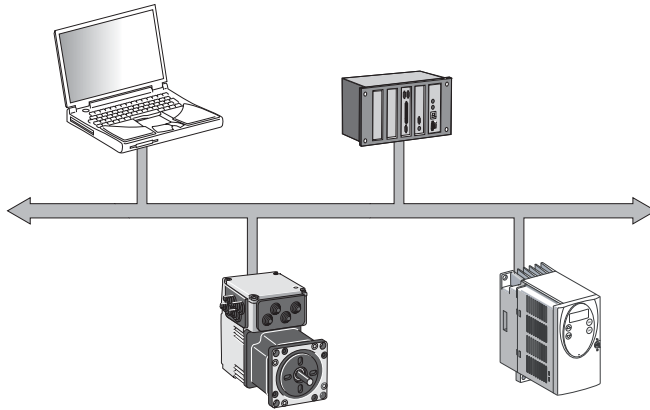


## Technical Documentation

---



Fieldbus manual

Fieldbus protocol for intelligent  
Compact Drive IclA IFx

### **Profibus DP V0**

Order no.: 0098441113193

Edition: V1.03, 04.2006

Berger Lahr GmbH & Co. KG  
Breslauer Str. 7  
D-77933 Lahr

## Important information

The drive systems described here are products for general use that conform to the state of the art in technology and are designed to prevent any dangers. However, drives and drive controllers that are not specifically designed for safety functions are not approved for applications where the functioning of the drive could endanger persons. The possibility of unexpected or unbraked movements can never be totally excluded without additional safety equipment. For this reason personnel must never be in the danger zone of the drives unless additional suitable safety equipment prevents any personal danger. This applies to operation of the machine during production and also to all service and maintenance work on drives and the machine. The machine design must ensure personal safety. Suitable measures for prevention of property damage are also required.

See safety section for additional critical instructions.

**Not all product variants are available in all countries.**

Please consult the current catalogue for information on the availability of product variants.

We reserve the right to make changes during the course of technical developments.

All details provided are technical data and not promised characteristics.

In general, product names must be considered to be trademarks of the respective owners, even if not specifically identified as such.

## Table of Contents

<b>Important information</b> .....	<b>-2</b>
<b>Table of Contents</b> .....	<b>-3</b>
<b>Writing conventions and symbols</b> .....	<b>-5</b>
<b>1 Introduction</b>	
1.1 Documentation and literature references .....	1-1
1.2 Directives and standards .....	1-1
<b>2 Safety</b>	
2.1 Qualification of personnel .....	2-1
2.2 Intended use .....	2-1
2.3 General safety instructions .....	2-2
<b>3 Basics</b>	
3.1 Profibus technology .....	3-1
3.1.1 Profibus transmission technology .....	3-1
3.1.2 Network topology .....	3-1
3.1.3 Access procedures .....	3-2
3.1.4 Transmission technology in the network .....	3-2
3.1.5 Device identification .....	3-2
3.2 Fieldbus devices in the Profibus-DP network .....	3-3
3.3 Operating modes and functions .....	3-3
<b>4 Installation</b>	
4.1 Electromagnetic compatibility, EMC .....	4-1
4.2 Profibus DP interface .....	4-2
<b>5 Commissioning</b>	
5.1 Requirements for commissioning .....	5-1
5.2 Initiating network operation .....	5-2
5.3 Running function test .....	5-2
<b>6 Operation</b>	
6.1 Profibus parameters .....	6-1
6.2 Profibus communication profile .....	6-1
6.2.1 Profibus DP V0 communication .....	6-1
6.2.2 Data structure .....	6-2
6.2.3 Parameters and process data channel .....	6-3

6.3	Parameter channel . . . . .	6-4
6.3.1	Overview . . . . .	6-4
6.3.2	Structure of the parameter channel . . . . .	6-5
6.4	Process data channel . . . . .	6-7
6.4.1	Overview . . . . .	6-7
6.4.2	Structure of the process data channel . . . . .	6-8
6.5	PLC as a fieldbus master device . . . . .	6-16
6.5.1	Response with special master commands. . . . .	6-18
<b>7</b>	<b>Examples</b>	
7.1	Overview of examples . . . . .	7-1
7.2	Use of the parameter channel . . . . .	7-2
7.2.1	Write parameter . . . . .	7-2
7.2.2	Read parameter . . . . .	7-2
7.2.3	Synchronous errors . . . . .	7-3
7.3	Operating states in the process data channel . . . . .	7-4
7.3.1	Switch power amplifier on and off . . . . .	7-5
7.3.2	Trigger Quick Stop . . . . .	7-5
7.3.3	Fault reset . . . . .	7-6
7.4	Operating modes in the process data channel . . . . .	7-7
7.4.1	Absolute positioning . . . . .	7-8
7.4.2	Relative positioning . . . . .	7-9
7.4.3	Profile velocity . . . . .	7-10
7.4.4	Homing by dimension setting . . . . .	7-11
7.4.5	Reference movement . . . . .	7-12
7.5	Error signalling in process data channel. . . . .	7-13
7.5.1	Synchronous errors . . . . .	7-13
7.5.2	Asynchronous errors . . . . .	7-14
<b>8</b>	<b>Service, maintenance and disposal</b>	
8.1	Replacing units . . . . .	8-1
<b>9</b>	<b>Diagnostics and troubleshooting</b>	
9.1	Fieldbus communication error diagnosis . . . . .	9-1
9.2	Error messages . . . . .	9-2
9.2.1	Synchronous errors . . . . .	9-2
9.2.2	Asynchronous errors . . . . .	9-3
9.2.3	Errors during operating mode control . . . . .	9-4
<b>10</b>	<b>Glossary</b>	
10.1	Terms and Abbreviations . . . . .	10-1
<b>11</b>	<b>Index</b>	

## Writing conventions and symbols

*Work steps* If work steps must be carried out in sequence, they are shown as follows:

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

If a response to a work step is specified, this will inform you that the step has been carried out correctly.

Unless otherwise stated, the individual instruction steps must be carried in the given sequence.

*Lists* Lists can be sorted alphanumerically or by priority. Lists are structured as follows:

- Point 1
- Point 2
  - Subpoint to 2
  - Subpoint to 2
- Point 3

*Making work easier* Information on making work easier can be found at this symbol:



*This offers supplementary information on making work easier.  
See the chapter on safety for an explanation of the safety instructions.*

*Parameters* Parameters are shown as follows:

Group.Name Index:Subindex



# 1 Introduction

The 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 the online command processing for products in the Profibus-DP V0 fieldbus network.

## 1.1 Documentation and literature references

- Documentation*
- Product manuals for IclA compact drives:
    - IclA IFS fieldbus stepper motor intelligent compact drive
    - EC motor IclA IFE fieldbus intelligent compact drive
    - Servo motor IclA IFA fieldbus intelligent compact drive
  - Data sheets for IclA in the IclA Intelligent Compact Drives catalogue
- Literature*
- PROFIBUS Specification (FMS, DP, PA); Profibus User Organisation
  - Popp, M: PROFIBUS-DP/DPV1; Grundlagen, Tipps und Tricks für Anwender [Basics, Tips and Tricks for Users]; ISBN 3-7785-2781-9

## 1.2 Directives and standards

- Regulations, standards*
- DIN 19245, Parts 1 to 3: PROFIBUS-FMS
  - EN50170, fieldbus standard
- Profibus User Association*
- Profibus Nutzerorganisation e.V. (PNO)  
Interessenvertretung der Profibusanwender  
Haid-und-Neu-Str. 7  
D-76131 Karlsruhe
- Profibus international in the Internet*
- <http://www.profibus.com>





## 2 Safety

### 2.1 Qualification of personnel

Only technicians who are familiar with and understand the contents of this manual and the other relevant manuals are authorised to work on and with this drive system. The technicians must be able to detect potential dangers that may be caused by setting parameters, changing parameter values and generally by the mechanical, electrical and electronic equipment.

The technicians must have sufficient technical training, knowledge and experience to recognise and avoid dangers.

The technicians must be familiar with the relevant standards, regulations and safety regulations that must be observed when working on the drive system.

### 2.2 Intended use

The drive systems described here are products for general use that conform to the state of the art in technology and are designed to prevent any dangers. However, drives and drive controllers that are not specifically designed for safety functions are not approved for applications where the functioning of the drive could endanger persons. The possibility of unexpected or unbraked movements can never be totally excluded without additional safety equipment. For this reason personnel must never be in the danger zone of the drives unless additional suitable safety equipment prevents any personal danger. This applies to operation of the machine during production and also to all service and maintenance work on drives and the machine. The machine design must ensure personal safety. Suitable measures for prevention of property damage are also required.

In the system configuration described the drive systems must be used in industrial applications only and must have a fixed connection only.

In all cases the applicable safety regulations and the specified operating conditions, such as environmental conditions and specified technical data, must be observed.

The drive system must not be commissioned and operated until completion of installation in accordance with the EMC regulations and the specifications in this manual.

To prevent personal injury and damage to property damaged drive systems must not be installed or operated.

Changes and modifications of the drive systems are not permitted and if made all no warranty and liability will be accepted.

The drive system must be operated only with the specified wiring and approved accessories. In general, use only original accessories and spare parts.

The drive systems must not be operated in an environment subject to explosion hazard (ex area).

## 2.3 General safety instructions

### **⚠ DANGER**

#### **Risk of injury by complex system.**

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

- Only start the system if there are no persons in the operating zone of the moving components and the system can be operated safely.

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

### **⚠ WARNING**

#### **Danger of injury by loss of control!**

- Observe the accident prevention regulations. (For USA see also NEMA ICS1.1 and NEMA ICS7.1)
- The system manufacturer must take the potential error possibilities of the signals and the critical functions into account to ensure a safe state during and after errors. Some examples are: emergency stop, final position limitation, power failure and restart.
- The assessment of error possibilities must also include unexpected delays and the failure of signals or functions.
- Suitable redundant control paths must be in place for dangerous functions.
- Check that measures taken are effective.

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

### **⚠ CAUTION**

#### **Danger of injury and damage to system components by evaluation of faulty control commands!**

If a PLC is used as the master unit, the exchange of data can lead to inconsistent transmission data as a result of fieldbus and PLC cycles not operating synchronously.

- Please observe the notes concerning the operation using PLC.

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

## 3 Basics

### 3.1 Profibus technology

#### 3.1.1 Profibus transmission technology

Profibus is available in three types that can be used for time-critical and complex communications tasks:

- Profibus-FMS
- Profibus-PA
- Profibus-DP

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

Profibus-PA (PA: Process Automation) is primarily used in process technology, such as process automation. Profibus-PA networks are characterised by their ability to use sensors and actuators in explosion-endangered areas, and to provide data communication and power to devices over the bus.

Profibus-DP (DP: Distributed Periphery) is the fast Profibus version. It is specially tailored for communications in manufacturing areas. Features include simple connection of new products into the bus and high transmission speeds.

The drive system with Profibus-DP described here supports various parameter message frames as per the Profibus-DP V0 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 linked together by the Profibus-DP network cable.

*Master* The master controls the data traffic in the network. Examples for master:

- automation devices, e.g. PLCs
- PCs
- programming devices

*Slave* They receive control commands and supply data to the master. Examples for 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:

- the Token-Passing method
- The Master-Slave method

*the 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 authorisation for a specified period.

*The Master-Slave method*

Data is exchanged with the product with the master-slave method. The slave has a transmit and receive buffer through which it provides and receives data. The master reserves a memory area with a transmit and receive buffer for every slave.

Data exchange between master and slave is cyclical. The master device sends commands to the slave device and receives data back from the slave in the next cycle. The bus cycle is extended for transmission of repeat message frames only in case of fault

Drive systems are linked into the network as slaves, therefore they do not use the token passing system.

**3.1.4 Transmission technology in the network**

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

Drive systems work with RS-485 technology and are linked to a Profibus-DP network with two-wire cables.

*RS485 technology*

RS485 technology is a simple method of transmission over two-wire twisted-pair cables. It can handle transmission rates from 9.6 kbit/s to 12 Mbit/s.

**3.1.5 Device identification**

*Device master data file*

The specific features of a Profibus product are described in the device master data file (GSD file). This file is supplied with the product by the manufacturer and must be read by the network configuration programme.

The GSD file contains all information on the operation of the product in the Profibus-DP network, such as manufacturer's specifications and product identification, supported baud rate, levels and meaning of plug 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.

