes in the product manual. All parameters for the operating modes and functions as well as the number format of the parameter values are listed in the product manual.

7.2 Use of the parameter channel

7.2.1 Writing parameters

Task The parameter `RAMPacc`, 1556:00 (acceleration) is to be set to the value 10000.

This requires the index and subindex to be converted into hexadecimal notation:

- Index: 1556 = 06 14_h
- Subindex: 00 = 00_h
- Value: 10000 = 00002710_h

The value 30_h must be entered as a PKE (parameter identifier) since the parameter has a 32 bit data type.

Transmit data

Parameters	PKE, 1st byte request identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Tx 0614 _h :00 _h RAMPacc	30 _h	00 _h	0614 _h	0000 2710 _h	Sets the acceleration to 10000 min ⁻¹ *s = 2710 _h as a 32 bit value

Refer to the corresponding column in the parameter description of the product manual for the data type of the value to be written. The Profibus protocol used transmits 16 bit values and 32 bit values in the format "lowest value byte first – highest value byte last". When an INT16 or a UINT16 value is transmitted, the PKE corresponding to the data type must be included. The value must be stored in the last two data bytes, the first two data bytes must be zero (0).

Receive data

Parameters	PKE, 1st byte response identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Rx 0614 _h :00 _h RAMPacc	20 _h	00 _h	0614 _h	xxxx xxxx	The response data has no significance, the positive acknowledgement is signaled by PKE=20.

7.2.2 Reading a parameter

Task The parameter `_n_act`, 7696:0 (actual speed) is to be read. Index and subindex must be converted to hexadecimal format for this purpose:

- Index: 7696 = 1E10_h
- Subindex: 0 = 00_h

The value 10_h must be entered as PKE. This value identifies a Read Request.

Transmit data

Parameters	PKE, 1st byte request identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Tx 1E10 _h :00 _h _n_act	10 _h	00 _h	1E10 _h	xxxx xxxx	Reads the actual speed. The data has no significance.

The 4 data bytes have no significance for a read request.

Receive data

Parameters	PKE, 1st byte response identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Rx 1E10 _h :00 _h _n_act	20 _h	00 _h	1E10 _h	0000 03E8h	The data 000003E8 correspond to 1000 min ⁻¹ ; PKE=20 signals successful execution.

A distinction is made between parameter values with 32 bit data and parameter values with 16 bit data (described in the product manual as INT32 or UINT32 and INT16 or UINT16 data types), based on the response identifier (2 or 1). In the case of 16 bit data, it is important to evaluate only the last two data bytes and to ignore the first two data bytes.

7.2.3 Synchronous errors

Receive data with error telegram (error response)

If a write or read command is unsuccessful, the product responds with an error telegram (error response). The transmitted error number provides information on the exact cause.

Parameters	PKE, 1st byte response identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Rx 0101 _h :00 _h	70 _h	00 _h	0065 _h	0000 B30A _h	Error number 0000B30A _h means: parameter does not exist

The example shows the response to a write or read request for a non-existent parameter 0101:00.

The table of error numbers can be found in the product manual, chapter Diagnostics.

7.3 Operating states in the process data channel

▲ WARNING

UNINTENDED OPERATION

- Note that any changes to the values of these parameters are executed by the drive controller immediately on receipt of the data set.
- Verify that the system is free and ready for movement before changing these parameters.

Failure to follow these instructions can result in death, serious injury or equipment damage.

The product operates in different operating states. The individual operating states are numbered from 1 to 9. The operating states and the transition conditions are described in the product manual.

Operating state	Name	Power stage	Description
3	Switch On disabled	off	Activation of power stage disabled, motor without current
4	Ready To Switch On	off	Power stage is ready to switch on, motor without current
6	Operation Enable	on	Active operating state, current available to motor
7	Quick Stop active	on	Fault state, power stage remains enabled
9	Fault	off	Fault state, power stage is disabled

Table 7.1 Operating states

Requests for switching operating state are transmitted by the master via the process data channel PZD1 in the field `driveCtrl`. The slave provides feedback on the current operating state to the master via the process data channel PZD1, field `driveStat`.

Table 7.2, Seite 7-43 shows the bit assignment of the field `driveCtrl` in the transmit data in the process data channel (byte 9):

Bit no.	Value	Meaning
0	01 _h	Disable, Operation enable ---> Ready to switch on
1	02 _h	Enable, Ready to switch on ---> Operation Enable
2	04 _h	Quick Stop, Operation enable ---> Quick Stop active
3	08 _h	Fault Reset, Fault ---> Ready to switch on
4	10 _h	0, reserved
5	20 _h	SetHALT
6	40 _h	ClearHALT
7	80 _h	Continue, resume operating mode interrupted by HALT

Table 7.2 Transmit data byte 9, driveCtrl, bit assignment

7.3.1 Enabling and disabling the power stage

The power stage is enabled by the transition from operating state 4 to 6 . Byte 9 transmit data, `driveCtrl`, provides the two bits `Enable` and `Disable`. One of them must be 1, the other 0.

Enable power stage Prerequisite: the product in in operating state 4.
To enable on the power stage, a 1>0 edge must be generated in `driveCtrl`, bit 1 (Enable). This can be done by deleting bit 0 (Disable) and setting bit 1 . The master then waits until the product signals operating state 6 .

