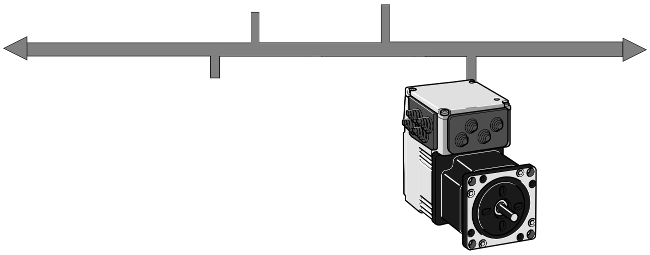
BERGER LAHR

Technical Documentation



Field bus protocol for the IclA IFx
Intelligent Compact Drive

Profibus DP

Order no.: 0098 441 113 193

Edition: -000 09.03

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

a company of Schneider Electric

Important information

Berger Lahr drive systems are products intended for general use that conform to the state of the art in technology and are designed to eliminate dangers as much as possible. However, drives and drive controllers that are not specifically designed for safety engineering functions are general engineering equipment that is not approved for applications in which the drive functions could endanger persons. Unexpected or unbraked movements can never be completely excluded without additional safety equipment. For this reason no person should be in the danger zone of the drives unless additional suitable safety equipment is installed to prevent danger to persons. This is applicable for the machine in production operation and for all repairs and maintenance work on drives and machine. The machine must be designed to ensure personal safety. Suitable precautions must also be taken to prevent property damage.

For more information see the chapter on safety.

0098 441 113 193, -000, 09.03

Table of Contents

0098 441 113 193, -000, 09.03

5 Commissioning

6 Profibus communication

7 Examples

0098 441 113 193, -000, 09.03

8 Diagnostics and troubleshooting

9 Parameters

10 Accessories and spare parts

11 Service, maintenance and disposal

12 Supplement

13 Glossaries

14 INDEX

0098 441 113 193, -000, 09.03

0098 441 113 193, -000, 09.03

Conventions and symbols

Instructions for use

Introduction to the following steps

▶ This the 1st work step

◀ This is the response to the 1st work step

▶ This is the 2nd work step

◀ This is the response to the 2nd work step

Operating instructions consist of an introduction and the actual operational steps.

Unless otherwise specified, the individual operational steps must be executed in the specified sequence.

If there is a significant response to an operational step, this response will be described after the operational step. This enables the correct execution of the operational step to be checked.

List symbol

Note on the content of the list

- 1st list item
- 2nd list item
  - 1st list subitem
  - 2nd list subitem
- 3rd list item

The actual list, which can consist of one or two levels, follows a note on the content of the list.

The list items are sorted alphanumerically or by priority.

User notes

The user notes contain general information, not safety instructions.

This contains additional information on the current subject.

See the Safety chapter for an explanation of the safety instructions.

Parameter

Parameter are shown as follows:

`Group.Name Index:Subindex`

0098 441 113 193, -000, 09.03

1 Introduction

This manual describes the online command processing for Intelligent Compact Drives that are addressed via Profibus-DP in the field bus network.

Berger Lahr Intelligent Compact Drives are variable speed drives with integrated controller and power electronics. They are available in stepper motor and EC motor designs in various models.

IFS stepper motor

The IclA IFS compact drive consists of a permanently excited synchronous motor (stepper motor) with integrated controller and power electronics. The compact drive can optionally be fitted with a gearbox and a holding brake. Another option is an internal Hall sensor, which sends an index pulse and can be used for blocking detection.

IFE EC motor

The IclA IFE compact drive consists of an electronically commutated EC motor with integrated controller and power electronics and spur-wheel gear. The compact drive can also be supplied with a PLE planetary gearbox or without a gearbox.

1.1 Documentation and literature references

Documentation

- Data sheets for IclA in the IclA Intelligent Compact Drives catalog
- Controller manuals for IclA compact drives:
  - Intelligent Compact Drive IclA IFS6x field bus stepper motor
  - Intelligent Compact Drive IclA IFE7x field bus EC motor

Literature references

- PROFIBUS Specification (FMS,DP,PA)
  Profibus user organisation
- Popp, M: PROFIBUS-DP/DPV1
  Basics, Tips and Tricks for Users
  ISBN 3-7785-2781-9

1.2 Directives and standards

Regulations, standards

- DIN 19245, Part 1 to 3: PROFIBUS-FMS
- EN50170, field bus standard

Interest group of Profibus users

Profibus user organisation e.V. (PNO)

Interest group of Profibus users

Haid-und-Neu-Str. 7

D-76131 Karlsruhe

Profibus international in the Internet

http://www.profibus.com

0098 441 113 193, -000, 09.03

2 Safety

2.1 Qualifications of personnel

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

The technicians must also have the technical training, knowledge and experience to be able to assess the work assigned to them.

The technicians must be familiar with current standards, regulations and accident prevention regulations that must be observed when working on the drive system.

2.2 Intended use

Berger Lahr drive systems are products intended for general use that conform to the state of the art in technology and are designed to eliminate dangers as much as possible. However, drives and drive controllers that are not specifically designed for safety engineering functions are general engineering equipment that is not approved for applications in which the drive functions could endanger persons. Unexpected or unbraked movements can never be completely excluded without additional safety equipment. For this reason no person should be in the danger zone of the drives unless additional suitable safety equipment is installed to prevent danger to persons. This is applicable for the machine in production operation and for all repairs and maintenance work on drives and machine. The machine must be designed to ensure personal safety. Suitable precautions must also be taken to prevent property damage.

In the system configuration described the drive systems must only be installed in an industrial environment with a fixed connection.

The applicable safety regulations and the specified operating conditions, such as environmental conditions and specifications, must be observed.

The drive systems may be commissioned and operated only after installation in accordance with EMC requirements and the product-specific specifications.

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

Changes and modifications to the drive systems are not permitted and will render all warranties and liability null and void.

The drive system must be operated only with the specified cables and approved accessories. Use original spare parts whenever possible.

Operation of the drive systems outside the described and specified limit values is not designated use.

0098 441 113 193, -000, 09.03

2.3 Hazard categories

Safety notes and general information are indicated by hazard messages in the manual. In addition there are symbols and instructions affixed to the product that warn of possible hazards and help to operate the product safely.

Depending on the seriousness of the hazard, the messages are divided into three hazard categories.

DANGER!

DANGER indicates an imminently hazardous situation, which, if not avoided, **will result** in death, serious injury, or equipment damage.

WARNING!

WARNING indicates a potentially hazardous situation, which, if not avoided, **can result** in death, serious injury, or equipment damage.

CAUTION!

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

2.4 Safety instructions

DANGER!

Danger of injury by complex system!

When starting field bus operation the attached controllers are generally out of view of the operator and cannot be directly monitored.

- Only start the system when there are no persons within the actuation zone of the moving system components and the system can be operated safely.

0098 441 113 193, -000, 09.03

WARNING!

Danger of injury and damage to system components by loss of control!

- The system manufacturer must consider the possible errors that could occur with the signals and in particular the critical functions to ensure a safe status during and after errors. Critical functions include emergency stop and limiting end positions. Observe the accident prevention regulations
- Consideration of possible errors must include unexpected delay and failure of signals or functions
- Separate redundant controller paths must be provided for critical functions.

CAUTION!

Danger of injury and damage to system components by evaluation of faulty controller commands.

Data exchange when a PLC is used as a master device can result in inconsistent transmitted data, because the field bus and PLC cycles do not operate synchronously.

- Follow the instructions for operation with a PLC.

0098 441 113 193, -000, 09.03

3 Basics

3.1 The Profibus technology

3.1.1 Profibus transmission technology

The Profibus is a serial field bus system in which devices from different manufacturers can be networked together without special interface adaptations. Profibus is available in three versions that can handle time-critical and complex communications tasks:

- Profibus FMS
- Profibus-PA
- Profibus-DP

Profibus FMS (FMS: Field bus Message Specification) is a universal, flexible solution for communications tasks in general automation engineering and is, for example, used for communications between production cells.

Profibus-PA (PA: Process Automation) is primarily used in process engineering, such as process automation. A feature of Profibus-PA networks is the option of using sensors and actuators in explosive environments and data communications and power supply over the bus.

Profibus-DP (DP: Decentralised Peripheral) is the fast Profibus type, which is specially designed for communications in production. Characteristics are simple integration of new devices into the bus and high transmission speeds.

Drive systems with Profibus-DP are equipped for operation in Profibus-DP networks. They support various parameterising telegrams in accordance with the Profibus-DP V0 specification.

3.1.2 Network topology

A Profibus-DP network consists of one or more master devices and slave devices. All devices are linked together over the Profibus-DP network cable.

Master Masters are active bus devices that control data traffic in the network. The following are examples of master devices:

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

Slave Slaves are passive bus devices. They receive control commands and supply data to the master. The following are examples of slave devices:

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

3.1.3 Access procedures

There are two possible access procedures for the variety of network devices on the bus:

- Token-passing procedure
- Master-slave procedure

Token-passing procedure

The token-passing procedure is used between multiple master devices in a Profibus-DP network. The master devices form a logical token ring in which every master receives transmission authorisation in succession for a specified period.

Master-slave procedure

Data exchange with drive controllers is processed with the master-slave procedure. The slave device has a transmit and receive buffer through which it sends and receives data. The master reserves a memory range with transmit and receive buffer for every slave.

Data is exchanged cyclically between master and slave. The master sends command information to the slave and in the next cycle receives the data sent by the slave. The bus cycle is extended for transmission of telegram repetitions only in the event of malfunction

Both transmit and receive buffer for data exchange for the drive system are 20 bytes in size.