*Identity number*

A master device uses the ID number to identify the device class of the connected slave. The ID number is a unique number allocated for a specific device class by the Profibus user organisation.

*Slave address*

In the network every device must be allocated a unique address between 1 and 126. Slaves normally occupy the address space 3...126. The master (normally address 0...2) can contact every slave directly from this address. The exact setting of the address for the product described here is described in the product manual.

0098441113193, V1.03, 04.2006

### 3.2 Fieldbus devices in the Profibus-DP network

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

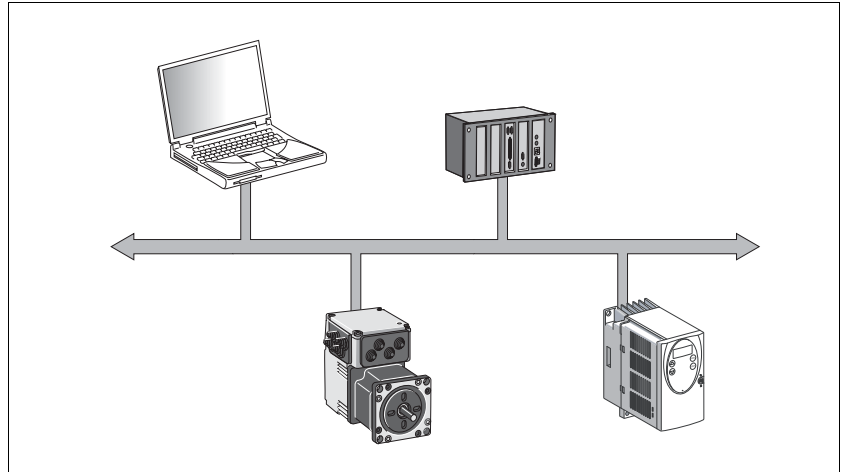


Figure 3.1 Fieldbus products in the network

### 3.3 Operating modes and functions

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

*Setting options* The following settings can be made over the fieldbus:

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



## 4 Installation

### ▲ WARNING

**Danger of injury by loss of control!**

- Observe the accident prevention regulations. (For USA see also NEMA ICS1.1 and NEMA ICS7.1)
- The system manufacturer must take the potential error possibilities of the signals and the critical functions into account to ensure a safe state during and after errors. Some examples are: emergency stop, final position limitation, power failure and restart.
- The assessment of error possibilities must also include unexpected delays and the failure of signals or functions.
- Suitable redundant control paths must be in place for dangerous functions.
- Check that measures taken are effective.

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

### 4.1 Electromagnetic compatibility, EMC

The following measures are necessary for trouble-free fieldbus operation. They supplement the EMC measures in the product manual.

EMC measures	Effect
Use wiring with braided and foil shielding	Discharge of interference currents
Fieldbus lines and signal lines must not be laid out in the same cable conduit as lines for DC and AC voltage over 60 V <sup>1)</sup> Recommendation: lay in separate conduits at least 20 cm apart.	Prevention of mutual interference
Use bonding conductors in system with – wide-area installation – different voltage infeed – networking between different buildings	Discharge of interference currents
Use fine-core bonding conductors	Deflect even high-frequency interference currents
Circuit breaker if there is danger of overvoltage or lightning strike	Protection against damage by overvoltage

1) Fieldbus lines can be laid out in one conduit with signal and analogue lines

Table 4.1 EMC measures

*Equipotential bonding conductors*

The shields are connected at both ends for fault protection. Potential differences can result in excessive currents on the shield and must be prevented by equipotential bonding conductor cables.

If lines over 100 m are approved, the following applies: up to 200 m length a cable cross section of 16 mm<sup>2</sup> is sufficient, for greater lengths a cable cross section of 20 mm<sup>2</sup> is required.

For more information please see the product manual.

## 4.2 Profibus DP interface

### **⚠ WARNING**

#### **Danger of injury and damage to system components by loss of degree of protection**

Foreign bodies, deposits or humidity can cause unexpected device responses.

- Prevent any foreign bodies from entering the terminal unit.
- Do not remove the electronic case cover. Only remove the plug cover.
- Check that seals and cable glands are correctly seated.

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

*Function* With the Profibus-DP interface you can connect the drive system to a Profibus network as a slave.

The drive system includes data and commands from a higher level bus device, the master. Status information such as operating status and processing status are sent to the master as acknowledgment.

Consult the product manual for the exact terminal assignments and the settings for the address and the terminating resistor.



## 5 Commissioning

### DANGER

#### **Risk of injury by complex system.**

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

- Only start the system if there are no persons in the operating zone of the moving components and the system can be operated safely.

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

### WARNING

#### **Danger of personal injury and damage to system parts by uncontrolled system operation!**

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

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

### 5.1 Requirements for commissioning

The following components are required for commissioning:

- product with Profibus-DP interface
- GSD file on data medium
- Product manual for the described product
- manual for the Profibus-DP fieldbus (this manual)

Read the manuals carefully before commissioning and take particular note of the safety instructions!



*Drivers for Siemens PLCs can be obtained from your local agent as required.*

## 5.2 Initiating network operation

Network operation is started via a master. This can be a PLC or a PC with the appropriate user software with which commands can be sent and received data read.

## 5.3 Running function test

Test all functions that are important for your system. Run the function tests first with no coupled load. Also check the operating temperature under normal operation and the response of the system to power failure.

### *Steps for troubleshooting*

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

- Is the product switched on and is the master started for network operation?
- Are all cable connections in good mechanical repair?
- Is the LED at the fieldbus input on? The LED shows data traffic over the network interface.
- Is the correct address set?

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

## 6 Operation

The "Operation" section describes the basic operating statuses, operating modes and functions of the product.

### 6.1 Profibus parameters

Several specific parameters are used when working with the Profibus protocol – these are explained in the fieldbus manual. Please consult the product manual for an overview of all parameters including explanations and examples.

Parameter	Meaning, reference	Page
Profibus.MapIn	Mapping of PZD5+PZD6, drive ⇒ master	6-11
Profibus.MapOut	Mapping of PZD5+PZD6, master ⇒ drive	6-11
Profibus.PkInhibit	Lockout period for read jobs	6-5
Profibus.SafeState	Response of the drive in the Profibus master "Clear" status	6-18

### 6.2 Profibus communication profile

#### **▲ WARNING**

**Danger of personal injury and damage to system parts by uncontrolled system operation!**

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

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

#### 6.2.1 Profibus DP V0 communication

Profibus DP V0 provides the basic functions of DP. This includes cyclic data exchange, stations, module and channel-specific diagnostics and various alarm types for diagnostics and process alarms for removal and insertion of bus devices.

*Master-slave relationship* The master periodically reads receive information from the slaves and periodically writes transmit information to the slaves. Receive and transmit data for one slave are transmitted as one unit in one message cycle.

*Command processing: Transmitted data and received data* The master sends a command to the drive system (slave) to execute a travel command, enable operating functions or request information from the slaves. The slave executes the command and acknowledges it with a result message.

The exchange of data follows a fixed routine:

- Transmission data to the slave: The master device places a command in the transmission data memory. It is sent from there to the slave and executed.
- Received data from the slave: The slave acknowledges the execution status of the command in the received data. If the master device receives an acknowledgement with no error message, the command was correctly executed.

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

In the case of cyclical fieldbus transmission, the master device automatically receives the latest received data from the slave device at every cycle. It can recognise via the acknowledgement mechanism whether the received data were status information from the slave device or an answer to a command sent previously. The slave device also uses the acknowledgement mechanism to recognise a new command.

*Commands* The master device transmits control commands and action commands with the transmitted data.

- After a control command the master receives a feedback message indicating whether the process was successfully executed and could be completed
- With an action command the slave simply reports whether an action or a travel command could be successfully started.

The master must then continuously monitor the conclusion of the process command by evaluating the slave received data.

## 6.2.2 Data structure

In addition to command and control information, output and received data also contain administration data for monitoring network operations. Administration data are provided by the user programme in the master device.

The transmit data and received data must be programmed and evaluated for the purposes of cyclical communication in the network.

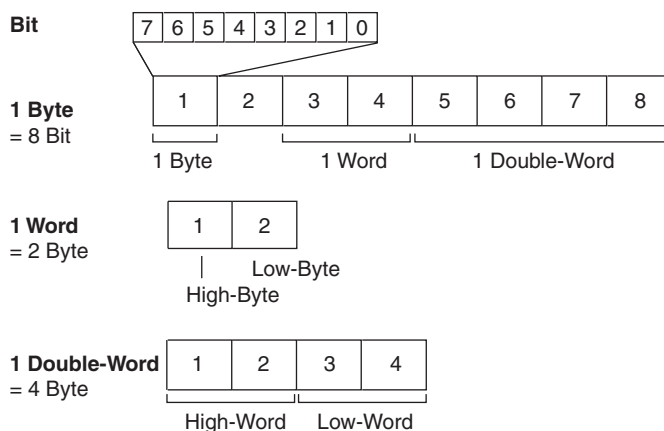


Figure 6.1 general data structure from the bit to the double word

The index and sub-index data in the parameter descriptions must be specified in hexadecimal form. The data frame with transmission and received data and all byte, word and double-word values are output in hexadecimal form. Hexadecimal characters are indicated by an "h" behind the numerical value, e.g. "31<sub>h</sub>", decimal characters have no special identification. Note the different counting format of bit (0...7, right to left) and byte (1-xx, left to right).

*Word order: big-endian format*

The Profibus data are transmitted in big-endian format, i.e. number values over one byte are treated as if they were in the decimal system. Example: the index value is transmitted in bytes 3 and 4, the index 21<sub>h</sub> is therefore shown as 0021<sub>h</sub>.

### 6.2.3 Parameters and process data channel

Communication between master and slave is based on the PPO Type 2 data frame of the PNO user organisation Profidrive profile. The content of the data does not correspond with the Profidrive profile! The data frame is 20 bytes. The first 8 bytes are used for parameter transmission, the following 12 bytes (bytes 9...20) transmit the process data. This is 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

The abbreviations in use have the following meanings:

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

0098441113193, V1.03, 04.2006

*Index, subindex* The index occupies bytes 3 and 4 of the data frame and the subindex corresponds with byte 2.

The parameters are addressed using an index with a 16-bit range and a subindex with an 8-bit range. A data field consists of one subindex entry value. The individual data fields of a parameter are given by the subindex entries.

Index and subindex are in hexadecimal form, recognisable by a subscript "h".

The following example shows index and subindex entries for the parameter IO.IO\_def, 0022<sub>h</sub> for configuring the digital inputs and outputs.

Index	subindex	Parameter	Description
0022 <sub>h</sub>	01 <sub>h</sub>	IO.IO0_def	Configuration of IO 0
0022 <sub>h</sub>	02 <sub>h</sub>	IO.IO1_def	Configuration of IO 1
0022 <sub>h</sub>	03 <sub>h</sub>	IO.IO2_def	Configuration of IO 2
0022 <sub>h</sub>	04 <sub>h</sub>	IO.IO3_def	Configuration of IO 3

Table 6.1 Examples of index and subindex entries

See the product manual for a list of all parameters.

*The number format of the parameter values can be found in the product manual.*



## 6.3 Parameter channel

### 6.3.1 Overview

The master can request or change a parameter value from the slave via the parameter channel (the first 8 bytes of the 20-byte data frame). The parameters are subdivided into different categories. Every category is allocated to an index. Every parameter can be uniquely addressed via the index and subindex.