Example:

Master <---> Slave			
Disable is requested	Transmit data	--->	<code>driveCtrl 01_h</code>
Product signals operating state 4	Receive data	<---	<code>driveStat xxx4_h</code>
Request Enable	Transmit data	--->	<code>driveCtrl 02_h</code>
Product signals operating state 5	Receive data	<---	<code>driveStat xxx5_h</code>
Request Enable	Transmit data	--->	<code>driveCtrl 02_h</code>
Product signals operating state 6	Receive data	<---	<code>driveStat xxx6_h</code>

Table 7.3 Enable power stage

Disable power stage Prerequisite: Product is in operating state 6 or 7.
To disable the power stage, a 0>1 edge must be generated in `driveCtrl`, bit 1 (Disable). This can be done by setting bit 0 and deleting bit 1 (Enable). The product switches to operating state 4.

Example:

Master <---> Slave			
Enable is requested	Transmit data	--->	<code>driveCtrl 02_h</code>
Product signals operating state 6	Receive data	<---	<code>driveStat xxx6_h</code>
Request disable	Transmit data	--->	<code>driveCtrl 01_h</code>
Product signals operating state 4	Receive data	<---	<code>driveStat xxx4_h</code>

Table 7.4 Disable power stage

7.3.2 Triggering a Quick Stop

A running motion command can be interrupted via the fieldbus at any time with the `QuickStop` command. It is triggered by a change from 0 to 1 in `driveCtrl`, Bit 2. After the transition to operating state 7 (Quick Stop), the product decelerates with the specified EMERGENCY STOP ramp until it comes to a standstill.

In order to start a new motion command, you must first set the product to operating state 6. To do so, cause a 0>1 edge in `driveCtrl`, bit 3, Fault Reset.

Example:

		Master <---> Slave	
Enable is requested	Transmit data	--->	<code>driveCtrl 02_h</code>
Product signals operating state 6	Receive data	<---	<code>driveStat xxx6_h</code>
Request Quick Stop and Enable	Transmit data	--->	<code>driveCtrl 06_h</code>
Product signals operating state 7	Receive data	<---	<code>driveStat xxx7_h</code>
Wait until the product has come to a standstill and the system is to resume operation			
Product signals operating state 7	Receive data	<---	<code>driveStat xxx7_h</code>
Clearing a Quick Stop request, performing a Fault Reset	Transmit data	--->	<code>driveCtrl 0A_h</code>
Product signals operating state 6	Receive data	<---	<code>driveStat xxx6_h</code>
Clearing Fault Reset	Transmit data	--->	<code>driveCtrl 02_h</code>
Product signals operating state 6	Receive data	<---	<code>driveStat xxx6_h</code>

Table 7.5 Triggering a Quick Stop

7.3.3 Resetting faults

If an error occurs during operation, the product switches to operating state 7 (Quick Stop) or operating state 9 (Fault), depending on the type of error.

After having remedied the cause of the fault, you can reset the fault state with a Fault Reset (0>1 edge in `driveCtrl`, bit 3).

If the operating state was 7, there is a transition to operating state 6 after the Fault Reset.

If the operating state was 9, there is a transition to operating state 4 after the Fault Reset. Then a 0>1 edge in `driveCtrl`, bit 1 (Enable) must be transmitted in order to enable on the power stage again.

Example:

In this example, the master first causes a change from 1 to 0 in Bit 1 (Enable) during the Fault Reset and then a change from 0 to 1. This causes a transition back to operating state 6.

		Master <---> Slave	
Request Enable	Transmit data	--->	driveCtrl 02 _h
Slave signals operating state 9 (Fault)	Receive data	<---	driveStat xxx9 _h
Remedy cause of error			
Request Fault Reset	Transmit data	--->	driveCtrl 08 _h
Slave signals operating state 4	Receive data	<---	driveStat xxx4 _h
Request Enable	Transmit data	--->	driveCtrl 02 _h
Slave signals operating state 5	Receive data	<---	driveStat xxx5 _h
Request Enable	Transmit data	--->	driveCtrl 02 _h
Slave signals operating state 6	Receive data	<---	driveStat xxx6 _h

Table 7.6 Resetting faults

7.4 Operating modes in the process data channel

Transmit data With the transmit data, you can start motion commands and change them while they are being processed.

The process data provides the following fields for this purpose:

- PZD1: modeCtrl, starting and changing operating modes
- PZD2: ref_16, e.g. reference speed, depends on operating mode
- PZD3 and 4: ref_32, e.g. reference position, depending on operating mode
- PZD5 and 6: mapped value

The values entered for these fields are not taken over by the product until modeCtrl, bit 7 (Toggle Mode) changes.

Proceed as follows to assign values to the product:

- ▶ Enter the desired operating mode and the corresponding values in the fields modeCtrl, PZD 2...6.
- ▶ "Toggle" modeCtrl, bit 7 (ModeToggle)

This avoids consistency problems in the transmit data.