Drive controllers are integrated into the network as slaves, so they operate in master-slave procedure only and not with token-passing.

3.1.4 Transmission technology

Profibus-DP networks can be laid out with fibre-optic cables or with RS485 technology.

Fibre-optic technology

Transmission by fibre-optic cables is primarily of interest in environments subject to strong EMC interference and where transmission of large amounts of data over long distances is required.

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.

Drive systems operate with RS485 technology and are connected to a Profibus-DP with two-wire lines. Transitions to and from fibre-optic lines can be made with auxiliary terminal boxes.

0098 441 113 193, -000, 09.03

3.1.5 Device identification

Device master data file The specific features of a Profibus device type are described in the device master data file (GSD file). The manufacturer supplies this file with the device and it must be read by the network configuration program.

The GSD file contains all information required to operate the device on the Profibus-DP network, such as manufacturer's details and device name, supported baud rates, signal assignment of the device connectors, time intervals for monitoring periods and device-specific values for network devices such as settings for inputs and outputs. The GSD file for Berger Lahr drive solutions on the Profibus-DP network is available for download from the internet.

ID number A master identifies the device class of the connected slave device by the ID number. The ID number is a unique number assigned to a specific device class by the Profibus user organisation.

Slave address The slave address is set by a DIP switch on the drive. The master can specifically address every slave on the Profibus-DP network with this address (assigned once). A maximum of 127 devices is possible on the network. The exact setting of the address is described in the controller manual.

3.2 Field bus devices in the network

Different field bus devices from Berger Lahr can be operated on the same field bus segment. The Profibus-DP provides a unified basis for exchange of commands and data between Berger Lahr drives and other network devices.

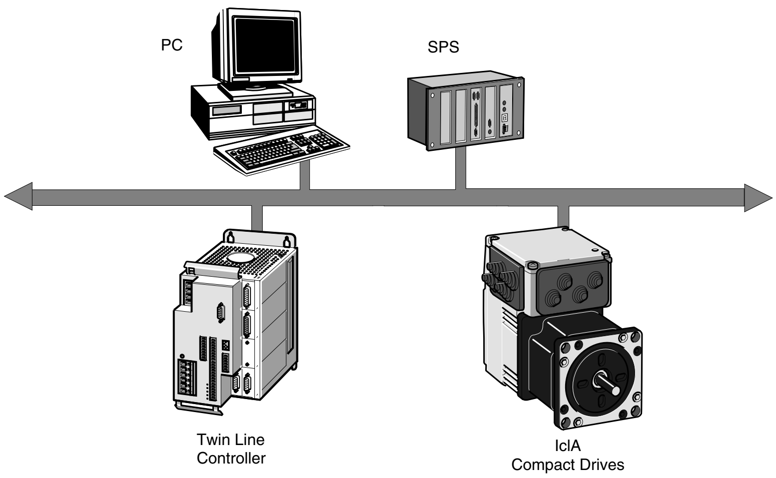


Figure 3.1 Field bus devices in the network

0098 441 113 193, -000, 09.03

3.3 Operating modes and functions

This manual only describes the protocol for field bus operation. See the chapters on "Operation" and "Parameters" in the controller manual for descriptions of the operating modes, operating functions and all parameters:

Operating modes

- Speed mode
- Point-to-point mode
- Referencing

Operating functions

- Definition of direction of rotation
- Generating movement profiles
- Quick-Stop
- Fast position capture

Setting options

The following settings can be made over the field bus:

- Read and write parameters
- Monitor inputs and outputs of 24-V signal interface
- Enable diagnostics and error monitoring functions in field bus operation

0098 441 113 193, -000, 09.03

4 Installation

WARNING!

Danger of injury and damage to system components by loss of control!

- The system manufacturer must consider the possible errors that could occur with the signals and in particular the critical functions to ensure a safe status during and after errors. Critical functions include emergency stop and limiting end positions. Observe the accident prevention regulations
- Consideration of possible errors must include unexpected decelerations and failure of signals or functions
- Separate redundant controller paths must be provided for critical functions.

4.1 Electromagnetic compatibility, EMC

EMC requirements must be taken into account when laying and connecting wire cables in an electromagnetic environment.

The following measures are necessary for trouble-free field bus operation. They are in addition to the device-specific EMC measures in the controller manual.

EMC measures	Effect
Use wiring with braided and foil shielding	Discharge of interference currents
Do not lay field bus wiring together with wiring for direct and alternating voltage over 60 V[1)]	Prevents mutual injection of interference signals
Use equipotent bonding conductors in systems with – wide-area installation – different voltage feed – networking between different buildings	Discharge of interference currents
Use fine-core equipotential bonding conductors	Also discharges high-frequency interference currents

[1)] Field bus wiring can be laid in one conduit with signal and analogue wiring.

Table 4.1 EMC measures

Equipotential bonding conductors

In digital cables the shields are connected at both ends to protect against interference. Potential differences can result in excessive currents on the shield and must be prevented by equipotential bonding conductor cables. For cables of up to 200 m in length a cross section of 16 mm^2 is sufficient, but for greater lengths a cable cross section of 20 mm^2 is required.

0098 441 113 193, -000, 09.03

4.2 Profibus DP interface

WARNING!

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

Foreign bodies, deposits or moisture can cause unexpected responses by the device.

- Make sure that no foreign material can enter.
- Do not remove the electronics housing cover. Remove the connector housing cover only.
- Check that the seals and cable glands are correctly seated.

4.2.1 Connection and cable specification

- Shielded cable
- Minimum cross section of signal wires: 0.34 mm^2
- Twisted-pair cables
- Two-way earthing of shield
- The maximum length depends on the baud rate and the signal runtimes. The higher the baud rate, the shorter the bus cable has to be.

Baud rate [kbaud]	max. cable length [m]
9,6	1200
19,2	1200
45,45	1200
93,75	1200
187,5	1000
500	400
1500	200
3000	100
6000	100
12000	100

Table 4.2 Baud rate and cable length for the Profibus DP

4.2.2 Terminating resistor

Both ends of the complete bus system must be terminated. If the compact drive is the last device on the network cable, a terminating resistor must be connected parallel to the field bus cables.

With Profibus DP the terminating resistors are integrated into compact drives and they can be activated with a DIP switch.

The illustration below shows the structure of the integrated terminating resistors.

0098 441 113 193, -000, 09.03

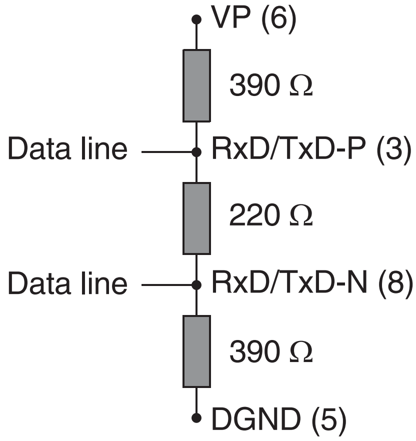


Figure 4.3 Profibus DP terminating resistors

4.2.3 Function

The compact drive can be connected as a slave to the following networks with the Profibus DP interface:

- Profibus

The compact drive receives data and commands from a higher-level device on the bus, a master device. The controller sends status information such as device status and processing status back to the master device as acknowledgement.

4.2.4 Field bus mode

Integration of a compact drive into the field bus is described in the relevant field bus manual in the chapter on `Communications on the field bus`.

4.2.5 Setting address and baud rates

Every device in the network is identified by a unique node address which can be set as desired. Only addresses 3-126 are allowed for a slave device on a Profibus network. Addresses 0-2 are reserved for master devices.

Baud rate The baud rate is detected automatically with the autobaud function.

0098 441 113 193, -000, 09.03

4.2.6 Drives with DIP switches

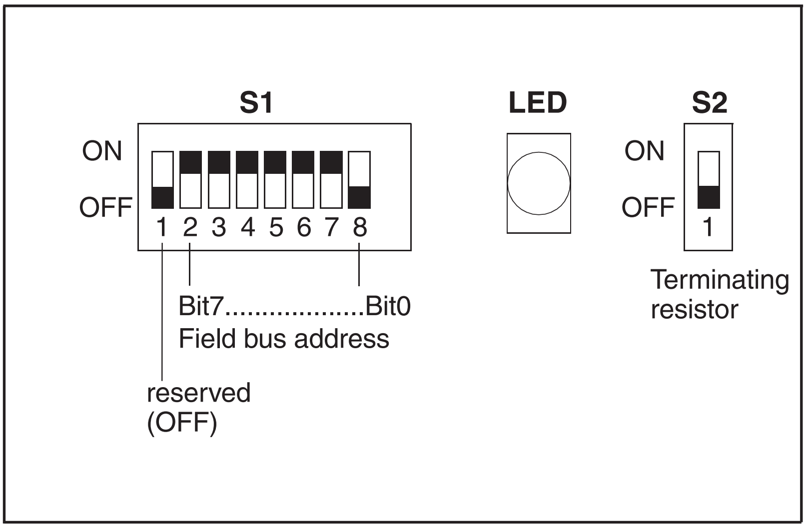


Figure 4.4 Assignment of Profibus DP DIP switches

Default settings:

- Address: 126
- Terminating resistor: OFF

LED	Display of Profibus communication to -> communication OK from -> no communication
S1	Field bus address assignment see above Valid addresses 3 to 126
S2	Terminating resistor S2.1 = on -> terminating resistor on S2.1 = off -> terminating resistor off

Table 4.5 DIP switch settings for Profibus DP

Reserved DIP switches are reserved for future upgrades and must be set to "OFF".