### 6.3.2 Structure of the parameter channel

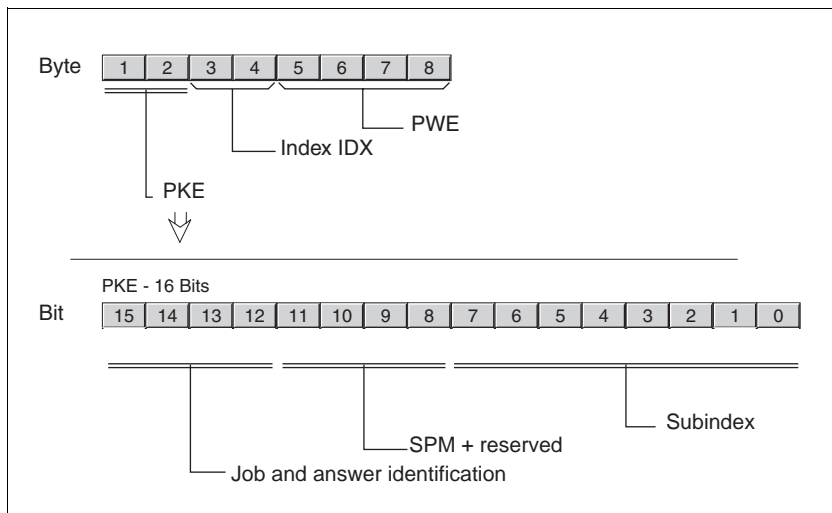


Figure 6.2 Parameter channel: parameter identifier in byte 1 and 2

The following abbreviations are used for the parameter channel:

Byte 1+2: PKE as parameter identifier (job identifier+subindex)

Byte 3+4: IDX for index of parameter (parameter number)

Byte 5...8: PWE for parameter value

*Parameter identifier PKE*

The job identifier or the response identifier and the subindex of the parameter are in the first two bytes (parameter identifier PKE). The index is entered in bytes 3 and 4. The parameter value is in bytes 5...8 (PWE).

The job identifier or response identifier (bit 12-15) indicates which fields of the parameter channel must be evaluated.

Job identifier	Function	Response identifier	
		positive	negative
0	no job	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 Job and response identifier

**Write jobs (change parameter value) are only executed by the slave if the value of the job identifier changes from 0 to 2 or 3.** Read jobs are executed as often as the value of job identifier is equal to 1. To limit the load on the system, a cycle time between two read processes is defined with the parameter `PKINHIBIT`. The read job is executed again after elapse of the cycle time.

If the response identifier to a requested job is 0, the slave has not yet completed the job. The slave signals to the master that the job has been

successfully carried out 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). For a negative response identifier, the error number is in bytes 5...8 (parameter value).

Only one job can be in process at any one time. The slave provides the response until the master sends a new job. In the case of responses that include parameter values the slave always answers with the current value on repeat (cyclic process).

Bits 8...11 (Reserved) must always be 0.

Group.Name Index:Sub-index dec. (hex.)	Meaning Bit assignment	Data type range (dec.)	Unit Default (dec.)	R/W/per. Info
Profibus.PkInhibit 24:4 (18:04 <sub>h</sub> )	Refresh cycle for static read jobs The reader value is refreshed cyclically at the defined period with a static pending read job.	UINT32 1..60000	ms 1000	R/W/per

*Example: error-free reading of a parameter*

In the example the programme number of the Profibus software must be read. The programme number is stored in the device configuration range (Index 0D<sub>h</sub>; Subindex B<sub>h</sub>). The master sends a read job to the slave. After processing the slave places the requested data in bytes 5...8 (parameter value PWE). The parameter value read has the decimal value 826<sub>d</sub> which corresponds to 33A<sub>h</sub>. Because this is a double word, the corresponding positive response identifier must be 2.

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

Transmission data: Index: 13=0D<sub>h</sub>, Subindex: 11=0B<sub>h</sub>

Parameter	PKE, 1st byte	PKE, 2nd byte (Sdx)	Idx	Data	Description
Tx 000D <sub>h</sub> :09 <sub>h</sub>	10 <sub>h</sub>	0B <sub>h</sub>	000D <sub>h</sub>	xxxx xxxx	Read the programme number. The data are meaningless.

The 4 data bytes are meaningless for a read request.

Received data:

Parameter	PKE, 1st byte	PKE, 2nd byte (Sdx)	Idx	Data	Description
Rx 000D <sub>h</sub> :09 <sub>h</sub>	20 <sub>h</sub>	0B <sub>h</sub>	000D <sub>h</sub>	0000 033A <sub>h</sub>	Data 0000033A correspond to the programme number.

Based on the response identifier (2 or 1) in the PKE (bit 12+13), the drive system 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 and UINT16 data types). For 16-bit data the user information is in the last two data bytes, the first two data bytes have no significance.

The message is provided until the master sends the job identifier 0 to the slave before the next job.

*Example: error in writing a parameter*

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



The slave is now ready to execute new jobs. The value of a non-existent parameter must be changed for the example. The value of the parameter with index 01 = 01<sub>h</sub> and sub-index 255<sub>d</sub> (corresponds to FF<sub>h</sub>) should be changed to 222 = DE<sub>h</sub>.

Index:1 = 0001<sub>h</sub>

Subindex:255 = FF<sub>h</sub>

Value: 222= 0000 00DE<sub>h</sub>

Parameter	PKE, 1st byte Job identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Tx 0001 <sub>h</sub> :FF <sub>h</sub>	30 <sub>h</sub>	FF <sub>h</sub>	0001 <sub>h</sub>	0000 00DE <sub>h</sub>	Write a non-existent parameter

Because the slave cannot address the parameter, an error message is returned, the parameter value in this case is 0602 0000<sub>h</sub>. Error messages in the parameter channel are identified as synchronous errors, because they are processed in the normal cyclic data exchange.

Parameter	PKE, 1st byte Response identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Rx 0001 <sub>h</sub> :FF <sub>h</sub>	70 <sub>h</sub>	FF <sub>h</sub>	0001 <sub>h</sub>	0602 0000 <sub>h</sub>	Error message 0602 0000 <sub>h</sub> is returned = parameter does not exist

For information on synchronous errors and error numbers see page 7-13 and page 9-2 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, e.g. actual and setpoint position, current operating status. The transmission can be executed very fast, because it is sent without additional administration data and does not require a response from the recipient.

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

Changing the operating states and activating the operating modes must be executed separately. An operating mode can generally only be activated if the operating status is already "OPERATION-ENABLE".

Note that a new operating mode and a new acceleration are generally only imported if the motor is at a standstill. Acceleration values are accepted in the process data channel during the motor movement, but the value is only set at the next travel 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:

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

The data format between transmission data (master to slave) and received data (slave to master) differs as follows:

*Transmission data format*

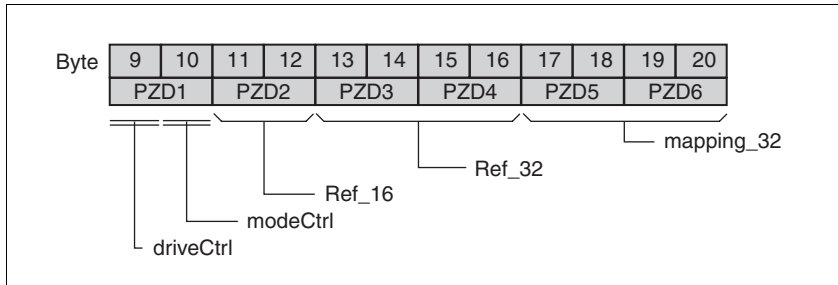


Figure 6.3 Transmission data in the process data channel: master to slave

Byte 9+10: driveCtrl and modeCtrl for setting the operating status and the operating mode. The structure is explained below, see page 6-9.

Byte 11+12: ref\_16, 16-bit reference value, assignment for set speed vel, of no significance for set dimensions.

Byte 13...16: ref\_32, 32-bit reference value, assignment dependent on operating mode, e.g. for position data Pos (consisting of Highword=Pos1 and Lowword=Pos2).

Byte 17...20: mapping, these bytes can be configured, the content is specified via index and subindex. The acceleration is entered as a default value (32-bit): acc (consisting of Highword=acc1 and Lowword=acc2). Parameters Profibus.MapOut, Profibus.MapIn are used to set which parameter is mapped in the PZD5 and PZD6. When setting parameters a check is made whether a permissible value is written. If mapping is disabled, the data in byte 17...20 are not relevant.

Description driveCtrl:

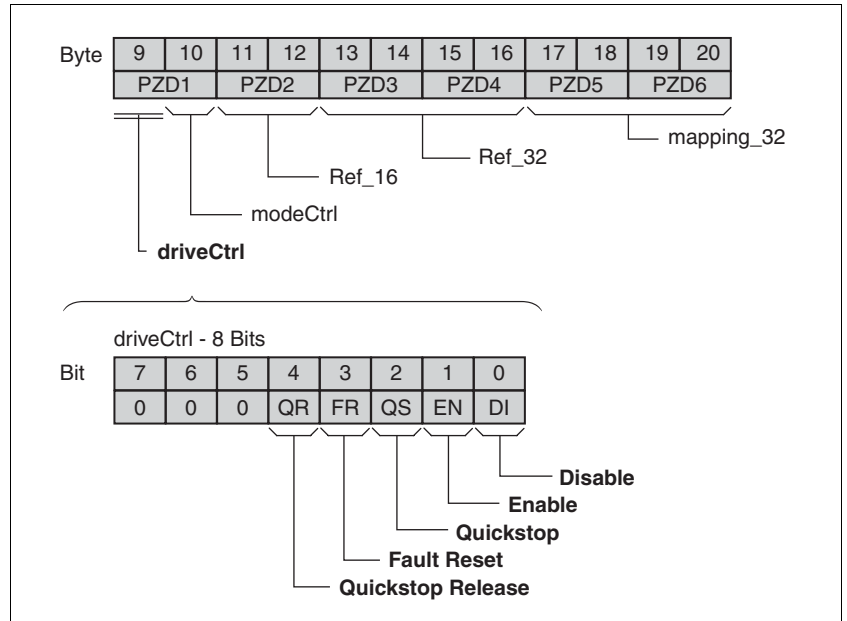


Figure 6.4 Transmit data in the process data channel: driveCtrl

The operating statuses are changed using the process data channel PZD1 driveCtrl via bits 0..7.

During access via the process data channel these bits are edge-selective, i.e. the function is triggered with a 0 > 1 edge.

*Note: the enable bit must always be set as long as the motor is powered.*



Change of operating status <sup>1) 2)</sup>	Effect on operating status <sup>3)</sup>
Bit 0: disable power amplifier	6 - 3 - 4 (Operation enable ⇒ Switch on disable ⇒ Ready to switch on)
Bit 1: enable power amplifier	4 - 5 - 6 (Ready to switch on ⇒ Switched on ⇒ Operation Enable)
Bit 2: Quick Stop	6 - 7 (Operation enable ⇒ Quick Stop active)
Bit 3: Fault Reset	9 - 3 - 4 (Fault ⇒ Switch on disable ⇒ Ready to switch on)
Bit 4: Quick Stop release	Reset Quick Stop

1) Process data channel: Process is executed at 0->1 edge  
 2) Parameter channel: Processing is run at write access if bit value =1  
 3) The operating statuses and mode transitions are described in the product manual.

Table 6.3 Change of operating status (driveCtrl)

The value zero is a special case: If all Bits 0..7 are zero on transmission, it is interpreted as a command Disable and the power amplifier is disabled.

*Fault processing*

If requests for changing the operating status cannot be implemented, these requests are ignored. There is no error response. The assignment of errors to error classes can be configured.

Non-unique bit combinations are treated in accordance with the following list:

- Bit 0 (power amplifier Disable) has priority over bit 1 (power amplifier Enable)
- Bit 2 (Quick Stop) has priority over bit 3 (Fault Reset) and bit 4 (Quick Stop Release)

Description modeCtrl:

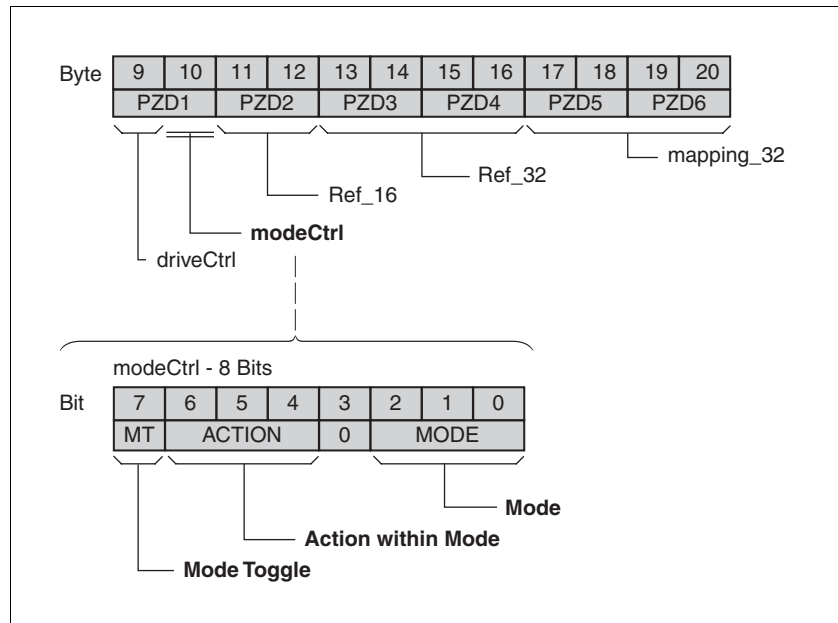


Figure 6.5 Transmit data in the process data channel: modeCtrl

The operating modes are controlled via the parameter modeCtrl. To trigger an operating mode or to change reference values the master must enter the following values:

- Reference values in fields PZD2, PZD3 and PZD4
- Select operating mode with modeCtrl, Bits 0..2 (MODE) A 0 must always be entered in bit 3.
- Select action for this operating mode with modeCtrl, Bits 4..6 (ACTION)
- Toggle modeCtrl, Bit 7 (MT)

The possible operating modes, operating actions and the associated reference values are shown in Table 6.4.