- Receive data* Motion commands are monitored by means of the receive data in the process data channel.
- The process data channel provides the following fields for this purpose:
- PZD1: `modeStat`, for Handshake purposes
 - PZD2: `driveStat`, signals the motion status and errors as well as I/O signals
 - PZD3 and PZD4: "32Bit actual position", actual position
 - PZD5 and PZD6: can be parameterized (mapped), but there is no time consistency with PZD1...4 apart from exceptions.
- Mapping* Parameter values can be mapped to PZD5 and PZD6 for transmit data as well as receive data, see chapter 6.4.2 "Structure of the process data channel", page 26 onwards.
- Mode Toggle* Bit `Mode-Toggle` is available both in the transmit data telegram and the receive data telegram. The master provides this bit in the transmit data telegram and the product mirrors it in the receive data telegram. This procedure allows the master to detect whether the data transmitted by the slave is current.
- Example* The master starts a positioning movement that will only take a very short time. The master waits for the end of the positioning movement by checking the receive data telegram for bit `x_end = 1` (end of positioning).
- The master may receive data from the slave that still originate from a point in time before the positioning movement was started. This data also contains `x_end = 1`. The master now detects that the data is obsolete because the included bit `ModeToggle` does not match that of its motion command.
- The master only evaluates data in which the received `ModeToggle` bit is identical to the last bit transmitted by the master.
- Acceleration* Prior to a positioning movement, you can first set the desired acceleration by mapping the acceleration to PZD5 and PZD6 or via the parameter channel (parameter `RAMPacc, 1556:00`). Note that the acceleration can only be changed when the product is at a standstill.
- Assumptions* The examples are based on the following assumptions:
- Operating state 6 (Operation Enable)
 - No homing done yet (bit `ref_ok = 0`)
 - `p_act = 0` (actual position of motor)
 - Transmit data PZD1: `modeCtrl`, bit 7 = 0 (`ModeToggle`)

7.4.1 Absolute positioning

⚠ WARNING**UNINTENDED OPERATION**

- Note that any changes to the values of these parameters are executed by the drive controller immediately on receipt of the data set.
- Verify that the system is free and ready for movement before changing these parameters.

Failure to follow these instructions can result in death, serious injury or equipment damage.

To start an absolute positioning movement, the following settings must be made in the transmit data telegram:

- ▶ Enter the reference speed in PZD2 ref_16 and the target position in PZD3 and PZD4 ref_32 .
- ▶ Enter operating mode 03_h (Profile Position operating mode, absolute positioning) in the field modeCtrl.
- ▶ "Toggle" modeCtrl, bit 7, so the data is taken over by the product.

Example 1: Absolute positioning to position 100,000 (0001 86A0_h) at a reference speed of 1000 min⁻¹ (03E8_h)

Master <--> Slave						
Triggering positioning	Transmit data	--->	driveCtrl 02 _h	modeCtrl 83 _h	ref_16 03E8 _h	Ref_32 0001 86A0 _h
Positioning running x_err = 0, x_end = 0	Receive data	<---	driveStat 0006 _h	modeStat 83 _h		32_Bit act. pos. xxxx xxxx _h
Triggering positioning	Transmit data	--->	driveCtrl 02 _h	modeCtrl 83 _h	ref_16 03E8 _h	Ref_32 0001 86A0 _h
Positioning finished x_err = 0, x_end = 1, x_info = 1	Receive data	<---	driveStat 6006 _h	modeStat 83 _h		32Bit act. pos. 0001 86A0 _h

Table 7.7 Absolute positioning at constant reference speed

The data telegram "Positioning Running" can be transmitted several times; the current actual position is contained in the field "32Bit actual position".

Example 2: As example 1, but the reference speed is changed to 2000 min⁻¹ (07D0_h) during the movement.

Master <---> Slave						
Triggering positioning	Transmit data	--->	driveCtrl 02 _h	modeCtrl 83 _h	ref_16 03E8 _h	Ref_32 0001 86A0 _h
Positioning running x_err = 0, x_end = 0	Receive data	<---	driveStat 0006 _h	modeStat 83 _h		32Bit act. pos. xxxx xxxx _h
Changing reference speed	Transmit data	--->	driveCtrl 02 _h	modeCtrl 03 _h	ref_16 07D0 _h	Ref_32 0001 86A0 _h
Positioning running x_err = 0, x_end = 0	Receive data	<---	driveStat 0006 _h	modeStat 03 _h		32Bit act. pos. xxxx xxxx _h
Changing reference speed	Transmit data	--->	driveCtrl 02 _h	modeCtrl 03 _h	ref_16 07D0 _h	Ref_32 0001 86A0 _h
Positioning complete x_err=0, x_end = 1, x_info = 1	Receive data	<---	driveStat 6006 _h	modeStat 03 _h		32Bit act. pos. 0001 86A0 _h

Table 7.8 Absolute positioning with change of reference speed

The ata telegram "Positioning Running" can be transmitted several times. The actual position is contained in the field "32Bit actual position". When the reference speed is changed, the same target position is sent because it does not change in this example.

7.4.2 Relative positioning

⚠ WARNING

UNINTENDED OPERATION

- Note that any changes to the values of these parameters are executed by the drive controller immediately on receipt of the data set.
- Verify that the system is free and ready for movement before changing these parameters.

Failure to follow these instructions can result in death, serious injury or equipment damage.

Relative positioning is similar to absolute positioning. You only need to enter the value 13_h (operating mode Profile Positioning, relative positioning) in field `modeCtrl`. Also note that several target positions transmitted in succession are added up.

Example: Relative positioning by 100,000 (000186A0_h) increments at a speed of 1000 min⁻¹ (03E8_h).
During the movement, the speed is to be changed to 2000 min⁻¹ (07D0_h).