0098 441 113 193, -000, 09.03

4.2.7 Circuit board plug connector types

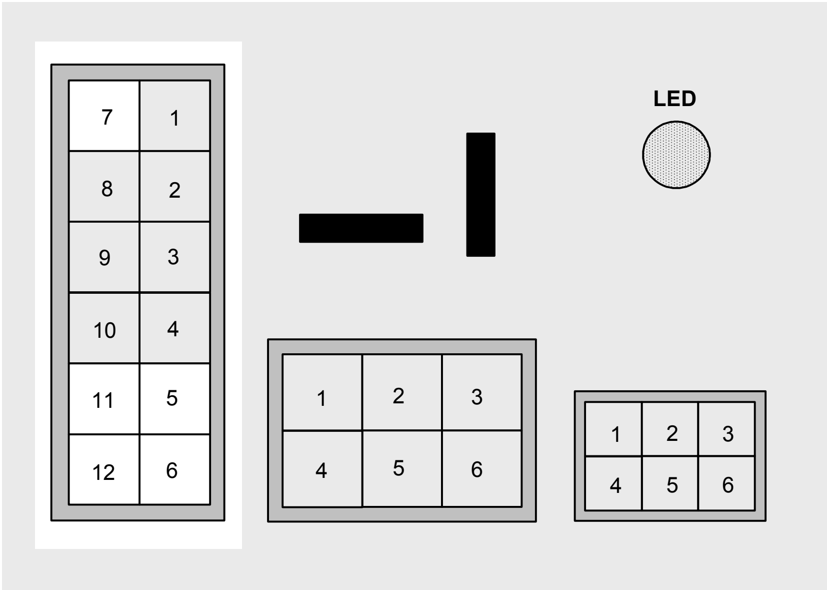


Figure 4.6 Pin assignment of the Profibus-DP field bus interface

Pin	Assignment	Signal type
5	B_LT out data line inverted	A
6	A_LT out data line	A
7	GND (optional connection only, compare earth concept)	
11	B_LT in Profibus data line inverted	E
12	A_LT in Profibus data line	E

Table 4.7 Pin assignment of the Profibus-DP field bus interface

4.2.8 Industrial plug connector types

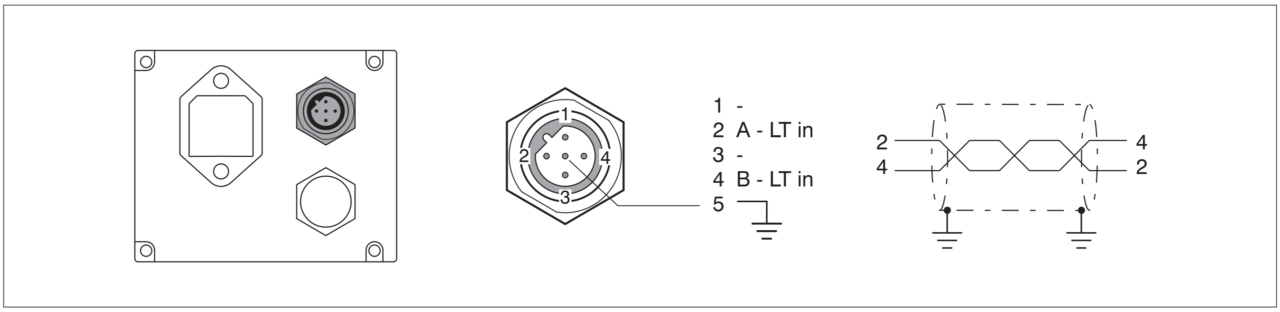


Figure 4.8 Pin assignment of the Profibus-DP field bus interface in

Pin	Assignment	Signal type
2	A_LT in Profibus data line	I
4	B_LT in Profibus data line inverted	I

0098 441 113 193, -000, 09.03

Pin	Assignment	Signal type
5	SHLD, shielded connection	

Table 4.9 Pin assignment of the Profibus-DP field bus interface in

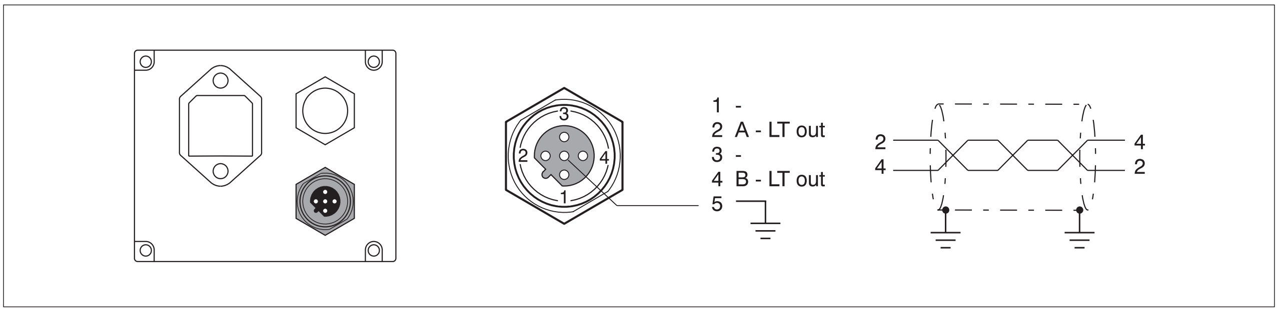


Figure 4.10 Pin assignment of the Profibus-DP field bus interface out

Pin	Assignment	Signal type
2	A_LT out Profibus data line	O
4	B_LT out Profibus data line inverted	O
5	SHLD, shielded connection	

Table 4.11 Pin assignment of the Profibus-DP field bus interface out

0098 441 113 193, -000, 09.03

5 Commissioning

DANGER!

Danger of injury by complex system!

When starting field bus operation the attached controllers are generally out of view of the operator and cannot be directly monitored.

- Only start the system when there are no persons within the actuation zone of the moving system components and the system can be operated safely.

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 controller manuals.
- 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 field bus communication: Bit 0 is far right (least significant). Bit 15 is far left (most significant).
- Check the use of the word sequence during field bus communication:
- Do not establish a field bus connection before you have understood the communications principles.

5.1 Requirements for commissioning

The drive system must be fitted with the Profibus DP field bus interface for operation on the field bus. At least the following components are required for commissioning:

- GSD file on data medium
- Manual for
  - IclA IFS 6x

    or
  - IclA IFE 7x

0098 441 113 193, -000, 09.03

- Manual for IclA IFx Profibus DP (this manual)

This information can also be found on the CD.

The appropriate connector cable is also required. A list of the order numbers of all components can be found in the `Accessories` chapter.

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

5.2 Initiating network operation

Network operation can be started from a master device. This can be a PLC or a PC with the appropriate application 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 how the system reacts to power failure.

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

- Units switched on and master device started for network operation?
- Cables mechanically sound?
- Is the LED on the field bus input on the controller on? The LED shows data traffic over the network interface.
- Address correctly set in controller?

See the controller manual for information on the cause of errors and troubleshooting.

0098 441 113 193, -000, 09.03

6 Profibus communication

6.1 Communication profile

6.1.1 Parameter channel and process data channel

With the compact drives master and slave communicate with each other analogous to the Profidrive profile V2 PPO Type 2 of the PNO user organisation in accordance with the protocols described below. The data frame is 20 bytes. The first 8 bytes are used for parameter transfer, the following 12 bytes (bytes 9-20) send the process data. They are interpreted depending on the operating mode.

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

The abbreviations used are defined as follows: PKE Parameter identification IDX Index PWE Parameter value PZD Process data

These terms are defined in detail in this chapter.

Index, Subindex

The parameters are addressed over a 16-bit long index. The individual data fields of a parameter are specified by the subindex entries. A data field consists of one or more subindex entries. Index and subindex are given in hexadecimal, which can be recognised by a subscript "$_h$". The following example shows index and subindex entries for the parameter `IO.IO_def, 0022`$_h$ for configuring the digital inputs and outputs.

Index	**Subindex**	**Object**	**Meaning**
0022_h	01_h	`IO.IO0_def`	Configuration of IO 0
0022_h	02_h	`IO.IO1_def`	Configuration of IO 1
0022_h	03_h	`IO.IO32def`	Configuration of IO 2
0022_h	04_h	`IO.IO3_def`	Configuration of IO 3

Table 6.1 Examples of index and subindex entries

There is a list of all parameters in the controller manual of your compact drive. It shows the parameters in their functional relationship to every operating mode and they are shown again in a group overview at the end of the manual. The parameters for the Profibus are described at the end of this documentation.

The number format of the parameter values in a field bus command can be found in the group overview in the 'Parameters' chapter of the controller manual.

6.1.2 Profibus DP V0 communications

Profibus DP V0 provides the basic functions of DP, including cyclic data exchange, station, module and channel-specific diagnostics and various alarm types for diagnostics and process alarm for disconnecting and connecting bus devices.

"Master-slave" relationship

The central controller (master) cyclically reads the input information from the slaves and cyclically writes the output information to the slaves. Input and output data are sent as one unit for one slave in one message cycle.

Command processing: Transmitted data and received data

The master sends a command to control the drive, to initiate a movement job, to enable operating functions or to request information from the controller. The controller executes the command and sends an acknowledgment.

Data exchange follows a fixed pattern:

- "Transmitted data" to the controller: The master saves a command in the transmitted data memory. The data is sent from there to the controller and executed.
- "Received data" from the controller: The controller acknowledges the execution status of the command in the received data. If the master receives an acknowledgment without an error message, the command was correctly executed.

The master can send new commands as soon as it has received the acknowledgment for the current command. Acknowledgment information and error information are sent bit-coded with the transmitted data.

With the cyclic field bus transmission the master automatically receives current received data from the slave at every cycle. It uses an acknowledgment mechanism to detect whether the received data contain status information from the slave or an answer to a command that was previously sent. The slave also uses the acknowledgment mechanism to detect a new command.