Operating mode	modeCtrl <sup>1)</sup> Bits 0..6	Description	corres- ponds to parameter <sup>2)</sup>	Reference value PZD2	Reference value PZD3+4
Jog	01h	Jog	41:1	Start (as parameter 41:1)	-
Homing	02h	Dimension setting	40:3	-	Dimension setting posi- tion
	12h	Reference movement	40:1	Type (as parameter 40:1)	-
Profile position	03h	Absolute positioning	35:1	Set speed	Setpoint position
	13h	Relative positioning	35:3	Set speed	Setpoint position
	23h	Continue positioning	35:4	-	-
Profile velocity	04h	Profile velocity	36:1	Set speed	-

1) Column corresponds to the value entered in byte modeCtrl, but without ModeToggle (bit 7)

2) Column shows Index:subindex (decimal) of the corresponding operating mode parameters, which are described in detail in the product manual.

Table 6.4 Set operating modes via modeCtrl

Setpoint positions are entered in increments, set speeds are entered in rpm.

### **▲ WARNING**

#### **Danger of personal injury and damage to system parts by uncontrolled system operation!**

- Note that inputs to these parameters are executed by the drive controller immediately on receipt of the data set.
- Make sure 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.**

With simultaneous transmission of operating mode, setpoint position and set speed in the process data channel data consistency must be ensured. Therefore, operating mode data are only evaluated if MT (Bit 7) has been toggled. Toggling means that a 0>1 or a 1>0 change of edge has been detected since the last transmission.

MT (Bit 7) is mirrored in the received data set through which the master detects the acceptance of the data by the slave.

For more information on the toggle flag see page 7-7.

#### *Mapping*

The following parameters may be entered in PZD5+PZD6: Temperatur, Spannung, Fehlernummer, Strom. The error number (matches status word) and the IO word are consistently adjustable. The (drive ⇒ master) and (master ⇒ drive) parameters are used to set which parameter is mapped with the transmit data and/or received data in the PZD5 and PZD6. During parameter setting a check is made to ensure that an allowable value is written. If mapping has been disabled, there are no defined values in bytes 17...20. .

Group.Name Index:subindex dec. (hex.)	Meaning Bit assignment	Data type range (dec.)	Unit Default (dec.)	R/W/per. Info
Profibus.MapIn 24:3 (18:03 <sub>h</sub> )	Value in PZD5+6 to the master device Index and subindex of the object that is mapped to the PPO2 during the data transfer from the drive to the master device. No mapping is active by default.Possible values:  00000000 <sub>h</sub> : No mapping active 00070020 <sub>h</sub> : error number (32:7) 0009001F <sub>h</sub> : actual speed (31:9) 0019001F <sub>h</sub> : temperature of power amplifier (31:25) 0014001F <sub>h</sub> : power supply (31:20) 000C001F <sub>h</sub> : current motor current (31:12)  Low word: index mapped object High word: Subindex mapped object	UINT32	- 0	R/W/per
Profibus.MapOut 24:2 (18:02 <sub>h</sub> )	Value in PZD5+6 to drive Index and subindex of the object that is mapped on data transfer from the master to the drive in the PPO2. The setpoint acceleration is mapped by default.  Possible values: 00000000 <sub>h</sub> : no mapping active 001A001D <sub>h</sub> : setpoint acceleration (29:26) 00010021 <sub>h</sub> : digital outputs (33:1)  Low word: Index mapped object High word: Subindex mapped object	UINT32	- see text left	R/W/per

*Example of transmit data and response in process data channel: Positioning*

A relative positioning of 20000usr (0000 4E20<sub>h</sub>) is to be carried out, current position=0. The set speed must be 1000 rpm (03E8<sub>h</sub>). The start position is 0usr.

	Master				Slave		
Trigger positioning	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 93 <sub>h</sub>	ref_16 03E8 <sub>h</sub>	ref_32 0000 4E20 <sub>h</sub>	»
Positioning running x_err = 0, x_end = 0	Received data	«	driveStat 0006 <sub>h</sub>	modeStat 83 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«
	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 93 <sub>h</sub>	ref_16 03E8 <sub>h</sub>	ref_32 0000 4 E20 <sub>h</sub>	»
Positioning complete x_err = 0, x_end = 1, x_info = 1	Received data	«	driveStat 6006 <sub>h</sub>	modeStat 83 <sub>h</sub>		32-bit act. pos. 0000 4E20 <sub>h</sub>	«

Table 6.5 Relative positioning

The entry is also made here 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.

*Example of transmit data and response in process data channel: Profile velocity*

The motor must rotate in the profile velocity at a set speed of 1000 rpm (03E8<sub>h</sub>). The master must send the following data to the slave:

		Master				Slave	
Start profile velocity at 1000 rpm	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 84 <sub>h</sub>	ref_16 03E8 <sub>h</sub>	ref_32 xxxxxxxx <sub>h</sub>	»
Motor accelerates	Received data	«	driveStat 0006 <sub>h</sub>	modeStat 84 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«
	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 84	ref_16 03E8 <sub>h</sub>	ref_32 xxxxxxxx	»
Set speed reaches x_err = 0, x_end = 0, x_info = 1	Received data	«	driveStat 2006 <sub>h</sub>	modeStat 84 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«

Table 6.6 Profile velocity

The response data always contain the current position and not the speed of the motor in "32-bit actual position". The achievement of the specified set speed is displayed via the mode bit in the status word.

*Received data format, detailed description*

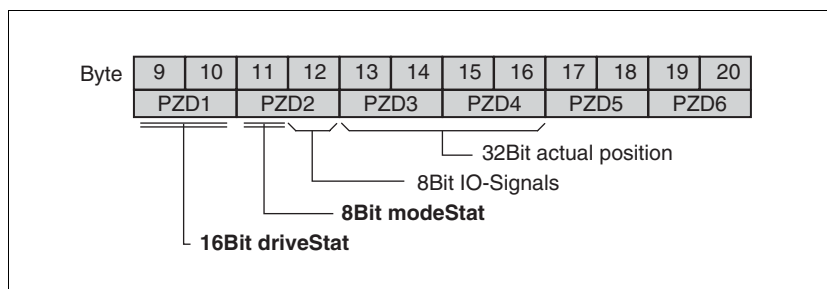


Figure 6.6 Received data in the process data channel: Slave to master

Byte 9+10: driveStat, contains the momentary operating status as field-bus status word, warning and error bits and the status of the current axis operating mode.

Byte 11: modeStat, return of the current operating mode

Byte 12: IO signals, status of input signals

Byte 13...16:"32-bit actual position", actual position data in increments

Byte 17...20: these bytes can be configured, the content is specified via index and subindex. They do not show a time consistency with bytes 9...16, by default no parameters are entered because of runtime disadvantages (value=0).

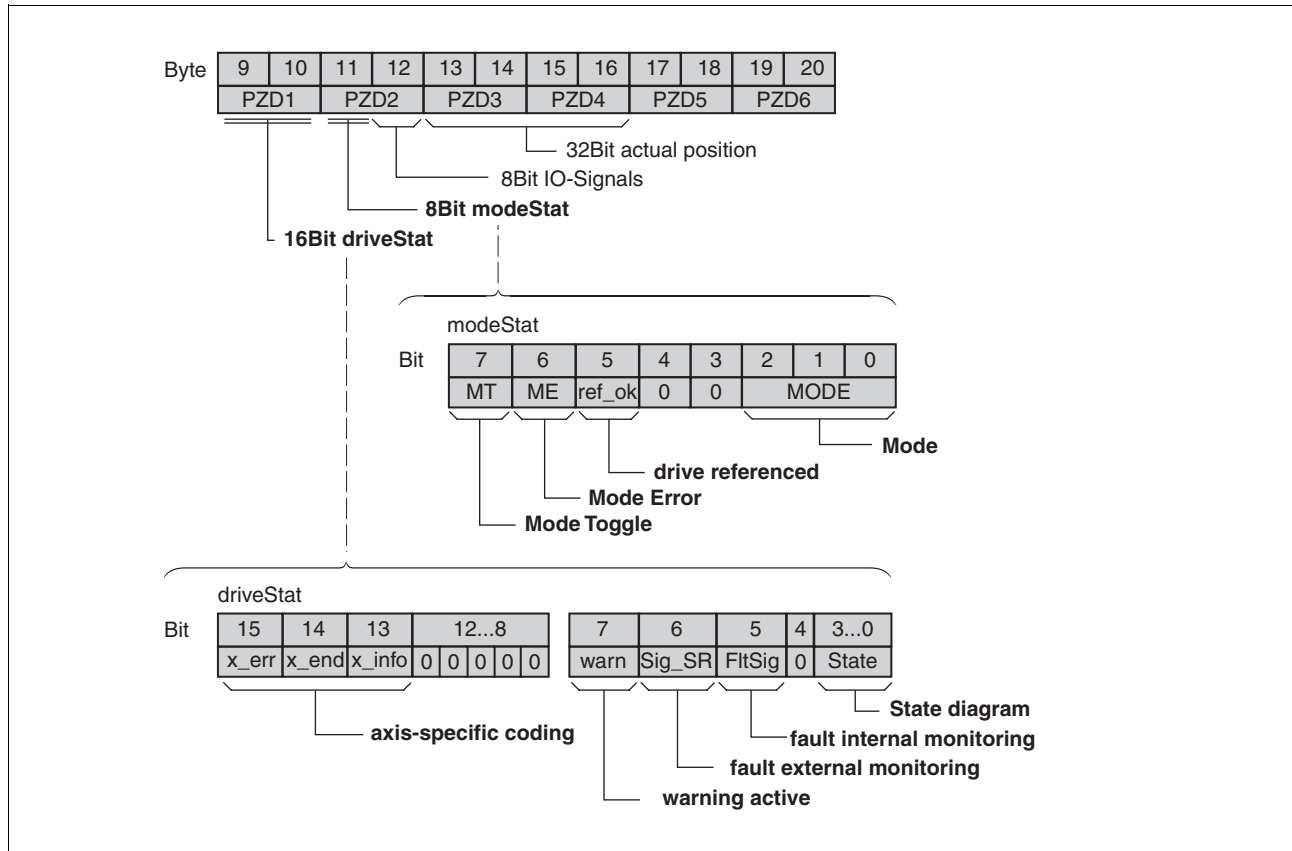


Figure 6.7 Structure of received data in the process data channel

*Description of driveStat* The information in the status word `driveStat` corresponds to Bits 0..15 of the parameter `Status.driveStat`, 28:2.

*Description of modeStat* The current processing status of the operating modes is shown with `modeStat`:

Bit	Name	Description
0...3	mode	currently specified operating mode as with transmission data
5	ref_ok	Set if the drive system was successfully referenced by a reference movement or set dimensions.
6	ME, ModeError	Set if a request of the master via transmission data was rejected.
7	MT, ModeToggle	Reflection of transmit data bit 7 (Mode Toggle), acceptance of transmit data is hereby acknowledged. Data may only be evaluated if the MT sent from the master is equal to the MT of the slave.

Table 6.7 Bitwise description of modeStat

Synchronised processing may be carried out with the transmit data `modeStat`, Bit 7 (ModeToggle – MT) and received data, Bit 6 und 7 (ModeError – ME and ModeToggle – MT). Synchronised processing means that the master waits for feedback messages from the slave and responds accordingly.