Master <--> Slave						
Triggering positioning	Transmit data	--->	driveCtrl	modeCtrl	ref_16	Ref_32
			02 _h	93 _h	03E8 _h	0001 86A0 _h
Positioning running: x_err = 0, x_end = 0	Receive data	<---	driveStat	modeStat		32Bit act. pos.
			0006 _h	83 _h		xxxx xxxx _h
Change reference speed Transmit relative position 0	Transmit data	--->	driveCtrl	modeCtrl	ref_16	Ref_32
			02 _h	13 _h	07D0 _h	0000 0000 _h
Positioning running, x_err = 0, x_end = 0	Receive data	<---	driveStat	modeStat		32Bit act. pos.
			0006 _h	03 _h		xxxx xxxx _h
Change reference speed Transmit relative position 0	Transmit data	--->	driveCtrl	modeCtrl	ref_16	Ref_32
			02 _h	13 _h	07D0 _h	0000 0000 _h
Positioning complete x_err = 0, x_end = 1, x_info = 1	Receive data	<---	driveStat	modeStat		32Bit act. pos.
			6006 _h	03 _h		xxxx xxxx _h

Table 7.9 Profile Position operating mode, relative positioning with change of reference speed

The data telegram "Positioning Running" can be transmitted several times; the actual position is contained in the field "32Bit actual position". When the reference speed is changed, the value zero (0) must be sent as the new target position because the new value is added to the previously calculated target position.

7.4.3 Profile Velocity

⚠ WARNING

UNINTENDED OPERATION

- Note that any changes to the values of these parameters are executed by the drive controller immediately on receipt of the data set.
- Verify that the system is free and ready for movement before changing these parameters.

Failure to follow these instructions can result in death, serious injury or equipment damage.

In Profile Velocity operating mode, a reference speed for the motor is set and a movement without a target position is started.

The following settings are required in the transmit data telegram to start a movement in Profile Velocity operating mode or to change the reference speed if the operating mode is already running:

- ▶ In PZD2, ref_16 enter the reference speed (ref_32 is of no significance in this case)
- ▶ Enter the operating mode 04_h (Profile Velocity) in modeCtrl.
- ▶ Toggle modeCtrl, bit 7, so the data is taken over by the product.

Example The operating mode Profile Velocity is started with a reference speed of 1000 min^{-1} ($03E8_h$) (ref_{16}).
The reference speed is changed to 2000 min^{-1} ($07D0_h$) during the movement.
The Profile Velocity operating mode is terminated when the reference speed 0 is transmitted; standstill is waited for.

Master <---> Slave						
Starting Profile Velocity with 1000 min^{-1}	Transmit data	--->	driveCtrl 02_h	modeCtrl 84_h	ref_16 $03E8_h$	Ref_32 xxxxxxx _h
Product accelerates $xerr=0$, $xend=0$, $xinfo=0$	Receive data	<---	driveStat 0006_h	modeStat 84_h		32Bit act. pos. xxxx xxx _h
Profile velocity with 1000 min^{-1}	Transmit data	--->	driveCtrl 02_h	modeCtrl 84_h	ref_16 $03E8_h$	Ref_32 xxxx xxx _h
Reference speed reached $xerr=0$, $xend=0$, $xinfo=1$	Receive data	<---	driveStat 2006_h	modeStat 84_h		32Bit act. pos. xxxx xxx _h
Change speed to Change 2000 min^{-1}	Transmit data	--->	driveCtrl 02_h	modeCtrl 04_h	ref_16 $07D0_h$	Ref_32 xxxx xxx _h
Product accelerates $xerr=0$, $xend=0$, $xinfo=0$	Receive data	<---	driveStat 0006_h	modeStat 04_h		32Bit act. pos. xxxx xxx _h
Change speed to 2000 min^{-1}	Transmit data	--->	driveCtrl 02_h	modeCtrl 04_h	ref_16 $07D0_h$	Ref_32 xxxx xxx _h
Reference speed reached $xerr=0$, $xend=0$, $xinfo=1$	Receive data	<---	driveStat 2006_h	modeStat 04_h		32Bit act. pos. xxxx xxx _h
Change speed to 0 min^{-1}	Transmit data	--->	driveCtrl 02_h	modeCtrl 84_h	ref_16 0000_h	Ref_32 xxxx xxx _h
Product decelerates $xerr=0$, $xend=0$, $xinfo=0$	Receive data	<---	driveStat 0006_h	modeStat 84_h		32Bit act. pos. xxxx xxx _h
Change speed to Change 0 min^{-1}	Transmit data	--->	driveCtrl 02_h	modeCtrl 84_h	ref_16 0000_h	Ref_32 xxxx xxx _h
Profile velocity terminated $xerr=0$, $xend=1$, $xinfo=1$	Receive data	<---	driveStat 6006_h	modeStat 84_h		32Bit act. pos. xxxx xxx _h

Table 7.10 Profile Velocity

Field "32-bit actual position" of the receive data telegram contains the current position in increments.

7.4.4 Homing by position setting

During position setting, a new position is assigned to the current motor position. This only shifts the coordinate system, the motor itself does not move.

You must make the following settings for position setting in the transmit data telegram:

- Enter the new position in ref_{32} . (PZD2 (ref_{16} is of no significance here))
- Enter operating mode 02_h in modeCtrl (Homing, Position Setting).
- Toggle modeCtrl , bit 7, so the data is taken over by the slave.