Commands

The master sends control and action commands with the transmitted data. After a control command, the master receives an acknowledgment indicating whether the process could be successfully executed and completed.

In the case of an action command the controller simply reports whether an action or a movement job could be successfully started. The master must then continuously monitor the completion of the processing job by evaluating the received data from the controller.

6.1.3 Data structure

In addition to command and controller information, the transmitted and received data contain administrative data for security of network operation. The administrative data are provided by the user program in the master device.

The transmitted and received data, which are exchanged in a 20-byte data frame, must be programmed and evaluated for communication with the controller in the network.

0098 441 113 193, -000, 09.03

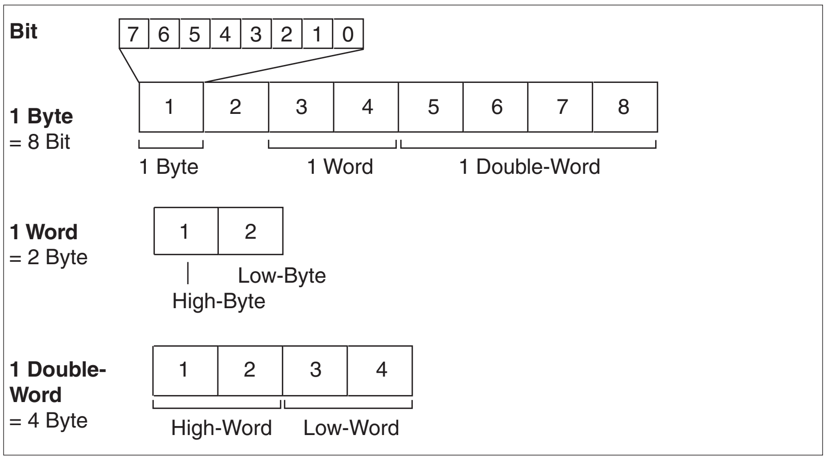


Figure 6.2 General data structure from the bit to the double word

The manual contains the data frames with transmitted and received data and all byte, word and double-word values in hexadecimal. Hexadecimal characters are identified with an "h" after the numerical value, e.g. "31_h". Note the different count method bit (0-7, right to left) and byte (1-xx, left to right). The Profibus data are sent in Motorola format, i.e. numerical values are handled by one byte as in the decimal system. Example: the index value is transmitted in bytes 3 and 4, index 21_h is therefore shown as 00_h21_h.

6.2 Parameter channel

6.2.1 Overview

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

0098 441 113 193, -000, 09.03

6.2.2 Structure of the parameter channel

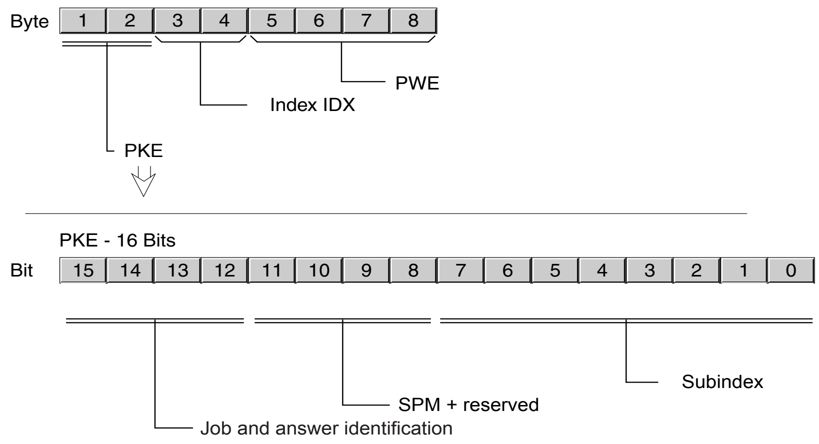


Figure 6.3 Parameter channel: Parameter detection in byte 1 and 2

The following abbreviations used for the parameter channel:

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

Byte 3+4: IDX for index of the parameter

Byte 5-8: PWE for parameter value

Parameter identification

The job or answer identification and the subindex of the parameter are in the first two bytes, the parameter identification. The index is input into bytes 3 and 4. The parameter value is in bytes 5-8 (PWE). The data are input in the Motorola format.

The job/answer identification indicates the fields of the parameter channel that must be evaluated.

Job identification	Function	Answer identification	
		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 6.4 Job and answer identification

Write jobs (change parameter value) are only executed by the slave if the value of the job identification changes from 0 to 2 or 3. Read jobs are executed so long as the value of the job identification is equal to 1. To reduce the system load, a cycle time between two read processes is defined with the parameter `PKINHIBIT`. The read job is run again on expiry of the cycle time.

0098 441 113 193, -000, 09.03

The slave uses the response identification to signal the master that the job was successfully executed (response identification positive) or that an error has occurred (response identification negative). If the response identification to a current requested job is 0, the slave has not yet completed the job. In the event of a negative response identification the error number is in bytes 5-8 (parameter value).

Basically only one job can be in process at one time. The slave sends the response until the master sends a new job. In the case of responses that contain parameter values, the slave always responds with the current value on repetition (cyclic processing).

Bits 8-11 (SPM and Reserved) must always be 0 for compact drives. Format and messages settings for these bits are described in the PROFIdrive application profile, but they are not supported by the compact drive.

Example: Error-free reading of a parameter

The program number of the Profibus software must be read in the example. The program number is stored in the device configuration range (Index $0D_h$;Subindex B_h). The master sends a read job to the slave. After processing the slave sends the requested data in bytes 5-7 (parameter value PWE). The parameter has the decimal value 825_d, corresponding to 339_h. Because this is a double word, the corresponding positive response identification must be 2.

Therefore, the master sends the following transmitted data to the slave (values that are not relevant for the example are shown by x).

Transmitted data

Index: $13=0D_h$

Subindex: $11=0B_h$

Object	PKE	Sdx	Idx	Data	Description
Tx $000D_h:09_h$	10_h	$0B_h$	$00\ 0D_h$	XX XX XX XX	Reading the program number. The data have no meaning.

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

Received data

Object	PKE	Sdx	Idx	Data	Description
Rx $001D_h:09_h$	20_h	$0B_h$	$00_h\ 0D_h$	$00\ 00\ 03_h\ 39_h$	The data 00000339 correspond to the program number.

The compact drive distinguishes between parameter values with 32-bit data and 16-bit data (described as INT32 or UINT32 and INT16 or UINT16 data types in the controller manual) by the response identification (2 or 1). For 16-bit data it is important to evaluate the last two data bytes only and to ignore the first two data bytes.

The message is available until the master sends the job identification 0 to the slave before the next job.

Example: Aberrant writing of a parameter

After the master has read the information from the above example, it must first reset the slave with the job identification “No job” (PKE:00).

0098 441 113 193, -000, 09.03

The slave is then ready to execute new jobs. The value of a non-existent parameter must be changed for the example. The value of the parameter with the index 18_h and the subindex 255_d (corresponds to FF_h) must be changed to DE_h.

Index: 24 = 00 18_h

Subindex: 255 = FF_h

Value: 222= 0000 $00DE_h$

Object	PKE	Sdx	Idx	Data	Description
Tx 0018_h:FF_h	30_h	FF_h	00 18_h	0000 $00DE_h$	Write a non-existent parameter

Because the slave cannot address the parameter, an error message is returned; the parameter value in this case is 06020000_h. Error messages in the parameter channel are designated as synchronous errors, because they are processed in the standard, cyclic data exchange.

Object	PKE	Sdx	Idx	Data	Description
Rx 0018_h:FF_h Status.n_actT	70_h	FF_h	00 18_h	0602 0000_h	Error message 0602 0000_h is returned

For information on synchronous errors see page 8-2

6.3 Process data channel

6.3.1 Overview

The process data channel is used for real-time data exchange of process data such as actual and setpoint position data or the operating status of the compact drive. The transmission can be executed very fast because it is sent without additional administration data and does not require confirmation by the recipient.

The master can also use the process data channel to control the status machine of the slave, enable and disable the power amplifier, trigger and reset a Quick-Stop, reset errors and enable operating modes.

Control of the status machine and enabling the operating modes must be done separately. An operating mode can generally only be enabled if the status is already `6 OPERATION-ENABLE`.

Note that a new operating mode and a new acceleration are generally only imported if the drive is stopped. In the process data channel acceleration values are accepted while the motor is moving, but the values are not set until the next movement job. All other information can be changed with the operating mode enabled.

6.3.2 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 transmitted data (master to slave) and received data (slave to master) is distinguished as follows:

Transmitted data format

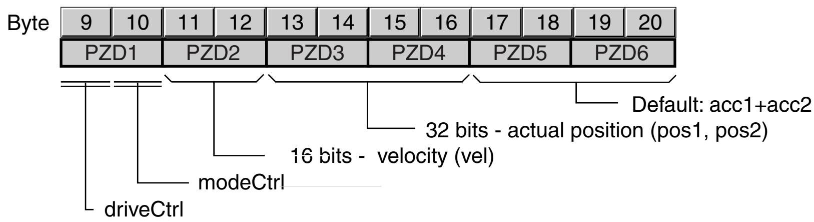


Figure 6.5 Transmitted data in the process data channel: master to slave

Byte 9+10: driveCtrl and modeCtrl, for setting the status machine and the operating mode. The exact structure is described below on page 6-7.

Byte 11+12: vel, setpoint speed for the operating mode, no meaning when setting dimensions.

Byte 13-16: Pos1, Pos2, position data in increments, depending on the operating mode

Byte 17-20: default:acc, these bytes can be configured, the content is specified by index and subindex. The acceleration is input as the default value (32bit); the parameter that is mapped in the PZD5 and PZD6 is set via the `MAPOUT` object. During configuration a check is made whether a legal value is written. If mapping is disabled, the data in bytes 17-20 are not relevant for the compact drive.