*Description of I/O signal (byte 12)* The current levels of the inputs are shown in byte 12:

Bit	Description
Bit 0	I/O level 0
Bit 1	I/O level 1
Bit 2	I/O level 2
Bit 3	I/O level 3
Bit 4	Level of safety input $\overline{PWRR\_B}$
Bit 5	Level of safety input $\overline{PWRR\_A}$
Bit 6	not assigned
Bit 7	not assigned

*Received data in the process data channel, example:* An absolute positioning of 12000usr (0000 2EE0<sub>h</sub>) is to be carried out, current position=0. The set speed must be 500 rpm (01F4<sub>h</sub>). The start position is 0usr.

		Master					Slave
Trigger positioning	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 83 <sub>h</sub>	ref_16 01F4 <sub>h</sub>	ref_32 0000 2EE0 <sub>h</sub>	»
Positioning running x_err = 0, x_end = 0	Received data	«	driveStat 0006 <sub>h</sub>	modeStat 83 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«
	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 83 <sub>h</sub>	ref_16 01F4 <sub>h</sub>	ref_32 0000 2 EE0 <sub>h</sub>	»
Positioning complete x_err = 0, x_end = 1, x_info = 1	Received data	«	driveStat 6006 <sub>h</sub>	modeStat 83 <sub>h</sub>		32-bit act. pos. 0000 2EE0 <sub>h</sub>	«

Table 6.8 Relative positioning

*Special case much shorter Positioning*

With a very short positioning it may occur that the setpoint position is already reached when the status of the received 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. This means that the case x\_end = 0 does not apply for the master. If an error has not occurred, the positioning was correctly executed even so.

*Fault processing*

If the master toggles Bit 7 (MT), this is interpreted by the slave as a request 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, Bit 6 (ModeError) is set in the received data. This bit remains set until MT (modeStat, Bit 7) (ModeToggle) is toggled again in the received data. The master can read the corresponding error code via read access to parameter BLErrorCode, 30:11..
- Continuation of the current operating mode

The current operating mode is thus not influenced and there is no change of state.

009844113193, V1.03, 04.2006

Possible reasons for a failed operating mode request:

- Reference values outside the value range
- Switching the operating mode during a process (not possible)
- Invalid mode requested
- The product is not in status 6 (Operation Enable). For more information, see product manual.

## 6.5 PLC as a fieldbus master device

The fieldbus master device allocates each connected slave its own memory space for transmitted and received data. The data exchange between PLC memory and fieldbus master can be executed via the peripheral equipment or the process image range.

The fieldbus transmission runs asynchronously to the application programme's read/write accesses of transmitted and received data. Therefore, it is possible that data are read by the fieldbus master from the PLC memory before the PLC was able to update the data in full.

### **▲ CAUTION**

#### **Danger of injury and damage to system components by evaluation of faulty control commands!**

If a PLC is used as the master unit, the exchange of data can lead to inconsistent transmission data as a result of fieldbus and PLC cycles not operating synchronously.

- Please observe the notes concerning the operation using PLC.

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

Safe 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 transmission data is copied from the memory for the process image to the memory of the fieldbus master. This copy process should ensure that no inconsistent data are created on the fieldbus.

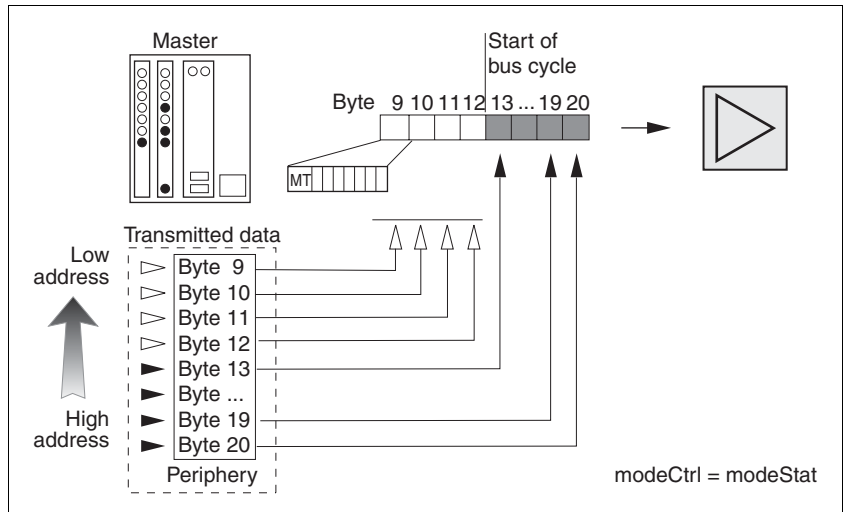


Figure 6.8 Secure data consistency, byte 9 (Bit 7) is copied last

*Data exchange via the peripheral memory*

Data consistency during data exchange is assured by the peripheral memory if MT (modeCtrl, Bit 7) is entered last. The controller ignores the transmitted data provided this bit is equal to MT (modeStat, Bit 7).

The following example shows the problems of non-assured data consistency:

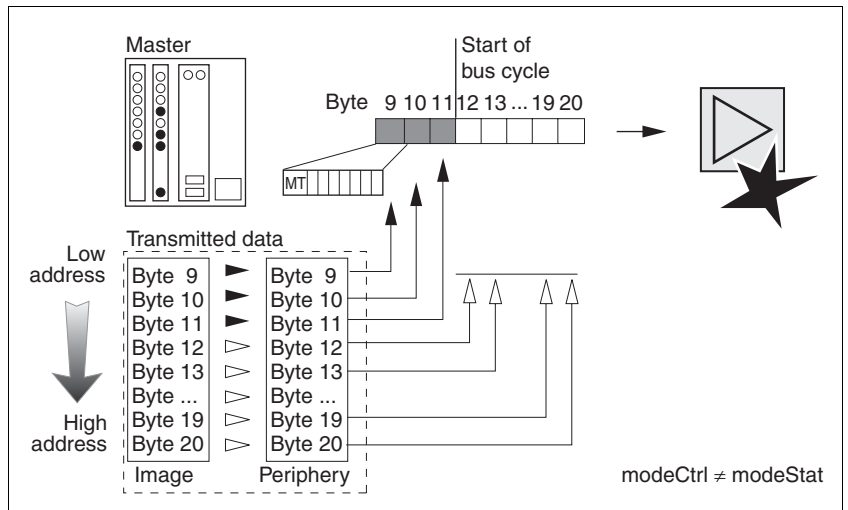


Figure 6.9 Non-assured data consistency: undefined movement!!

*Data exchange via process image memory*

The data consistency during data exchange via the process image memory is only assured 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 of a low to a high address.

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

6.5.1 Response with special master commands

*Safe state with master "Clear" command* The response when receiving a Global Control Command from a PLC as Profibus master may be configured. If a "Clear" command is sent from the master, a class 2 error or "no response" may be configured.

Group.Name Index:Sub-index dec. (hex.)	Meaning Bit assignment	Data type range (dec.)	Unit Default (dec.)	R/W/per. Info
Profibus.SafeState 24:5 (18:05 <sub>h</sub> )	Response to safe status Response of drive in status 'Clear' of the ProfibusDP master and response with watchdog triggering.  0 = no reaction 1 = error of class 2 , drive goes to FAULT status if power amplifier was enabled.	UINT32 0..1	- 1	R/W/per

## 7 Examples

### 7.1 Overview of examples

The program examples demonstrate practical applications for use on networks. There are generally two access methods over the Profibus field bus: via the parameter channel and the process data channel.

*Use of the parameter channel*

An access is always a write or read access on one single parameter. The available parameters are described in the product manual. In this chapter the use over the parameter channel is described for only a few parameters, because this type of communication can be used in unified fashion for all available user parameters and is always structured very similarly.

*Use of the process data channel*

The process data channel is recommended for the actual positioning mode, because the information is transferred much more effectively here. Various practical examples for application of the protocols supported by the drive system are shown and the general procedure is described.

*Structure of the examples*

The following is shown in the examples:

- description of task
- initial conditions
- Required commands in the transmitted data frame
- Response in the received data frame
- Possible restrictions for command execution.

You should be aware of the following to be able to reproduce the examples:

- Operating concept and functional scope of the drive system. For more information see your product manual.
- Fieldbus protocol and connection to the master
- Scope of function of the fieldbus profile.

*Product manual*

The examples are designed as a supplement to the function descriptions in the product manual. The basic functions of the operating modes and functions are described there in detail.

You will also find all parameters for the operating modes and functions and also the number format of the parameter values.

## 7.2 Use of the parameter channel

### 7.2.1 Write parameter

*Task* The parameter `Motion.acc`, 29:26 (acceleration) must be set to the value 10,000.

Index and subindex must be converted to hexadecimal format for this purpose:

- Index: 29 = 00 1D<sub>h</sub>
- Subindex: 26 = 1A<sub>h</sub>
- Value: 10000 = 00002710<sub>h</sub>

The value 30<sub>h</sub> must be input as PKE (parameter identification), because the parameter has a 32-bit data type.

*Transmitted data*

Parameter	PKE, 1st byte Job identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Tx 001D <sub>h</sub> :1A <sub>h</sub> Motion.acc	30 <sub>h</sub>	1A <sub>h</sub>	001D <sub>h</sub>	0000 2710 <sub>h</sub>	Set the acceleration to 10000 rpm*s = 2710 <sub>h</sub> as 32-bit value

The data type of the value to be written can be taken from the corresponding column in the parameter description of the product manual. With the Profibus protocol in use, 16-bit values and 32-bit values are transferred in the format "highest value bit first - lowest value bit last". The parameter identification corresponding to the data type must be input when transferring a INT16 or UINT16 value. The value must be stored in the last two data bytes and the first two data bytes must be described with zero (0).

*Received data*

Parameter	PKE, 1st byte Response identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Rx 001D <sub>h</sub> :1A <sub>h</sub> Motion.acc	20 <sub>h</sub>	1A <sub>h</sub>	001D <sub>h</sub>	xxxx xxxx	The response data have no meaning, the positive acknowledgement is signalled by PKE=20.

### 7.2.2 Read parameter

*Task* The parameter `Status.n_act`, 31:9 (actual speed) must be read. Index and subindex must be converted to hexadecimal format for this purpose:

- Index: 31 = 001F<sub>h</sub>
- Subindex: 9 = 09<sub>h</sub>

The value 10<sub>h</sub> must be input as PKE. This identifies a read request.

*Transmitted data*

Parameter	PKE, 1st byte Job identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Tx 001F <sub>h</sub> :09 <sub>h</sub> Status.n_actT	10 <sub>h</sub>	09 <sub>h</sub>	001F <sub>h</sub>	xxxx xxxx	Reading the actual speed. The data are meaningless.

The 4 data bytes are meaningless for a read request.

*Received data*

Parameter	PKE, 1st byte Response identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Rx 001F <sub>h</sub> :09 <sub>h</sub> Status.n_act	20 <sub>h</sub>	09 <sub>h</sub>	001F <sub>h</sub>	0000 03E8h	The data 000003E8 correspond to 1000 rpm; 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). However, for 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**

If a write or read command fails, the drive system responds with an error framework (Error Response). The transmitted error number shows information on the exact cause.

*Received data with error framework (Error Response)*

Parameter	PKE, 1st byte Response identifier	PKE, 2nd byte (Sdx)	Idx	Data	Description
Rx 3028 <sub>h</sub> :20 <sub>h</sub>	70 <sub>h</sub>	20 <sub>h</sub>	0028 <sub>h</sub>	0602 0000 <sub>h</sub>	Error number 06020000 <sub>h</sub> means: parameter not present in parameter directory

The example shows the response to a write or read request for a non-existent parameter 40 : 32.

The table of error numbers can be found in Section 9.2 "Error messages".

### 7.3 Operating states in the process data channel

#### ⚠ WARNING

**Danger of personal injury and damage to system parts by uncontrolled system operation!**

- Note that inputs to these parameters are executed by the drive controller immediately on receipt of the data set.
- Make sure 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.**

Drive systems detect various operating states. The different operating states are numbered from 1 to 9. The operating statuses and transition conditions are described in the product manual.

Operating status	Name	Power amplifier	Description
4	Ready To Switch On	off	passive operating status, motor without power
6	Operation Enable	on	active operating status, motor under power
7	Quick Stop active	on	error state, power amplifier remains on
9	Fault	off	error state, power amplifier switched off

Table 7.1 Important operating states

Requests for switching operating status are sent by the master in the process data channel PZD1 in the field `driveCtrl`. The slave reports the current operating status back to the master in the process data channel PZD1, field `driveStat`.