Example: The motor is at position -100000 (FFFE7960_h) (ref_32).
Position 200000 is assigned to the motor (00030D40_h).

Master <---> Slave					
Product signals Position 100000	Receive data <---	driveStat xxxx _h	modeStat xx _h		32Bit act. pos. FFFE 7960 _h
Position setting to 200000	Transmit data	driveCtrl 02 _h	modeCtrl 82 _h	ref_16 xxxx _h	Ref_32 0003 0D40 _h
Position accepted x_err = 0, x_end = 1, x_info = 0	Receive data <---	driveStat 4006 _h	modeStat A2 _h		32Bit act. pos. 0003 0D40 _h

Table 7.11 Position setting

7.4.5 Reference movement

⚠ WARNING

UNINTENDED OPERATION

- Note that any changes to the values of these parameters are executed by the drive controller immediately on receipt of the data set.
- Verify that the system is free and ready for movement before changing these parameters.

Failure to follow these instructions can result in death, serious injury or equipment damage.

During the reference movement a limit switch or reference switch is approached and then a new value is assigned to this position.

Before a reference movement is started, the parameters must be set via the parameter channel to satisfy the requirements. See the product manual for detailed information on parameterization and on performing a reference movement.

To start a reference movement the following settings must be made in the transmit data telegram:

- Enter the type of reference movement in PZD2 (ref_16) (PZD3 und PZD4 (ref_32 is of no significance here).

The available types of reference movement are described in the product manual.

- Enter operating mode 12_h (Homing, reference movement) in modeCtrl.
- Toggle modeCtrl, bit 7, so the data is taken over by the slave.

Example A reference movement to the negative limit switch (LIMN) is to be performed; this is reference movement type 2.

Master <---> Slave						
Trigger reference movement	Transmit data	--->	driveCtrl 02 _h	modeCtrl 92 _h	ref_16 0002 _h	Ref_32 xxxx xxxx _h
Reference movement running xerr=0, xend=0	Receive data	<---	driveStat 0006 _h	modeStat80 2 _h		32Bit act. pos. xxxx xxxx _h
Reference movement	Transmit data	--->	driveCtrl 02 _h	modeCtrl 92 _h	ref_16 0002 _h	ref32 xxxx xxxx _h
Reference movement complete, xerr=0, xend=1	Receive data	<---	driveStat 4006 _h	modeStat A2 _h		32_Bit act. pos. 0000 0000 _h

Table 7.12 Reference movement

7.5 Error signaling in the process data channel

7.5.1 Synchronous errors

If a request for an operating mode sent via the transmit data telegram cannot be processed, the slave rejects processing and sets `modeStat`, bit 6 (ModeError) in the receive data telegram. This does not interrupt the current process. To determine the cause of the error, the master can read the error number from the parameter `ModeError`, 6962:00by means of an access via the parameter channel. See the product manual for a list of the error numbers.

The error indication is reset when the next valid data telegram is transmitted.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
ModeError	Error code for synchronous errors (ME flag)	-	UINT16	Modbus 6962
-	Manufacturer-specific error code that caused the ModeError flag to be set.	0	UINT16	Profibus 6962
-	Usually, this is an error that was caused by the activation of an operating mode.	-	R/-	-

Example The product is in Profile Velocity operating mode. An attempt is made to perform position setting.

Master <---> Slave					
Profile Velocity, x_end = 0	Receive data <---	driveStat 0006 _h	modeStat 04 _h		32Bit act. pos. xxxx xxxx _h
Request: position setting to 0	Transmit data --->	driveCtrl 02 _h	modeCtrl 82 _h	ref_16 xxxx _h	Ref_32 0000 0000 _h
Request rejected, ModeError = 1	Receive data <---	driveStat 0006 _h	modeStat C4 _h		32Bit act. pos. xxxx xxxx _h

Table 7.13 Synchronous error, invalid operating mode request

NOTE: If a request for position setting is rejected, the operating mode Profile Velocity remains active.

7.5.2 Asynchronous errors

Asynchronous errors are triggered by internal monitoring (e.g. temperature) or by external monitoring (e.g. limit switch). If an asynchronous error occurs, the product responds by decelerating or by disabling off the power stage.

Asynchronous errors are indicated in the following way:

- Transition to operating state 7 (Quick Stop) or operating state 9 (Fault).
The transition is indicated in the receive data telegram `driveStat`, bits 0 ... 3.
- Setting of `driveStat`, bit 6 (Error) or `driveStat`, Bit 7 (Warning) as well as bit 15, `x_err`, fault state during processing
- In addition, an error number is assigned to each error. In the event of an asynchronous error, the corresponding error number can be read from the parameter `_StopFault` (7178:00).

Example: External monitoring triggers an error message: positive limit switch LIMP was hit.