Description `driveCtrl`:

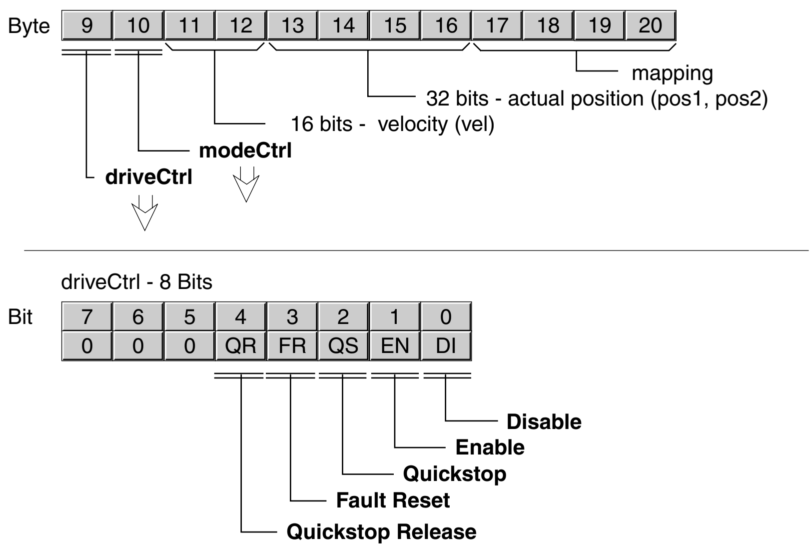


Figure 6.6 Transmitted data in the process data channel: `driveCtrl`

0098 441 113 193, -000, 09.03

The status machine is controlled over the process data channel or the parameter channel, `driveCtrl, 28:1` object, over the `Bits 0..4`.

During control via the process data channel these bits operate edge-selectively, i.e. the function is triggered with a “0 >> 1” edge.

One write access with set bit value is sufficient during access by parameter channel; an edge change is not required.

Note: The enable bit must always be set as long as the drive is receiving current!

Control of the status machine	**Process data channel bits 0...4**	**Parameter channel driveCtrl, 28:1 Bits 0...4**
Bit 0: Power amplifier disable	Processing with 0->1 edge	Processing with write access if bit value=1
Bit 1: Power amplifier enable		
Bit 2: Quick-Stop		
Bit 3: Fault Reset		
Bit 4: Quick-Stop release		

Table 6.7 Control of the status machine (driveCtrl)

The value “0” is a special case: If all `Bits 0..7` are “zero” during transmission, the compact drive interprets this as the“Disable” command and disables the power amplifier.

Error processing

If requests for control of the status machine by the compact drive cannot be implemented, the compact drive ignores these requests. There is no error response.

0098 441 113 193, -000, 09.03

Description `modeCtrl`

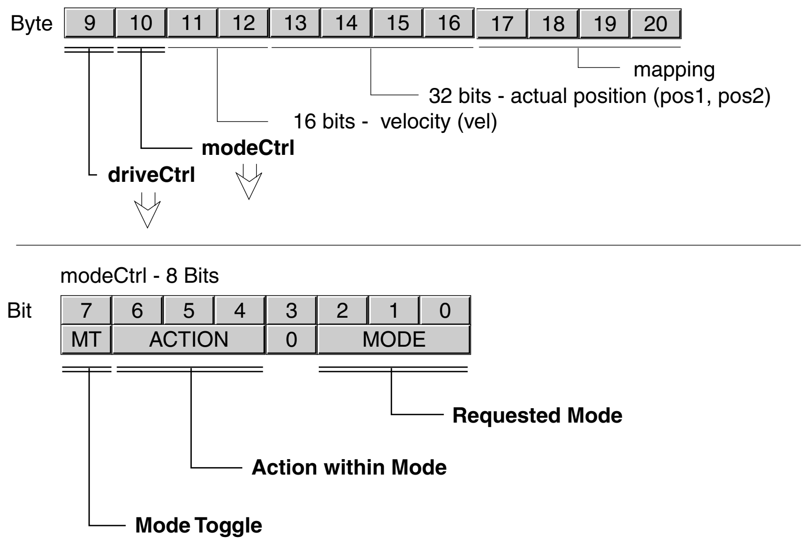


Figure 6.8 Transmitted data in the process data channel: `modeCtrl`

The operating modes are controlled via the `modeCtrl` object. The master must input the following values to trigger an operating mode or change setpoint values:

- Setpoint values in fields PZD3 and PZD4
- Select operating mode with `modeCtrl, Bits 0..2` (MODE)
- Select action for this operating mode with `modeCtrl, Bits 4..6` (ACTION)
- Toggle `modeCtrl, Bit 7` (MT)

The possible operating modes and the associated setpoint values are listed in Table 6.9 .

Mode Bits 0..2	Action Bits 4..6	modeCtrl* Bits 0..6	Description	corresponds to object**	Setpoint value PZD2	Setpoint value PZD4+5
2 (Homing)	0	02h	Dimension setting	40:3	-	Dimension setting position
	1	12h	Reference movement	40:1	Type (as obj. 40:1)	-
3 (PTP)	0	03h	Absolute positioning	35:1	Setpoint speed	Setpoint position
	1	13h	Relative positioning	35:3	Setpoint speed	Setpoint position
	2	23h	Continuation of positioning	35:4	Setpoint speed	-
4 (VEL)	0	04h	Speed mode	36:1	Setpoint speed	-

* column corresponds to the value to be input in byte `modeCtrl`, but without "ModeToggle" (Bit 7)

** column shows Index:Subindex (decimal) of the corresponding operating mode objects, which are described in more detail in the controller documentation.

Table 6.9 Setting operating modes via `modeCtrl`

0098 441 113 193, -000, 09.03

Setpoint positions are input in increments, setpoint speeds in rpm.

Data integrity must be guaranteed with simultaneous transmission of operating mode, setpoint position and setpoint speed in the process data channel. Therefore, the compact drive only evaluates the operating mode data if `Bit 7` has been toggled. Toggling means that a “0 >> 1” or a “1>> 0” edge change was detected at this bit since the last transmission.

`Bit 7` is mirrored in the received data set by the compact drive, the master detects by this that the slave has accepted the data.

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

Example of positioning: the drive must run a relative positioning of 20000 increments ($00004E20_h$). The setpoint speed will be 1000 rpm ($03E8_h$).

		Master					**Compact drive**
Trigger positioning	Transmitted data	>>	driveCtrl 02_h	modeCtrl 93_h	vel $03E8_h$	Pos1 + Pos2 $00004E20_h$	>>
Positioning is running x_err = 0, x_end = 0	Received data	<<	driveStat 0006_h	modeStat 83_h		Pos1 + Pos2 $XXXXXXXX_h$	<<
	Transmitted data	>>	driveCtrl 02_h	modeCtrl 93_h	vel $03E8_h$	Pos1 + Pos2 $00004E20_h$	>>
Positioning finished x_err = 0, x_end = 1, x_info = 1	Received data	<<	driveStat 6006_h	modeStat 83_h		Pos1 + Pos2 $00004E20_h$	<<

Table 6.10 Relative positioning

Input here is also done in the Motorola format. The acceleration can only be set over the process data channel if the corresponding object has been mapped in accordance with PZD5 and PZD6.

Example of speed mode: the compact drive is to rotate in speed mode at a setpoint speed of 1000 rpm ($03E8_h$). The master must send the following data to the slave:

		Master					**Compact drive**
Start speed mode at 1000 rpm	Transmitted data	>>	driveCtrl 02_h	modeCtrl 84_h	vel $03E8_h$	Pos1 + Pos2 $xxxxxxxx_h$	>>
Compact drive accelerates	Received data	<<	driveStat 0006_h	modeStat 84_h		Pos1 + Pos2 $XXXXXXXX_h$	<<
	Transmitted data	>>	driveCtrl 02_h	modeCtrl 84	vel $03E8_h$	Pos1 + Pos2 xxxxxxxx	>>
Setpoint speed reached x_err = 0, x_end = 0, x_info = 1	Received data	<<	driveStat 2006_h	modeStat 84_h		Pos1 + Pos2 xxxxxxxx	<<

Table 6.11 Speed mode

0098 441 113 193, -000, 09.03

The position data have no function in speed mode. The response data do not contain the current speed of the drive - when the specified set-point speed is reached this is shown by the mode bit in the status word.

Received data format

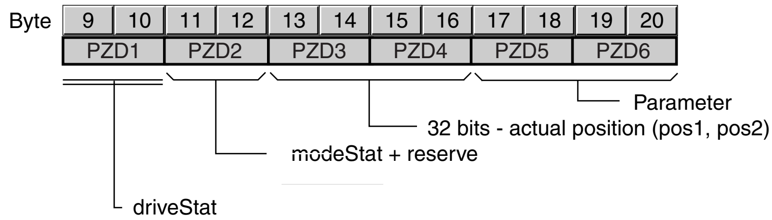


Figure 6.12 Received data in the process data channel: slave to master

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

Byte 11+12: modeStat, return of the currently set operating mode

Byte 13-16: Pos1, Pos2, current position data in increments

Byte 17-20: these bytes can be configured, the content is specified by index and subindex. With few exceptions, they demonstrate no time consistency with bytes 9-16, by default no objects are input because of runtime disadvantages.