Tabelle 7.2, Seite 7-4 shows the bit assignment of the field `driveCtrl` in the transmission data in the process channel (byte 9):

bit no.	Significance	Description
0	01 <sub>h</sub>	Disable, Operation enable ⇒ Ready to switch on
1	02 <sub>h</sub>	Enable, Ready to switch on ⇒ Operation enable
2	04 <sub>h</sub>	Quick Stop, Operation enable ⇒ Quick Stop active
3	08 <sub>h</sub>	Fault Reset, Fault ⇒ Ready to switch on
4	10 <sub>h</sub>	Quick Stop Release, Quick Stop clear

Table 7.2 Transmitted data byte 9, `driveCtrl`, bit assignment



### 7.3.1 Switch power amplifier on and off

The power amplifier is switched on by the transition from operating status 4 to 6. Byte 9 contains transmission data, `driveCtrl`, the two bits `Enable` and `Disable`. One must always be set to 1 and the other to 0.

#### *Switch on power amplifier*

Condition: drive system is in operating status 4.

A 0>1 edge must be generated to switch on the power amplifier in `driveCtrl`, Bit 1 (Enable). This can be done by deleting Bit 0 (Disable) and setting Bit 1. The master then waits until the drive system reports operating status 6.

#### *Example:*

		Master		Slave
Disable is requested	Transmitted data	»	<code>driveCtrl 01<sub>h</sub></code>	»
Drive system reports operating status 4	Received data	«	<code>driveStat xxx4<sub>h</sub></code>	«
Request Enable	Transmitted data	»	<code>driveCtrl 02<sub>h</sub></code>	»
Drive system reports operating status 5	Received data	«	<code>driveStat xxx5<sub>h</sub></code>	«
Request Enable	Transmitted data	»	<code>driveCtrl 02<sub>h</sub></code>	»
Drive system reports operating status 6	Received data	«	<code>driveStat xxx6<sub>h</sub></code>	«

Table 7.3 Switch on power amplifier

#### *Switch off power amplifier*

Condition: drive system is in operating status 6 or 7.

A 0>1 edge must be generated to switch off the power amplifier in `driveCtrl`, Bit 0 (Disable). This can be done by setting Bit 0 (Disable) and deleting Bit 1 (Enable). The drive system then switches to operating status 4.

#### *Example:*

		Master		Slave
Enable is requested	Transmitted data	»	<code>driveCtrl 02<sub>h</sub></code>	»
Drive system reports operating status 6	Received data	«	<code>driveStat xxx6<sub>h</sub></code>	«
Request disable	Transmitted data	»	<code>driveCtrl 01<sub>h</sub></code>	»
Drive system reports operating status 4	Received data	«	<code>driveStat xxx4<sub>h</sub></code>	«

Table 7.4 Switch off power amplifier

### 7.3.2 Trigger Quick Stop

A current travel command can be interrupted by the fieldbus at any time with the `QuickStop` command. This is triggered by a 0>1 edge in `driveCtrl`, Bit 2. When switching to operating status 7 (Quick Stop) the drive system brakes with the specified emergency stop ramp and comes to a standstill.

Operating status 6 must be reached first to start a new travel command. Run one of the following steps to do this:

- Fault Reset: 0>1 edge in `driveCtrl`, Bit 3
- Quick Stop release: 0>1 edge in `driveCtrl`, Bit 4

Example:

		Master		Slave
Enable is requested	Transmitted data	»	driveCtrl 02 <sub>h</sub>	»
Drive system reports operating status 6	Received data	«	driveStat xxx6 <sub>h</sub>	«
Quick Stop and Request enable	Transmitted data	»	driveCtrl 06 <sub>h</sub>	»
Drive system reports operating status 7	Received data	«	driveStat xxx7 <sub>h</sub>	«
Wait until drive system stops and system should continue running				
Drive system reports operating status 7	Received data	«	driveStat xxx7 <sub>h</sub>	«
Request Quick Stop release and enable	Transmitted data	»	driveCtrl 12 <sub>h</sub>	»
Drive system reports operating status 6	Received data	«	driveStat xxx6 <sub>h</sub>	«
Cancel Quick Stop release	Transmitted data	»	driveCtrl 02 <sub>h</sub>	»
Drive system reports operating status 6	Received data	«	driveStat xxx6 <sub>h</sub>	«

Table 7.5 Triggering Quick Stop

### 7.3.3 Fault reset

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

After correction of the error the error status can be reset by running a fault reset (0>1 edge in `driveCtrl`, Bit 3).

If the operating status was 7, it switches to operating status 6 after the fault reset.

If the operating status was 9, it switches to operating status 4 after the fault reset. Then a 0>1 edge in `driveCtrl`, Bit 1 (Enable) must be sent to switch on the power amplifier again.

Example:

		Master		Slave
Request Enable	Transmitted data	»	driveCtrl 02 <sub>h</sub>	»
Slave reports operating status 9 (Fault)	Received data	«	driveStat xxx9 <sub>h</sub>	«
Correcting error				
Request Fault Reset	Transmitted data	»	driveCtrl 08 <sub>h</sub>	»
Slave reports operating status 4	Received data	«	driveStat xxx4 <sub>h</sub>	«
Request Enable	Transmitted data	»	driveCtrl 02 <sub>h</sub>	»
Slave reports operating status 5	Received data	«	driveStat xxx5 <sub>h</sub>	«
Request Enable	Transmitted data	»	driveCtrl 02 <sub>h</sub>	»
Slave reports operating status 6	Received data	«	driveStat xxx6 <sub>h</sub>	«

Table 7.6 Fault reset

Note: In this example the master deletes the Bit 1 (Enable) during the fault reset to be able to run implicitly a 0>1 edge at Bit 1. Then it switches back to operating status 6.

## 7.4 Operating modes in the process data channel

- Transmitted data* You can start movement commands with the transmission data and change them during the process.
- The following fields are available in the process data channel:
- PZD1: `modeCtrl`, start and change operating mode
  - PZD2: `ref_16`, e.g. `vel`, set speed, depending on operating mode
  - PZD3+4: `ref_32`, e.g. setpoint position, depending on operating mode
  - PZD5+6: mapped value
- The default value of these fields is only imported if `modeCtrl`, Bit 7 (`ModeToggle`) has been changed.
- You must always proceed as follows to transfer values:
- ▶ Input the desired operating mode and the associated default values in the fields `modeCtrl`, PZD2...6.
  - ▶ "Toggle"`modeCtrl`, Bit 7 (`ModeToggle`)
- This is a method of always avoiding consistency problems within the transmitted data.
- Received data* Travel commands are monitored with the aid of the received data in the process data channel.
- The following fields are available in the process data channel:
- PZD1: `modeStat`, for handshake purposes
  - PZD2: `driveStat`, reports movement status and errors and I/O signals
  - PZD3+PZD4: "32-bit actual position", `p_act`, actual position
  - PZD5+PZD6: can be configured (mapped) but, excluding exceptions, do not show any time consistency with PZD1...4.
- Mapping* Parameter values may be mapped in the PZD5 + PZD6 for transmit data as well as received data, see Section , page onwards.
- Mode Toggle* Both transmit and receive protocol have the bit `Mode-Toggle`. The master sets this bit and the drive system reflects it in the receive protocol. The master uses this procedure to detect whether the data sent by the slave are current.
- Example* The master device starts a positioning movement that will only take a very short time. The master waits for the end of the positioning by checking the receive protocol for bit `x_end = 1` (positioning end).
- The master may receive data from the slave that still come from the time before the start of positioning. These data also contain `x_end = 1`. Now the master detects that the data are old because the included bit `Mode-Toggle` does not match that of the positioning job.
- In general, the master should only evaluate data in which the received bit `ModeToggle` is identical with the last bit sent by the master.
- Acceleration* Before a positioning you can first set the desired acceleration by mapping the acceleration to PZD5 and PZD6 or by using the parameter

channel (parameter `Motion.acc`, 29:26.). Note that the acceleration can only be changed with the drive system at a standstill.

*Assumptions* The examples are based on the following assumptions:

- Operating status 6 (Operation Enable)
- Homing not run (bit `ref_ok` = 0)
- `p_act` = 0 (actual position, motor)
- Transmitted data `PZD1:modeCtrl`, Bit 7 = 0(`ModeToggle`)

7.4.1 Absolute positioning

**⚠ WARNING**

**Danger of personal injury and damage to system parts by uncontrolled system operation!**

- Note that inputs to these parameters are executed by the drive controller immediately on receipt of the data set.
- Make sure 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 setting must be made in the transmitted log:

- ▶ Enter the set speed in `PZD2 ref_16` (= `vel`) and the target position in `PZD3` and `PZD4 ref_32` (= `Pos1+Pos2`).
- ▶ Input in the field `modeCtrl` operating mode `03h` (profile position mode, absolute positioning).
- ▶ "Toggle"`modeCtrl`, Bit 7 to import the data.

*Example 1:* Absolute positioning at position 100,000 (`0001 86A0h`) at a set speed of 1000 rpm (`03E8h`)

		Master				Slave	
Trigger positioning	Transmitted data	»	<code>driveCtrl</code> <code>02<sub>h</sub></code>	<code>modeCtrl</code> <code>83<sub>h</sub></code>	<code>ref_16</code> <code>03E8<sub>h</sub></code>	<code>ref_32</code> <code>0001 86A0<sub>h</sub></code>	»
Positioning running x <code>_err = 0, x_end = 0</code>	Received data	«	<code>driveStat</code> <code>0006<sub>h</sub></code>	<code>modeStat</code> <code>83<sub>h</sub></code>	32_bit act. pos. <code>xxxx xxxx<sub>h</sub></code>		«
Trigger positioning	Transmitted data	»	<code>driveCtrl</code> <code>02<sub>h</sub></code>	<code>modeCtrl</code> <code>83<sub>h</sub></code>	<code>ref_16</code> <code>03E8<sub>h</sub></code>	<code>ref_32</code> <code>0001 86A0<sub>h</sub></code>	»
Positioning complete x <code>_err = 0, x_end = 1, x_info = 1</code>	Received data	«	<code>driveStat</code> <code>6006<sub>h</sub></code>	<code>modeStat</code> <code>83<sub>h</sub></code>	32-bit act. pos. <code>0001 86A0<sub>h</sub></code>		«

Table 7.7 Absolute positioning at constant set speed

Note: the data frame "Positioning running" can also be transmitted several times; in each case the current position is in the "32-bit actual position" field.

*Example 2:* As in example 1, except that the set speed is changed to 2000 rpm (`07D0h`) during the movement.

		Master				Slave	
Trigger positioning	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 83 <sub>h</sub>	ref_16 03E8 <sub>h</sub>	ref_32 0001 86A0 <sub>h</sub>	»
Positioning running x_err = 0, x_end = 0	Received data	«	driveStat 0006 <sub>h</sub>	modeStat 83 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«
Change set speed	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 03 <sub>h</sub>	ref_16 07D0 <sub>h</sub>	ref_32 0001 86A0 <sub>h</sub>	»
Positioning running x_err = 0, x_end = 0	Received data	«	driveStat 0006 <sub>h</sub>	modeStat 03 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«
Change set speed	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 03 <sub>h</sub>	ref_16 07D0 <sub>h</sub>	ref_32 0001 86A0 <sub>h</sub>	»
Positioning finished x_err=0, x_end = 1, x_info = 1	Received data	«	driveStat 6006 <sub>h</sub>	modeStat 03 <sub>h</sub>		32-bit act. pos. 0001 86A0 <sub>h</sub>	«

Table 7.8 Absolute positioning with change of set speed

Note: The data frame "positioning running" can also be sent multiple times. In each case, the actual position is in the "32-bit actual position" field. When the set speed is changed the same target position is sent, because this does not change in this example.

### 7.4.2 Relative positioning

<b>⚠ WARNING</b>
<b>Danger of personal injury and damage to system parts by uncontrolled system operation!</b>
<ul style="list-style-type: none"> <li>Note that inputs to these parameters are executed by the drive controller immediately on receipt of the data set.</li> <li>Make sure that the system is free and ready for movement before changing these parameters</li> </ul>
<b>Failure to follow these instructions can result in death, serious injury or equipment damage.</b>

A relative positioning is run similarly to the absolute positioning. You only need to input in field `modeCtrl` the value 13<sub>h</sub> (profile position mode, relative positioning). It is also important to ensure that multiple target positions transferred in succession are added.