Master <---> Slave						
Triggering positioning	Transmit data	--->	<code>driveCtrl</code> 02 _h	<code>modeCtrl</code> 03 _h	<code>ref_16</code> , vel 03E8 _h	<code>Ref_32</code> 0FFF 8765 _h
Positioning running <code>xerr=0</code> , <code>xend=0</code>	Receive data	<---	<code>driveStat</code> 0006 _h	<code>modeStat</code> 03 _h		32Bit act. pos. xxxx xxxx _h
Positioning	Transmit data	--->	<code>driveCtrl</code> 02 _h	<code>modeCtrl</code> 03 _h	<code>ref_16</code> , vel 03E8 _h	<code>Ref_32</code> 0FFF 8765 _h
Limit switch detected <code>xerr=1</code> , <code>xend=0</code>	Receive data	<---	<code>driveStat</code> 8047 _h	<code>modeStat</code> 03 _h		32Bit act. pos. xxxx xxxx _h
Positioning	Transmit data	--->	<code>driveCtrl</code> 02 _h	<code>modeCtrl</code> 03 _h	<code>ref_16</code> , vel 03E8 _h	<code>Ref_32</code> 0FFF 8765 _h
Motor stopped <code>xerr=1</code> , <code>xend=1</code>	Receive data	<---	<code>driveStat</code> C047 _h	<code>modeStat</code> 03 _h		32Bit act. pos. xxxx xxxx _h

Table 7.14 Asynchronous error

When the limit switch is detected, the motor is decelerated with the EMERGENCY STOP ramp until it comes to a standstill and the bit `x_err` is set. After the motor has come to a standstill, bit `x_end` is set.

8 Diagnostics and troubleshooting

8.1 Fieldbus communication error diagnostics

A properly functioning fieldbus operation is essential for evaluating operational and error messages.

Checking connections

If the product cannot be addressed via the fieldbus, first check the connections.

Check the following connections:

- ▶ System power supply
- ▶ Supply connections
- ▶ Fieldbus cables and wiring
- ▶ Fieldbus connection



If the internal terminating resistor is activated via the switch (terminated), output A2/B2 and any bus devices that may be connected to the output are automatically disconnected from the fieldbus.

LEDs for Profibus

The fieldbus status can be checked using the two LEDs at the HMI:

LED "RUN"	LED "ERR"	Meaning
off	off	Fieldbus communication inactive
lights	off	Fieldbus communication active
off	lights	Fieldbus error (e.g. watchdog)
off	flashes	Missing or incorrect parameterization

Fieldbus function test

If the connections are correct, check the settings for the fieldbus addresses. After correct configuration of the transmission data, test fieldbus mode.

In addition to the master that knows the product via the GSD file and addressing, a bus monitor should be installed that, as a passive device, displays messages.

- ▶ Switch the supply voltage of the drive system off and on.
- ▶ Observe the network messages shortly after switching on the drive system. A bus monitor can be used to record the elapsed time between telegrams and the relevant information in the telegram.

Possible errors: addressing, parameterization, configuration

If it is impossible to connect to a device, check the following:

- Addressing: All network devices must have an address between 1 and 126. Each network device must have a unique address.
- Parameterization: The parameterized Ident number and the user parameters must match the values stored in the GSD file.
- Configuration: The data length in input and output direction must be identical to the length specified in the GSD file.

8.2 Error messages

Error messages generated when the network is in operation are received by the master via the fieldbus.

The following error messages are possible:

- Synchronous errors
- Asynchronous errors
- Errors during operating mode control via process data channel.

8.2.1 Synchronous errors

If a command cannot be processed in the parameter channel, the master receives a synchronous error message from the slave.

If a request for an operating mode sent via the transmit data telegram cannot be processed, the slave rejects processing and sets `modeStat`, bit 6 (ModeError) in the receive data telegram. This does not interrupt the current process. To determine the cause of the error, the master can read the error number from the parameter `ModeError`, 6962:00 by means of an access via the parameter channel. See the product manual for a list of the error numbers.

The error indication is reset when the next valid data telegram is transmitted.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
ModeError	Error code for synchronous errors (ME flag)	-	UINT16	Modbus 6962
-	Manufacturer-specific error code that caused the ModeError flag to be set.	0	UINT16	Profibus 6962
-	Usually, this is an error that was caused by the activation of an operating mode.	-	-	-

Error message in parameter channel

The error message is generated as a response to a parameter transmission error. The cause of the error is output in the PWE as Error Code in bytes 5...8.

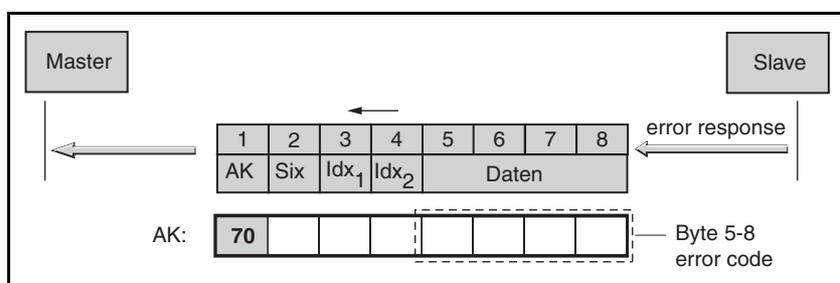


Figure 8.1 Error message in parameter channel

Causes of a synchronous error Possible causes of a synchronous error are:

- Error during execution of an action command or control command
- Parameter value outside the permissible value range
- Invalid action command or control command during processing
- Access to unknown parameter (index/subindex)

8.2.2 Asynchronous errors

Asynchronous errors are triggered by internal monitoring (e.g. temperature) or by external monitoring (e.g. limit switch). If an asynchronous error occurs, the product responds by decelerating or by disabling the power stage.