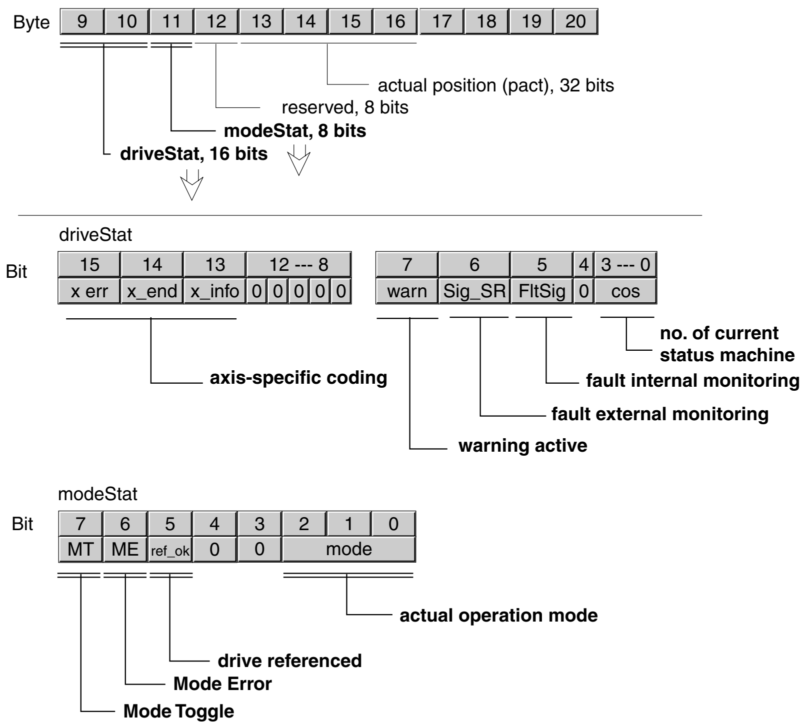


Figure 6.13 Structure of the received data in the process data channel

0098 441 113 193, -000, 09.03

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

Bit	Name	Description
0...2	mode	current specified operating mode as with transmitted data
5	ref_ok	Set if the compact drive has been successfully referenced by reference movement or dimension setting.
6	ME, ModeError	Set if a request by the “master” was rejected via transmitted data from the compact drive.
7	MT, ModeToggle	Mirroring of bit 7 (“Mode Toggle”) of the transmitted data

Table 6.14 Operating mode

A synchronised process can be run with the transmitted data , `Bit 7` (“ModeToggle”- MTreq) and received data, `Bit 6 and 7`. Synchronised process means that the “master” waits for acknowledgments from the compact drive and responds to them.

Example

Positioning with concluding check for correct execution.

		Master		Compact drive
Setpoint values for positioning in the fields PZD2, PZD3 and PZD4 - Setting the desired operating mode in field “modeCtrl” - Toggle bit 7	Transmitted data on the process data channel	>>	MTreq <>MTstat	>>
If MT is toggled, then - import values - start desired operating mode - update status: x_end = 0 - mirror MT from transmitted data in accordance with received data	Received data on the process data channel	<<	MTstat = MTreq x_end = 0	<<
If MTstat = MTreq the status is current: - check ME: 1 = request failed - Wait for x_end = 1: 1 = end of positioning		>>		>>
Positioning completed: x_end = 1	Received data on the process data channel	<<	MTstat = MTreq x_end = 1	<<

Table 6.15 Example: Positioning with final check

Special case of very short positioning

In the case of a very short positioning, it may occur that the compact drive has already reached the setpoint position when the status of the received data is returned to the “master”. In this case PZD1, `modeCtrl, Bit 7` is equal to PZD2, `modeStat, Bit 7` and `Bit x_end = 1` are already set. Thus there is not the case `x_end = 0` for the “master”. If no error has occurred, the positioning was correctly executed regardless.

Error processing

If the “master” toggles `Bit 7`, this applies as a request to the compact drive to start an operating mode or to change data of the current operating mode. If the compact drive cannot process the request, the compact drive signals this to the “master” by the following actions:

- Sending a diagnostic telegram with the corresponding error code

- In the received data in `modeStat Bit 6` ("ModeError") is set.

  This bit remains set until in the received data in `mode-Stat, Bit 7` ("ModeToggle") is toggled again.

  The "master" can read the corresponding error code via a read access to the `BLErrorcode, 30:11` object.

- Continuation of the current operating mode

The current operating mode is thus not influenced and a status change does not take place.

Some reasons for a failed operating mode request can be the following:

- Setpoint values outside the value range
- Switching the operating mode during processing (not possible)
- Invalid operating mode requested
- Status machine not in status 6 ("Operation Enable")

6.4 Diagnostic telegram

If an internal unit error occurs, the compact drive switches to error mode in accordance with the unit status machine. Because a diagnostic message is required on the slave side, the slave returns a response telegram of high priority. In the next bus cycle the master requests a diagnosis from this slave instead of running a normal data exchange. The response telegram of the compact drive contains a message with error register and error code.

If the request for an operating mode via the process data channel fails, the compact drive also sends a diagnostic request telegram.

The data format of the first 7 bytes is described in the Profibus-DP. The error code, the error register and a specific error code is input in bytes 9-16. Byte 8 contains information on internal device communication.

Byte	Description
1-3	Slave status
4	Master address
5	ID number high byte
6	ID number low byte
7	external diagnosis: Header and length information
8	external diagnosis: internal device communication
9+10	external diagnosis: error code
11	external diagnosis: error register
12-16	external diagnosis: specific error code (power amplifier)

Table 6.16 Structure of the diagnostic telegram

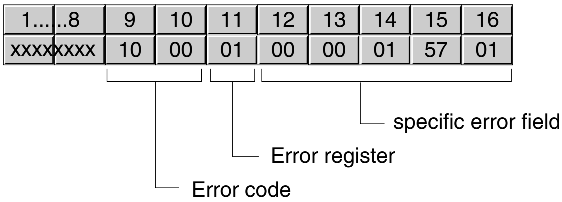


Figure 6.17 Diagnostic telegram

- Byte 9+10 (“Error Code”): Error code This value is always 1000h (generic error) for the compact drive.
- Byte 11 (“Error Register”): Error register The value is also saved in the `Error register, 1001`$_h$ object
- Byte 12-16 (“Specific Error Field”): Specific error of the power amplifier
  - Byte 12+13 = 0
  - Byte 14+15: manufacturer-specific error number (Motorola-Format)
  - Byte 16: error-class

The error numbers are listed in the controller manual.

6.5 PLC as field bus master

The field bus master supplied a separate memory area for the transmitted and received data for every connected slave. The data exchange between the PLC memory and the field bus master module can be run via the peripheral or the process image range.

The field bus transmission run asynchronously to the write and read accesses of the application program to the transmitted and received data. Therefore data from the field bus master may be read from the PLC memory before the PLC could completely update the data.

WARNING!

Danger of injury and damage of system parts by evaluation of erroneous control commands!

Data exchange with the use of a PLC as master may result in inconsistent transmitted data, because the field bus and PLC cycles do not operate synchronously.

For safe operation of a master PLC note the following:

- Copying data from high to low addresses
- finally toggle `modeCtrl, Bit 7` (MT)

- During data exchange via the process image the transmitted data must be copied from the memory for the process image to the memory of the field bus master module. Inconsistent data must not be created on the field bus by this copy process.

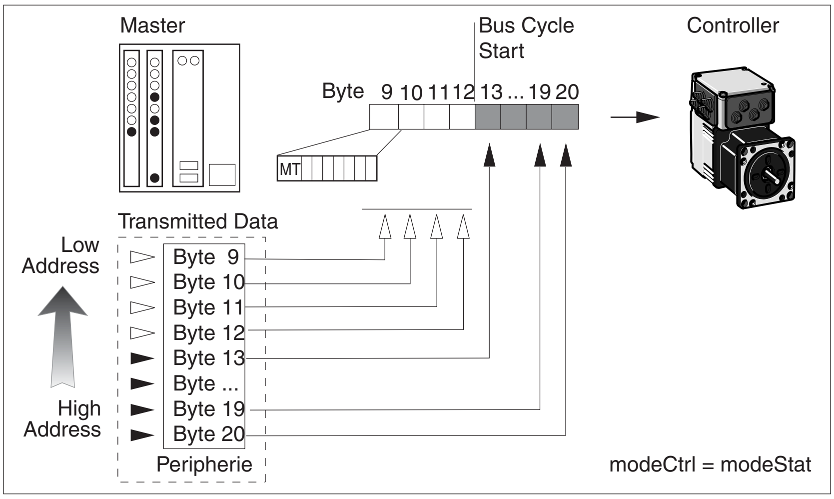


Figure 6.18 Secure data integrity, byte 9 (bit 7) is copied last

Data exchange over the peripheral memory

The data integrity is secured during data exchange over the peripheral memory if the `modeCtrl, Bit 7` is input last. The controller ignores the transferred data as long as this bit is equal to `modeStat, Bit 7`.

The following example shows the problems of unsecured data integrity:

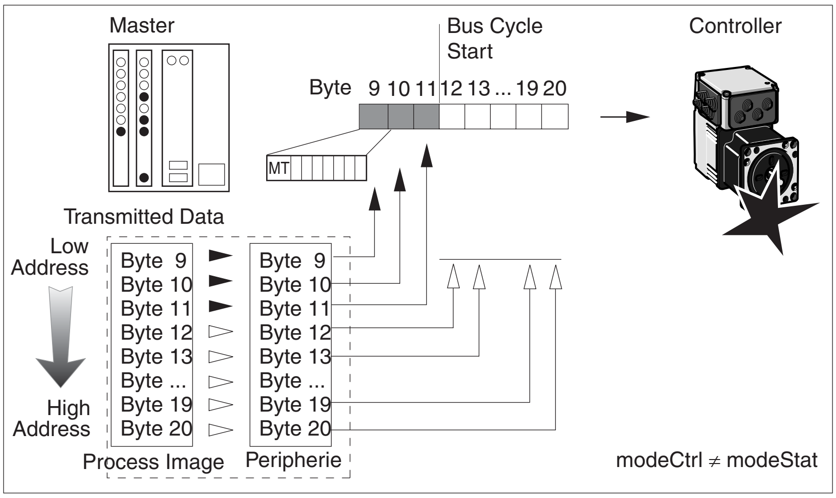


Figure 6.19 Unsecured data integrity -> undefined movement!!