*Example:* Relative positioning at 100,000 (000186A0<sub>h</sub>) increments at a set speed of 1000 rpm (03E8<sub>h</sub>)  
During the movement the speed must be changed to 2000 rpm (07D0<sub>h</sub>).

		Master				Slave	
Trigger positioning	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 93 <sub>h</sub>	ref_16 03E8 <sub>h</sub>	ref_32 0001 86A0 <sub>h</sub>	»
Positioning running: x_err = 0, x_end = 0	Received data	«	driveStat 0006 <sub>h</sub>	modeStat 83 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«
Change set speed Send relative position 0	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 13 <sub>h</sub>	ref_16 07D0 <sub>h</sub>	ref_32 0000 0000 <sub>h</sub>	»
Positioning running, x_err = 0, x_end = 0	Received data	«	driveStat 0006 <sub>h</sub>	modeStat 03 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«
Change set speed Send relative position 0	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 13 <sub>h</sub>	ref_16 07D0 <sub>h</sub>	ref_32 0000 0000 <sub>h</sub>	»
Positioning finished x_err = 0, x_end = 1, x_info = 1	Received data	«	driveStat 6006 <sub>h</sub>	modeStat 03 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«

Table 7.9 Profile position mode, relative positioning with change of set speed

Comments: the data frame "Positioning running" can also be transmitted several times; in each case the current position is in the "32-bit actual position" field. When the set 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

<b>⚠ WARNING</b>
<b>Danger of personal injury and damage to system parts by uncontrolled system operation!</b>
<ul style="list-style-type: none"> <li>• Note that inputs to these parameters are executed by the drive controller immediately on receipt of the data set.</li> <li>• Make sure that the system is free and ready for movement before changing these parameters</li> </ul>
<b>Failure to follow these instructions can result in death, serious injury or equipment damage.</b>

In profile velocity a set speed is specified and a movement is initiated with no defined finishing point.

You must make the following settings in the transmitted log to start speed mode or to change the set speed while speed mode is running:

- ▶ Enter the set speed in PZD2, ref\_16( vel,) (ref\_32 (Pos1 + Pos2) is of no significance here)
- ▶ In modeCtrl enter operating mode 04<sub>h</sub> (profile velocity)
- ▶ Switch modeCtrl, Bit 7 to import the data.

*Example* A profile velocity is started at a set speed of 1000 rpm (03E8<sub>h</sub>) (ref\_16). The set speed is changed to 2000 rpm (07D0<sub>h</sub>) during the movement. The profile velocity is ended by transfer of the set speed 0 and standstill is waited for.

		Master				Slave	
Start profile velocity at 1000 rpm	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 84 <sub>h</sub>	ref_16 03E8 <sub>h</sub>	ref_32 xxxxxxx <sub>h</sub>	»
Drive system accelerates xerr=0, xend=0, xinfo=0	Received data	«	driveStat 0006 <sub>h</sub>	modeStat 84 <sub>h</sub>		32-bit act. pos. xxxx xxx <sub>h</sub>	«
Profile velocity at 1000 rpm	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 84 <sub>h</sub>	ref_16 03E8 <sub>h</sub>	ref_32 xxxx xxx <sub>h</sub>	»
Set speed reached xerr=0, xend=0, xinfo=1	Received data	«	driveStat 2006 <sub>h</sub>	modeStat 84 <sub>h</sub>		32-bit act. pos. xxxx xxx <sub>h</sub>	«
Change speed to 2000 rpm	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 04 <sub>h</sub>	ref_16 07D0 <sub>h</sub>	ref_32 xxxx xxx <sub>h</sub>	»
Drive system accelerates xerr=0, xend=0, xinfo=0	Received data	«	driveStat 0006 <sub>h</sub>	modeStat 04 <sub>h</sub>		32-bit act. pos. xxxx xxx <sub>h</sub>	«
Speed at 2000 rpm	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 04 <sub>h</sub>	ref_16 07D0 <sub>h</sub>	ref_32 xxxx xxx <sub>h</sub>	»
Set speed reached xerr=0, xend=0, xinfo=1	Received data	«	driveStat 2006 <sub>h</sub>	modeStat 04 <sub>h</sub>		32-bit act. pos. xxxx xxx <sub>h</sub>	«
Change speed to 0 rpm	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 84 <sub>h</sub>	ref_16 0000 <sub>h</sub>	ref_32 xxxx xxx <sub>h</sub>	»
Drive system decelerates xerr=0, xend=0, xinfo=0	Received data	«	driveStat 0006 <sub>h</sub>	modeStat 84 <sub>h</sub>		32-bit act. pos. xxxx xxx <sub>h</sub>	«
Change speed to 0 rpm	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 84 <sub>h</sub>	ref_16 0000 <sub>h</sub>	ref_32 xxxx xxx <sub>h</sub>	»
Profile velocity ended xerr=0, xend=1, xinfo=1	Received data	«	driveStat 6006 <sub>h</sub>	modeStat 84 <sub>h</sub>		32-bit act. pos. xxxx xxx <sub>h</sub>	«

Table 7.10 Profile velocity

Note: the current position in increments is in the "32-bit actual position" field of the received protocol.

#### 7.4.4 Homing by dimension setting

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

You must make the following settings for dimension settings in the transmitted log:

- Enter the new position in `ref_32`. (PZD2 (`ref_16`) is of no significance here)
- In `modeCtrl` enter operating mode 02<sub>h</sub> (homing, set dimensions)
- Switch `modeCtrl`, Bit 7 to import the data from the slave.

*Example:* The motor is at position -100000 (FFFE7960<sub>h</sub>) (`ref_32`). Position 200000 is assigned to the motor (00030D40<sub>h</sub>).

		Master			Slave		
Drive system reports position 100000	Received data	«	driveStat xxxh	modeStat xxh		32-bit act. pos. FFFE 7960h	«
Dimension setting at 200000	Transmitted data	»	driveCtrl 02h	modeCtrl 82h	ref_16 xxxh	ref_32 0003 0D40h	»
Position applied x_err = 0, x_end = 1, x_info = 0	Received data	«	driveStat 4006h	modeStat A2h		32-bit act. pos. 0003 0D40h	«

Table 7.11 Dimension setting

7.4.5 Reference movement

**⚠ WARNING**

**Danger of personal injury and damage to system parts by uncontrolled system operation!**

- Note that inputs to these parameters are executed by the drive controller immediately on receipt of the data set.
- Make sure 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 or reference switch is approached and then a new value is assigned to this position.

Before starting a reference movement the parameters must be set appropriately to the requests over the parameter channel. See the product manual for more information on parameterisation and on running a reference movement.

To start a reference movement the following settings must be made in the transmitted log:

- Enter the type of reference movement in PZD2 (ref\_16) (PZD3+PZD4 (ref\_32) is of no significance here).  
The types of reference movement are described in the product manual.
- In modeCtrl enter operating mode 12h (homing, reference movement)
- Switch modeCtrl, Bit 7 to import the data from the slave.

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



		Master				Slave	
Trigger reference movement	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 92 <sub>h</sub>	ref_16 0002 <sub>h</sub>	ref_32 xxxx xxxx <sub>h</sub>	»
Reference movement running xerr=0, xend=0	Received data	«	driveStat 0006 <sub>h</sub>	modeStat80 2 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«
Reference movement	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 92 <sub>h</sub>	ref_16 0002 <sub>h</sub>	ref32 xxxx xxxx <sub>h</sub>	»
Reference movement ended, xerr=0, xend=1	Received data	«	driveStat 4006 <sub>h</sub>	modeStat A2 <sub>h</sub>		32_bit act. pos. 0000 0000 <sub>h</sub>	«

Table 7.12 Reference movement

## 7.5 Error signalling in process data channel

### 7.5.1 Synchronous errors

If request of an operating mode sent via the transmission protocol cannot be processed, the slave rejects the process and sets modeStat, Bit 6 (ModeError) in the receive protocol. This does not interrupt the current process. To find the cause of the error the master can read the error number from the parameter Status.modeError, 30:11 by accessing the parameter channel. See the product manual for a list of the error numbers.

The error display is reset when the next valid data protocol is transmitted.

Group.Name Index:Sub-index dec. (hex.)	Meaning Bit assignment	Data type range (dec.)	Unit Default (dec.)	R/W/per. Info
Status.ModeError 30:11 (1E:0B <sub>h</sub> )	Manufacturer-specific error code that results in setting the ModeError flag In general this is an error that was triggered by starting an operating mode.	UINT16	- 0	R/-/-

*Example* The drive system is in profile velocity.  
An attempt is made to run a dimension setting.

		Master				Slave	
Profile velocity, x_end = 0	Received data	«	driveStat 0006 <sub>h</sub>	modeStat 04 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«
Request: Dimension setting to 0	Transmitted data	»	driveCtrl 02 <sub>h</sub>	modeCtrl 82 <sub>h</sub>	ref_16 xxxx <sub>h</sub>	ref_32 0000 0000 <sub>h</sub>	»
Request rejected, ModeError = 1	Received data	«	driveStat 0006 <sub>h</sub>	modeStat C4 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«

Table 7.13 Synchronous error, Invalid request of an operating mode

Note: when the set dimensions request is rejected the drive system continues unchanged in profile velocity.

009844113193, V1.03, 04.2006

7.5.2 Asynchronous errors

Asynchronous errors are triggered by internal monitoring (temperature, for example) or external monitoring (limit switch, for example). If an asynchronous error occurs, the drive system responds by braking or by switching off the power amplifier.

Asynchronous errors are displayed as follows:

- Switch to operating status 7 (Quick Stop) or operating status 9 (Fault).  
The switch is displayed in the receive protocol `driveStat`, Bits 0..3.
- Setting of `driveStat`, Bit 5 (malfunction due to internal monitoring), `driveStat`, Bit 6 (malfunction due to external monitoring), or `driveStat`, bit 7 (warning) and Bit 15, `x_err` (error status during processing)
- In the event of an error message by the internal monitoring: Set the bit corresponding to the malfunction in parameter `Status.FltSig_SR`, 28:18.  
In the event of an error message by the external monitoring: Set the bit corresponding to the malfunction in parameter `Status.Sign_SR`, 28:15
- An error number is also assigned to every error. In the event of an asynchronous error the corresponding error number can be read from the parameter `Status.StopFault` (32:7).

*Example:* Trigger an error message by the external monitoring: movement to the positive limit switch LIMP

		Master				Slave	
Trigger positioning	Transmitted data	»	<code>driveCtrl</code> 02 <sub>h</sub>	<code>modeCtrl</code> 03 <sub>h</sub>	<code>ref_16</code> , <code>vel</code> 03E8 <sub>h</sub>	<code>ref_32</code> 0FFF 8765 <sub>h</sub>	»
Positioning running <code>xerr=0</code> , <code>xend=0</code>	Received data	«	<code>driveStat</code> 0006 <sub>h</sub>	<code>modeStat</code> 03 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«
Positioning	Transmitted data	»	<code>driveCtrl</code> 02 <sub>h</sub>	<code>modeCtrl</code> 03 <sub>h</sub>	<code>ref_16</code> , <code>vel</code> 03E8 <sub>h</sub>	<code>ref_32</code> 0FFF 8765 <sub>h</sub>	»
Limit switch detected <code>xerr=1</code> , <code>xend=0</code>	Received data	«	<code>driveStat</code> 8047 <sub>h</sub>	<code>modeStat</code> 03 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«
Positioning	Transmitted data	»	<code>driveCtrl</code> 02 <sub>h</sub>	<code>modeCtrl</code> 03 <sub>h</sub>	<code>ref_16</code> , <code>vel</code> 03E8 <sub>h</sub>	<code>ref_32</code> 0FFF 8765 <sub>h</sub>	»
Motor stopped <code>xerr=1</code> , <code>xend=1</code>	Received data	«	<code>driveStat</code> C047 <sub>h</sub>	<code>modeStat</code> 03 <sub>h</sub>		32-bit act. pos. xxxx xxxx <sub>h</sub>	«

Table 7.14 Asynchronous errors

Note: when the limit switch is detected the motor is braked at the emergency stop ramp until it reaches standstill and the bit `x_err` is set. After standstill of the motor, bit `x_end` is set.