Asynchronous errors are indicated in the following way:

- Transition to operating state 7 (Quick Stop) or operating state 9 (Fault). The transition is indicated in the receive data telegram in `driveStat`, bits 0 ... 3.
- Setting of `driveStat`, bit 6 (error) or `driveStat`, bit 7 (warning) as well as bit 15, `x_err`, fault state during processing

The error bits have the following meaning:

- Bit 6
Error message of an (e.g. interruption of movement by limit switch). The exact cause is contained in parameter `_StopFault`, 7178:00 in a bit-coded way.
- Bit 7
Warning (e.g. overtemperature warning)
The error information is contained in parameter `FLT_err_num`, 15362:00 in a bit-coded way.

The last cause of interruption is entered in parameter `_StopFault`, 7178:00 as error number.

For a list of error numbers and their meanings see the chapter "Diagnostics and Troubleshooting" in the product manual.

For more information on parameters, error classes and troubleshooting see the chapter "Diagnostics and Troubleshooting" in the product manual.

8.2.3 Errors during operating mode control

Motion commands can be triggered and modified via the process data channel. If the request cannot be processed, an error bit is set in the receive data.

9 Glossary

9.1 Units and conversion tables

The value in the specified unit (left column) is calculated for the desired unit (top row) with the formula (in the field).

Example: conversion of 5 meters [m] to yards [yd]
 $5 \text{ m} / 0.9144 = 5.468 \text{ yd}$

9.1.1 Length

	in	ft	yd	m	cm	mm
in	-	/ 12	/ 36	* 0.0254	* 2.54	* 25.4
ft	* 12	-	/ 3	* 0.30479	* 30.479	* 304.79
yd	* 36	* 3	-	* 0.9144	* 91.44	* 914.4
m	/ 0.0254	/ 0.30479	/ 0.9144	-	* 100	* 1000
cm	/ 2.54	/ 30.479	/ 91.44	/ 100	-	* 10
mm	/ 25.4	/ 304.79	/ 914.4	/ 1000	/ 10	-

9.1.2 Mass

	lb	oz	slug	kg	g
lb	-	* 16	* 0.03108095	* 0.4535924	* 453.5924
oz	/ 16	-	* $1.942559 \cdot 10^{-3}$	* 0.02834952	* 28.34952
slug	/ 0.03108095	/ $1.942559 \cdot 10^{-3}$	-	* 14.5939	* 14593.9
kg	/ 0.45359237	/ 0.02834952	/ 14.5939	-	* 1000
g	/ 453.59237	/ 28.34952	/ 14593.9	/ 1000	-

9.1.3 Force

	lb	oz	p	dyne	N
lb	-	* 16	* 453.55358	* 444822.2	* 4.448222
oz	/ 16	-	* 28.349524	* 27801	* 0.27801
p	/ 453.55358	/ 28.349524	-	* 980.7	* $9.807 \cdot 10^{-3}$
dyne	/ 444822.2	/ 27801	/ 980.7	-	/ $100 \cdot 10^3$
N	/ 4.448222	/ 0.27801	/ $9.807 \cdot 10^{-3}$	* $100 \cdot 10^3$	-

9.1.4 Power

	HP	W
HP	-	* 746
W	/ 746	-

9.1.5 Rotation

	min ⁻¹ (RPM)	rad/s	deg./s
min ⁻¹ (RPM) -		* $\pi / 30$	* 6
rad/s	* $30 / \pi$	-	* 57.295
deg./s	/ 6	/ 57.295	-

9.1.6 Torque

	lb-in	lb-ft	oz-in	Nm	kp-m	kp-cm	dyne-cm
lb-in	-	/ 12	* 16	* 0.112985	* 0.011521	* 1.1521	* $1.129 \cdot 10^6$
lb-ft	* 12	-	* 192	* 1.355822	* 0.138255	* 13.8255	* $13.558 \cdot 10^6$
oz-in	/ 16	/ 192	-	* $7.0616 \cdot 10^{-3}$	* $720.07 \cdot 10^{-6}$	* $72.007 \cdot 10^{-3}$	* 70615.5
Nm	/ 0.112985	/ 1.355822	/ $7.0616 \cdot 10^{-3}$	-	* 0.101972	* 10.1972	* $10 \cdot 10^6$
kp-m	/ 0.011521	/ 0.138255	/ $720.07 \cdot 10^{-6}$	/ 0.101972	-	* 100	* $98.066 \cdot 10^6$
kp-cm	/ 1.1521	/ 13.8255	/ $72.007 \cdot 10^{-3}$	/ 10.1972	/ 100	-	* $0.9806 \cdot 10^6$
dyne-cm	/ $1.129 \cdot 10^6$	/ $13.558 \cdot 10^6$	/ 70615.5	/ $10 \cdot 10^6$	/ $98.066 \cdot 10^6$	/ $0.9806 \cdot 10^6$	-

9.1.7 Moment of inertia

	lb-in ²	lb-ft ²	kg-m ²	kg-cm ²	kp-cm-s ²	oz-in ²
lb-in ²	-	/ 144	/ 3417.16	/ 0.341716	/ 335.109	* 16
lb-ft ²	* 144	-	* 0.04214	* 421.4	* 0.429711	* 2304
kg-m ²	* 3417.16	/ 0.04214	-	* $10 \cdot 10^3$	* 10.1972	* 54674
kg-cm ²	* 0.341716	/ 421.4	/ $10 \cdot 10^3$	-	/ 980.665	* 5.46
kp-cm-s ²	* 335.109	/ 0.429711	/ 10.1972	* 980.665	-	* 5361.74
oz-in ²	/ 16	/ 2304	/ 54674	/ 5.46	/ 5361.74	-