Data exchange over process image memory

The data integrity during data exchange over the process image memory is only secured if there is no bus access to the data in the peripheral memory during the copy process between image and peripheral memory.

Inconsistent data are created if bit 7 of byte 9 (MT) is sent over the bus before the controller has received the remaining correct data. As soon as MT is transferred, the controller sets the status change when checking the bit and interprets it as a new command, which is executed immediately.

0098 441 113 193, -000, 09.03

Your local dealer offers drivers for control by Siemens PLC controllers. If you need these drivers, contact your local dealer's technical service.

0098 441 113 193, -000, 09.03

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 to one single object. The available objects are described in the Parameters chapter in the controller manual. Use over the parameter channel is described in this chapter with only a few objects as examples, because this type of communication can be used for all available user objects and always has a very similar structure.

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 are given for the application of the protocols supported by the compact drive, and the general procedure is described.

Structure of the examples

The following is shown in the examples:

- Task description
- Start conditions
- Required commands in the transmitted data frame
- Reaction of device 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. You will find information on this in the controller documentation.
- Field bus protocol and connection to the master controller
- Scope of function of the field bus profile.

Controller manual

The examples are designed to supplement the function descriptions in the manuals. The manual describes the basic functions of the operating modes and functions.

You will also find all parameters regarding the operating modes and functions listed there.

The number format of the parameter values in a field bus command is described in the controller manual.

7.2 Use of the parameter channel

7.2.1 Writing parameters

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 $1D_h$
- Subindex: 26 = $1A_h$
- Value:10000 = 00002710_h

The value 30_h must be input as PKE (parameter identification), because the parameter has a 32-bit data type.

Transmitted data

Object	PKE	Sdx	Idx	Data	Description
Tx $001D_h$:$1A_h$ Motion.acc	30_h	$1A_h$	00 $1D_h$	00 00 27_h 10_h	Set the acceleration to 10000 rpm*s = 2710_h 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 controller 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

Object	PKE	Sdx	Idx	Data	Description
Rx $001D_h$:$1A_h$ Motion.acc	20_h	$1A_h$	00 $1D_h$	XX XX XX XX	The response data have no meaning, the positive acknowledgment 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 = 00 $1F_h$
- Subindex 9 = 09_h

The value 10_h must be input as PKE. This identifies a read request.

Transmitted data

Object	PKE	Sdx	Idx	Data	Description
Tx $001F_h$:09_h Status.n_actT	10_h	09_h	00 $1F_h$	XX XX XX XX	Reading the actual speed. The data are meaningless.

The 4 data bytes are meaningless for a read request.

0098 441 113 193, -000, 09.03

Received data

Object	PKE	Sdx	Idx	Data	Description
Rx $001F_h$:09_h Status.n_act	20_h	09_h	00_h $1F_h$	00 00 03_h $E8_h$	The data 000003E8 correspond to 1000 rpm; PKE=20 signals successful execution.

The compact drive distinguishes between parameter values with 32-bit data and 16-bit data (described as INT32 or UINT32 and INT16 or UINT16 data types in the controller manual) by the response identification(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 compact drive responds with an error data frame (error response). For example, an error source can be trying to read or write a non-existent object. The transmitted error number shows information on the exact cause.

Received data with error data frame (error response)

Object	PKE	Sdx	Idx	Data	Description
Rx 3028_h:20_h	70_h	20_h	00 28_h	06_h 02_h 00 00	Error number 06020000h means: Object not in object directory

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

The table of error numbers can be found in Kapitel 4.2.1, "Synchrone Fehler".

7.3 Operating states 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

The compact drive recognises different operating states. The different operating states are numbered from 1 to 9. The operating states and the transition conditions are described in more detail in the controller manual, "Operation/Basics" chapter.

0098 441 113 193, -000, 09.03

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 status, power amplifier remains on
9	Fault	off	error status, power amplifier switched off

Table 7.1 Important operating states

Requirements for changing operating states are sent in the process data channel PZD1 in the field `driveCtrl` to the compact drive. The drive returns the current operating status in process data channel PZD1, driveStat field, to the master device.

Tabelle7.2, Seite 7-4 shows the bit assignment of the field `driveCtrl` in the transmitted data in the process channel (byte 9):

bit no.	Significance	Meaning
0	01_h	Disable
1	02_h	Enable
2	04_h	Quick-Stop
3	08_h	Fault-Reset
4	10_h	Quick-Stop release

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 state 4 to 6. For this purpose in byte 9 transmitted data, `driveCtrl`, the two bits `Enable` and `Disable` are present. One must always be set to 1 and the other to 0.

Switch on power amplifier

Condition: Compact drive is in operating state 4.

To switch on the power amplifier in `driveCtrl`, `Bit 1` (Enable) a 0>1 edge must be generated. This can be done by deleting `Bit 0` (Disable) and setting `Bit 1`. The master device waits until the compact drive reports operating state 6.

Example:

		Master		Compact drive
Disable is requested	Transmitted data	»	driveCtrl 01_h	»
Compact drive reports operating status 4	Received data	«	driveStat XXX4_h	«
Request enable	Transmitted data	»	driveCtrl 02_h	»
Compact drive reports operating status 5	Received data	«	driveStat XXX5_h	«
Request enable	Transmitted data	»	driveCtrl 02_h	»

0098 441 113 193, -000, 09.03

		Master		Compact drive
Compact drive report operating status 6	Received data	«	driveStat XXX6_h	«

Table 7.3 Switching on the power amplifier

Switch off power amplifier

Condition: Compact drive is in operating state 6 or 7.

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

Example:

		Master		Compact drive
Enable is requested	Transmitted data	»	driveCtrl 02_h	»
Compact drive reports operating status 6	Received data	«	driveStat XXX6_h	«
Request disable	Transmitted data	»	driveCtrl 01_h	»
Compact drive reports operating status 4	Received data	«	driveStat XXX4_h	«

Table 7.4 Switching off the power amplifier

7.3.2 Trigger Quick-Stop

A current movement job can be interrupted over the field bus at any time with the `Quick-Stop` command. It is triggered by a 0>1 edge in `driveCtrl, Bit 2`. The compact drive brakes with the specified emergency stop ramp by switching to operating state 7 (Quick-Stop) and comes to a standstill.

The compact drive must be placed in operating state 6 to start a new movement job. 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		Compact drive
Enable is requested	Transmitted data	»	driveCtrl 02_h	»
Compact drive reports operating status 6	Received data	«	driveStat XXX6_h	«
Quick Stop and Request enable	Transmitted data	»	driveCtrl 06_h	»
Compact drive reports operating status 7	Received data	«	driveStat XXX7_h	«
Wait until compact drive is at standstill and system should continue to run				
Compact drive reports operating status 7	Received data	«	driveStat XXX7_h	«
Request Quick-Stop release and enable	Transmitted data	»	driveCtrl 12_h	»
Compact drive reports operating status 6	Received data	«	driveStat XXX6_h	«
Cancel Quick-Stop release	Transmitted data	»	driveCtrl 02_h	»
Compact drive reports operating status 6	Received data	«	driveStat XXX6_h	«

0098 441 113 193, -000, 09.03

Table 7.5 Triggering Quick-Stop

7.3.3 Fault reset

If an error occurs during operation, the compact drive switches to operating state 7 (Quick-Stop) or operating state 9 (Fault) depending on the error that has occurred.

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 compact drive was originally in operating status 7, after the Fault Reset it switches to operating status 6.

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

Example:

		Master		**Compact drive**
Request enable	Transmitted data	»	driveCtrl 02_h	»
Compact drive reports operating status 9 (Fault)	Received data	«	driveStat $XXX9_h$	«
Correct error				
Request Fault Reset	Transmitted data	»	driveCtrl 08_h	»
Compact drive reports operating status 4	Received data	«	driveStat $XXX4_h$	«
Request enable	Transmitted data	»	driveCtrl 02_h	»
Compact drive reports operating status 5	Received data	«	driveStat $XXX5_h$	«
Request enable	Transmitted data	»	driveCtrl 02_h	»
Compact drive reports operating status 6	Received data	«	driveStatXXXX6_h	«

Table 7.6 Fault reset

Note: In this example with the Fault Reset the master device deletes the `Bit 1` (Enable) to be able to run an implicit 0>1 edge at `Bit?1`. This returns the compact drive to operating state 6.

7.4 Examples: Operating modes process data channel

Transmitted data

You can start movement commands and change them during processing with the transmitted data.

The following fields are available in the process data channel:

- PZD1: `modeCtrl` Start and change operating mode
- PZD2: `vel` Setpoint speed, mode-dependent
- PZD3+4: `Pos1` and `Pos2` position, mode-dependent

0098 441 113 193, -000, 09.03

- PZD5+6: `Def:acc1` and `Def:acc2` mapped value, mode.dependent

The default values of these fields are only applied by the compact drive if `modeCtrl, Bit 7` (`ModeToggle`) has been changed.

Always proceed as follows to transfer values to the compact drive:

▶ Input the desired operating mode and the associated default values into the fields `modeCtrl`, PZD2-6 on.

▶ Change `modeCtrl, Bit 7` (`ModeToggle`)

This is a method of always avoiding consistency problems within the transmitted data.