## 8 Service, maintenance and disposal

### 8.1 Replacing units

The response of a system should typically not change when a slave is replaced. This is ensured by import of the specified parameter values to the new slave.

Default values of parameters can be saved on the master. After start-up of the slave these values must be sent to the slave again.



*Please see the product manual for service, maintenance and disposal procedures.*



## 9 Diagnostics and troubleshooting

### 9.1 Fieldbus communication error diagnosis

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

#### *Checking connections*

If the drive system cannot be addressed over the fieldbus, first check the connections. The product manual contains the technical data and information on network and product installation.

Check the following connections:

- ▶ System power supply
- ▶ Power connections
- ▶ Fieldbus cable and wiring
- ▶ Fieldbus terminal



*If the internal terminating resistor is enabled with switch S1 (terminated), the A2/B2 output and any additional connected bus devices are automatically disconnected from the fieldbus.*

#### *Operational errors*

When a drive system with the power amplifier is switched off is functioning correctly, the LED in the connector shell flashes constantly at 0.5 Hz (1 second on, 1 second off). If this is not the case, there is an operating error in the system. See the product manual for information on the causes of errors and troubleshooting.

#### *Function test on the fieldbus*

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

In addition to the master that knows the drive system by GSD and polling, a bus monitor that as a passive device displays messages should be installed.

- ▶ Switch the supply voltage of the drive system off and on.
- ▶ Observe the network messages shortly before switching on the drive system. A bus monitor can be used to record the elapsed time between message frames and the relevant information in the message frame during recording.

#### *possible errors: Polling, parameter setting, configuration*

If the connection to a device cannot be established, check the following:

- Polling: all network devices must have an address between 1 and 126. Every network device must have a different address.
- Setting parameters: The parameterised ID number and the user parameters must match the values stored in the GSD file.
- Configuration: The data length in the input and output direction must be identical with the length agreed in the GSD file.

## 9.2 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.

### 9.2.1 Synchronous errors

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

If request of an operating mode sent via the transmission protocol cannot be processed, the slave rejects the process and sets `modeStat`, Bit 6 (`ModeError`) in the receive protocol. This does not interrupt the current process. To find the cause of the error the master can read the error number from the parameter `Status.modeError`, 30:11 by accessing the parameter channel. See the product manual for a list of the error numbers.

The error display is reset when the next valid data protocol is transmitted.

Group.Name Index:Sub-index dec. (hex.)	Meaning Bit assignment	Data type range (dec.)	Unit Default (dec.)	R/W/per. Info
Status.ModeError 30:11 (1E:0B <sub>h</sub> )	Manufacturer-specific error code that results in setting the ModeError flag In general this is an error that was triggered by starting an operating mode.	UINT16	- 0	R/-/-

#### Error message in parameter channel

The error message is output as a response to a faulty parameter transmission. The cause of the error is output in the PWE as `ErrorCode` in bytes 5...8.

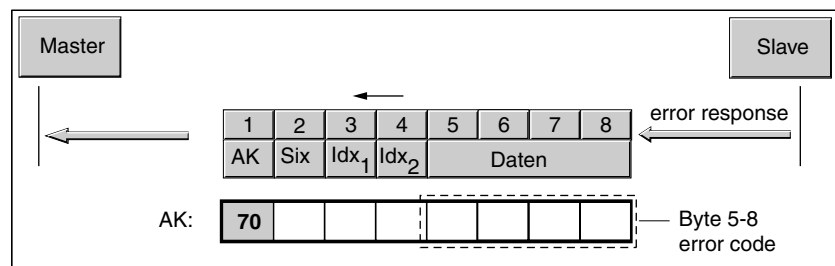


Figure 9.1 Error message in parameter channel

#### Causes of a synchronous error

Possible causes of a synchronous error are:

- Error while executing an action or control command.
- Parameter value outside the permissible value range
- illegal action or control command during a running process
- Access to unknown parameter (Index/Subindex)

The table on page 9-3 shows all error messages that can occur during data exchange with the drive system.

Error code	Description
0504 0001 <sub>h</sub>	Job identification not correct or unknown
0601 0002 <sub>h</sub>	read parameter (ro), no write access allowed
0602 0000 <sub>h</sub>	parameter not present in parameter directory
0607 0010 <sub>h</sub>	Data type and parameter length do not match
0609 0011 <sub>h</sub>	Subindex not supported
0609 0030 <sub>h</sub>	Parameter value too large or too small (only relevant for write access)
0800 xxxx <sub>h</sub>	manufacturer-specific error, xxxx corresponds to the error number of the drive system. The error number can be found in the product manual error number table

Table 9.1 Error codes

## 9.2.2 Asynchronous errors

Asynchronous errors are triggered by internal monitoring (temperature, for example) or external monitoring (limit switch, for example). If an asynchronous error occurs, the drive system responds by braking or by switching off the power amplifier.

Asynchronous errors are displayed as follows:

- Switch to operating status 7 (Quick Stop) or operating status 9 (Fault).  
The switch is displayed in the receive protocol `inDriveStat`, Bits 0..3.
- Setting of `driveStat`, Bit 5 (malfunction due to internal monitoring), `driveStat`, Bit 6 (malfunction due to external monitoring), or `driveStat`, bit 7 (warning) and Bit 15, `x_err` (error status during processing)

The error bits have the following meaning:

- Bit 5  
Error message generated by an internal monitoring signal (such as overheating)  
The error message is bit coded and entered in parameter `Status.FltSig_SR`, 28:18.
- Bit 6  
Error message generated by an external monitoring signal (such as interruption of movement by limit switch).  
The exact cause is bit coded and entered in parameter `Status.Sign_SR`, 28:15 entered in bit code.
- Bit 7  
Warning message (such as overheating warning)  
The error message is bit coded and entered in parameter `Status.WarnSig`, 28:10.

The last cause of interruption is entered as an error number in parameter `Status.StopFault`, 32:7 entered as error number.

The error numbers and their meanings are listed in the "Diagnostics and Troubleshooting" section of the product manual.

The meaning of the error message can be determined via the corresponding parameters.

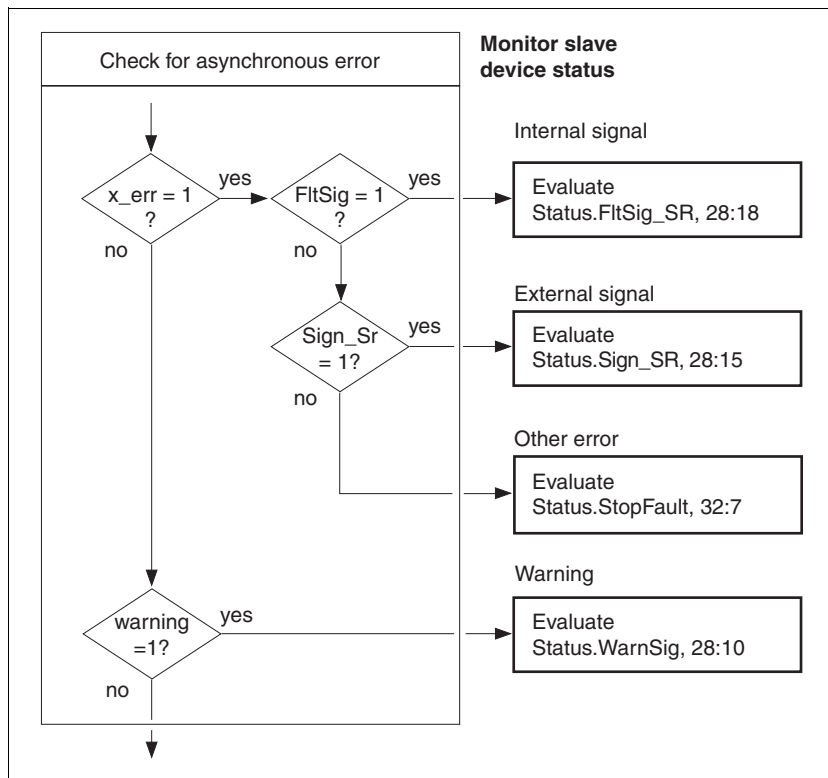


Figure 9.2 Evaluating asynchronous errors

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

### 9.2.3 Errors during operating mode control

Travel commands can be triggered and modified via the process log. If the request cannot be processed, an error bit is set in the received data.



## 10 Glossary

### 10.1 Terms and Abbreviations

<i>Address</i>	Memory location which can be accessed by its unique number. See also Slave address.
<i>AK</i>	Job/answer identification
<i>Broadcast</i>	Type of data transmission in the network, one device sends a message to all devices on the network
<i>Default value</i>	Factory settings.
<i>Direction of rotation</i>	Rotation of the motor shaft in a positive or negative direction of rotation. A positive direction of rotation is defined as the motor shaft rotating clockwise as the observer faces the end of the protruding shaft.
<i>DP</i>	<b>D</b> ecentralized <b>P</b> eriphery
<i>EMC</i>	Electromagnetic compatibility
<i>Error class</i>	Classification of operational faults into groups corresponding to the error responses
<i>FMS</i>	Fieldbus-Message-Specification
<i>GSD file</i>	The specific characteristics of a Profibus device type are described in the device master data file (GSD file). This file is supplied with the device by the manufacturer, and must be read by the network configuration program..
<i>I/O</i>	Inputs/Outputs
<i>Idx</i>	Index value of a parameter
<i>LED</i>	Light-Emitting Diode
<i>Limit switch</i>	Switch that signals an overrun of the permissible travel range.
<i>LWL</i>	Optic fiber
<i>Master</i>	Active bus user that controls the data traffic in the network.
<i>MT</i>	<b>M</b> ode <b>T</b> oggle, bit change 0 » 1 or 1 » 0
<i>Parameter</i>	Device functions and values that can be set and called by the user.
<i>PKE</i>	Parameter code
<i>PNO</i>	Profibus User Organisation
<i>Profibus</i>	Standardised open fieldbus compliant with EN 50254-2 over which drives and other devices from different manufacturers communicate with one another.
<i>PWE</i>	Parameter value
<i>PZD</i>	Process data
<i>Quick Stop</i>	Quick stop, function used to provide quick braking of the motor via a command or in the event of a fault.
<i>Six</i>	Subindex value of a parameter

<i>Slave</i>	Passive bus user that receives control commands and sends data to the master.
<i>Slave address</i>	Direct communication between master and slave devices is only possible after assignment of addresses.
<i>PLC</i>	Programmable Logic Controller
<i>Toggle</i>	see MT, ModeToggle
<i>Watchdog</i>	Equipment that monitors cyclic basic functions in the drive system. Power amplifier and outputs are switched off in the event of error.

## 11 Index

### A

Abbreviations 10-1

### B

Big-endian format 6-3

bit 6-2

byte 6-2

### C

Commissioning 5-1

Commissioning, requirements 5-1

### D

data frame 6-2

Data structure 6-2

Device master data file 3-2

Diagnostics 9-1

Directives and standards 1-1

Disposal 8-1

Documentation and literature references 1-1

double word 6-2

driveStat 6-13

### E

EMC 4-1

Equipotential bonding conductors 4-1

Error

codes 9-3

Examples 7-1

### F

Function

Profibus fieldbus interface 4-2

Function tests 5-2

### G

Glossary 10-1

### I

Identity number 3-2

Index 6-4, 6-5

Intended use 2-1

Introduction 1-1

### L

Literature 1-1

**M**

Maintenance 8-1  
Mapping 6-11, 7-7  
Master 3-1  
Master-slave relationship 3-2  
Mode Toggle 7-7  
modeStat 6-13

**N**

Network operation 5-2

**O**

Operation 6-1

**P**

Parameter channel 6-4  
Parameter identifier 6-5  
Peripheral memory 6-17  
PKE 6-5  
Pos1, Pos2 6-13  
Process data channel 6-7  
Process image 6-17  
Profibus fieldbus interface  
    function 4-2  
Profibus User Association 1-1  
PWE 6-5

**Q**

Qualifications, personnel 2-1

**R**

Real-time data exchange 6-7  
RS485 technology 3-2

**S**

Service 8-1  
Slave 3-1  
subindex 6-4  
synchronous error 7-13

**T**

Table of error codes 9-3  
Terms 10-1  
the Token-Passing method 3-2  
Troubleshooting 9-1

**W**

word 6-2