9.1.8 Temperature

	°F	°C	K
°F	-	(°F - 32) * 5/9	(°F - 32) * 5/9 + 273.15
°C	°C * 9/5 + 32	-	°C + 273.15
K	(K - 273.15) * 9/5 + 32	K - 273.15	-

9.1.9 Conductor cross section

AWG	1	2	3	4	5	6	7	8	9	10	11	12	13
mm ²	42.4	33.6	26.7	21.2	16.8	13.3	10.5	8.4	6.6	5.3	4.2	3.3	2.6
AWG	14	15	16	17	18	19	20	21	22	23	24	25	26
mm ²	2.1	1.7	1.3	1.0	0.82	0.65	0.52	0.41	0.33	0.26	0.20	0.16	0.13

9.2 Terms and Abbreviations

<i>Address</i>	Storage which can be accessed by its unique number. See also Slave address.
<i>AK</i>	Request/response identifier
<i>Big Endian format</i>	Method of storing data; the highest-value byte of a data word is at the first position (big end first).
<i>Default value</i>	Factory setting.
<i>DP</i>	D e z entralized P eriphery
<i>Direction of rotation</i>	Rotation of the motor shaft in a clockwise or counterclockwise direction of rotation. Clockwise rotation is when the motor shaft rotates clockwise as you look at the end of the protruding motor shaft.
<i>EMC</i>	Electromagnetic compatibility
<i>Error</i>	Discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition.
<i>Error class</i>	Classification of errors into groups. The different error classes allow for specific responses to faults, for example by severity.
<i>Fatal error</i>	In the case of fatal error, the product is not longer able to control the motor, so that an immediate deactivation of the power stage is necessary.
<i>Fault</i>	Operating state of the drive caused as a result of a discrepancy between a detected (computed, measured or signaled) value or condition and the specified or theoretically correct value or condition.
<i>Fault reset</i>	A function used to restore the drive to an operational state after a detected error is cleared by removing the cause of the error so that the error is no longer active (transition from operating state "Fault" to state "Operation Enable").
<i>FMS</i>	Fieldbus Message Specification
<i>GSD file</i>	A file provided by the vendor; contains specific information on a Profibus device and is required for commissioning the device
<i>I/O</i>	Inputs/outputs
<i>Idx</i>	Index value of a parameter
<i>LED</i>	Light Emitting Diode
<i>Limit switch</i>	Switches that signal overtravel of the permissible range of travel.
<i>Little Endian format</i>	Method of storing data; the lowest-value byte of a data word is at the first position (little end first).
<i>Master</i>	Active bus device that controls the data traffic on the network.
<i>MT</i>	M ode T oggle, toggling a bit from 0 -> 1 or 1 -> 0
<i>Parameter</i>	Device data and values that can be set by the user.
<i>PKE</i>	Parameter identifier
<i>PNO</i>	Profibus user organization
<i>Profibus</i>	Standardized open fieldbus as per EN 50254-2 which allows drives and other devices from different manufacturers to communicate.
<i>PWE</i>	Parameter value

<i>PZD</i>	Process data
<i>Quick Stop</i>	Function used to enable fast deceleration of the motor via a command or in the event of an error.
<i>Six</i>	Subindex value of a parameter
<i>Slave address</i>	Communication between master and slave is only possible after the assignment of unique addresses.
<i>Slave</i>	Passive bus device that receives control commands and provides data to the master.
<i>PLC</i>	Programmable logic controller
<i>Toggle</i>	see MT, ModeToggle
<i>Warning</i>	If the term is used outside the context of safety instructions, a warning alerts to a potential problem that was detected by a monitoring function. A warning is not an error and does not cause a transition of the operating state.
<i>Watchdog</i>	Unit that monitors cyclic basic functions in the product. The power stage is disabled and the outputs are switched off in the event of errors.

10 Index

A

Abbreviations 61

B

Before you begin
 Safety information 9
Big Endian format 21
Bit 21
Byte 21

C

Commissioning 17
 prerequisites 17

D

Data structure 21
Data telegram 21
Diagnostics 55
Directives and standards 7
Documentation and literature references 7
Double word 21

E

Examples 39

G

Generic Station Description GSD 12
Glossary 59

I

Ident number 12
Index 22, 23
Introduction 7

L

LEDs at HMI
 for Profibus 55
Literature 7

M

Mapping 31, 46
Master 11
Master-slave relationship 12
Mode Toggle 46

N

Network mode yyyy 18

O

Operation 19

P

Parameter channel 22

Parameter identifier 23

Peripheral memory 37

PKE 23

Prerequisites for commissioning 17

Process data channel 26

Process image 37

Profibus

 LEDs at HMI 55

Profibus User Association 7

PWE 23

R

Real-time data exchange 26

RS-485 technology 12

S

Slave 11

Subindex 22

Synchronous errors 52

T

Terms 61

Token-Passing method 12

Troubleshooting 55

U

Units and conversion tables 59

W

Word 21