Empfangsdaten Movement jobs 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
- PZD3+PZD4: `p_act` Actual position of the compact drive
- PZD5+PZD6: parameters can be set, but no temporal consistency with PZD1-4 except for the following. The following objects can be input: `Temp, EA, Spannung, Fehlernummer, Strom`. The error number (matches status word) and the IO word can be set consistently. The object `MAPIN` is used to set the parameter that will be mapped to 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 definable values in bytes 17-20, (siehe Abschnitt 4.5.2, Seite 10)).

Mode Toggle The transmitted and received logs both contain the bit `Mode-Toggle`. The master device sets this bit in the transmitted log and the compact drive mirrors it in the received log. The procedure enables the master device to detect whether the data sent by the compact drive are current.

Example

The master device starts a positioning movement that will only take a very short time. The master device waits for the end of the positioning by checking the received log for bit `x_end = 1` (identified positioning end).

The master device may receive data from the compact drive that originate from the period before the start of the positioning movement. They also include `x_end = 1`. Now the master device detects that the data are old, because the included bit `ModeToggle` does not match that of its positioning job.

In general, the master device should only evaluate data in which the received bit `ModeToggle` is identical with the last bit it sent.

Acceleration Before a positioning the desired acceleration can be set first by mapping the acceleration to PZD5 and PZD6 or via the parameter channel (object `Motion.acc, 29:26`). Note that the acceleration can only be changed when the compact drive is at a standstill.

Assumptions The examples in this chapter are based on the following assumptions:

- Operating status 6 (Operation Enable)

0098 441 113 193, -000, 09.03

- Compact drive has not yet been referenced (bit `ref_ok = 0`)
- `p_act = 0` (actual position)
- 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

To start an absolute positioning movement the following setting must be made in the transmitted log:

▶ Input in PZD2 `(= vel)` the setpoint speed and in PZD3 and PZD4 (`= Pos1+Pos2`) the target position.

▶ Input in the field `modeCtrl` operating mode 03_h (point-to-point mode, absolute positioning).

▶ Change `modeCtrl, Bit 7`, so the data will be applied by the compact drive.

Example 1:

Absolute positioning to position 100.000 ($000186A0_h$)

at a setpoint speed of 1000 rpm ($03E8_h$)

			Master				Compact drive
Trigger positioning	Transmitted data	»	driveCtrl 02_h	modeCtrl 83_h	vel $03E8_h$	Pos1 + Pos2 $000186A0_h$	»
Positioning running x_err = 0, x_end = 0	Received data	«	driveStat 0006_h	modeStat 83_h		Pos1 + Pos2 $XXXXXXXX_h$	«
Trigger positioning	Transmitted data	»	driveCtrl 02_h	modeCtrl 83_h	vel $03E8_h$	Pos1 + Pos2 $000186A0_h$	»
Positioning complete x_err = 0, x_end = 1, x_info = 1	Received data	«	driveStat 6006_h	modeStat 83_h		Pos1 + Pos2 $000186A0_h$	«

Table 7.7 Point-to-point mode, absolute positioning at constant setpoint speed

Note: The data frame "positioning running" can also be sent multiple times; the field `Pos1 + Pos2` contains the current actual position.

Example 2:

As in example 1, except that the setpoint speed is changed to 2000 rpm ($07D0_h$) during the movement.

0098 441 113 193, -000, 09.03

			Master				Compact drive
Trigger positioning	Transmitted data	»	driveCtrl 02_h	modeCtrl 83_h	vel $03E8_h$	Pos1 + Pos2 $000186A0_h$	»
Positioning running x_err = 0, x_end = 0	Received data	«	driveStat 0006_h	modeStat 83_h		Pos1 + Pos2 $XXXXXXXX_h$	«
Change setpoint speed	Transmitted data	»	driveCtrl 02_h	modeCtrl 03_h	vel $07D0_h$	Pos1 + Pos2 $000186A0_h$	»
Positioning running x_err = 0, x_end = 0	Received data	«	driveStat 0006_h	modeStat 03_h		Pos1 + Pos2 $XXXXXXXX_h$	«
Change setpoint speed	Transmitted data	»	driveCtrl 02_h	modeCtrl 03_h	vel $07D0_h$	Pos1 + Pos2 $000186A0_h$	»
Positioning complete xerr=0, x_end = 1, x_info = 1	Received data	«	driveStat 6006_h	modeStat 03_h		Pos1 + Pos2 $000186A0_h$	«

Table 7.8 Point-to-point mode, absolute positioning with change of setpoint speed

Note: The data frame "positioning running" can also be sent multiple times. The field `Pos1 + Pos2` contains the current actual position. When the setpoint speed is changed the same target position is sent, because it does not change in this example.

7.4.2 Relative 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

A relative positioning is run similarly to the absolute positioning. The only change is that the field `modeCtrl` must contain the value 13_h (point-to-point mode, relative positioning). It is also important to ensure that multiple target positions transferred in succession are added.

Example: Relative positioning by 100.000 ($000186A0_h$) Increments with a speed of 1000 rpm ($03E8_h$)

During the movement the speed must be changed to 2000 rpm ($07D0_h$).

			Master				Compact drive
Trigger positioning	Transmitted data	»	driveCtrl 02_h	modeCtrl 93_h	vel $03E8_h$	Pos1 + Pos2 $000186A0_h$	»
Positioning running x_err = 0, x_end = 0	Received data	«	driveStat 0006_h	modeStat 83_h		Pos1 + Pos2 $XXXXXXXX_h$	«
Change setpoint speed Send relative position 0	Transmitted data	»	driveCtrl 02_h	modeCtrl 13_h	vel $07D0_h$	Pos1 + Pos2 00000000_h	»

0098 441 113 193, -000, 09.03

		Master					Compact drive
Positioning running x_err = 0, x_end = 0	Received data	«	driveStat 0006_h	modeStat 03_h		Pos1 + Pos2 $XXXXXXXX_h$	«
Change setpoint speed Send relative position 0	Transmitted data	»	driveCtrl 02_h	modeCtrl 13_h	vel $07D0_h$	Pos1 + Pos2 00000000_h	»
Positioning complete x_err = 0, x_end = 1, x_info = 1	Received data	«	driveStat 6006_h	modeStat 03_h		Pos1 + Pos2 $000186A0_h$	«

Table 7.9 Point-to-point mode, relative positioning with change of setpoint speed

Comments: The data frame "positioning running" can also be sent multiple times; the field `Pos1 + Pos2` contains the current actual position. When the setpoint 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 Speed mode

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

In speed mode a setpoint speed is specified for the motor, and 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 setpoint speed while speed mode is running:

▶ Input in PZD2, `vel`, the setpoint speed (Pos1 + Pos2 has no meaning here)

▶ Input in `modeCtrl` operating mode 04_h (speed mode)

▶ Change `modeCtrl, Bit 7`, so the data will be applied by the compact drive

Example

Speed mode is started at a setpoint speed of 1000 rpm ($03E8_h$).

The setpoint speed is changed to 2000 rpm ($07D0_h$) during the movement.

Speed mode is ended by sending the setpoint speed 0 and then waiting for the compact drive to come to a standstill.

0098 441 113 193, -000, 09.03

		Master					Compact drive
Start speed mode at 1000 rpm	Transmitted data	»	driveCtrl 02_h	modeCtrl 84_h	vel $03E8_h$	Pos1 + Pos2 $XXXXXXXX_h$	»
Compact drive accelerates xerr=0, xend=0, xinfo=0	Received data	«	driveStat 0006_h	modeStat 84_h		Pos1 + Pos2 $XXXXXXXX_h$	«
Speed mode at 1000 rpm	Transmitted data	»	driveCtrl 02_h	modeCtrl 84_h	vel $03E8_h$	Pos1 + Pos2 $XXXXXXXX_h$	»
Setpoint speed reached xerr=0, xend=0, xinfo=1	Received data	«	driveStat 2006_h	modeStat 84_h		Pos1 + Pos2 $XXXXXXXX_h$	«
Change speed to 2000 rpm	Transmitted data	»	driveCtrl 02_h	modeCtrl 04_h	vel $07D0_h$	Pos1 + Pos2 $XXXXXXXX_h$	»
Compact drive accelerates xerr=0, xend=0, xinfo=0	Received data	«	driveStat 0006_h	modeStat 04_h		Pos1 + Pos2 $XXXXXXXX_h$	«
Speed at 2000 rpm	Transmitted data	»	driveCtrl 02_h	modeCtrl 04_h	vel $07D0_h$	Pos1 + Pos2 $XXXXXXXX_h$	»
Setpoint speed reached xerr=0, xend=0, xinfo=1	Received data	«	driveStat 2006_h	modeStat 04_h		Pos1 + Pos2 $XXXXXXXX_h$	«
Change speed to 0 rpm	Transmitted data	»	driveCtrl 02_h	modeCtrl 84_h	vel 0000_h	Pos1 + Pos2 $XXXXXXXX_h$	»
Compact drive decelerates xerr=0, xend=0, xinfo=0	Received data	«	driveStat 0006_h	modeStat 84_h		Pos1 + Pos2 $XXXXXXXX_h$	«
Change speed to 0 rpm	Transmitted data	»	driveCtrl 02_h	modeCtrl 84_h	vel 0000_h	Pos1 + Pos2 $XXXXXXXX_h$	»
Speed mode ended xerr=0, xend=1, xinfo=1	Received data	«	driveStat 6006_h	modeStat 84_h		Pos1 + Pos2 $XXXXXXXX_h$	«

Table 7.10

Note: The field `Pos1 + Pos2` of the received log contains the current position of the drive in increments.

7.4.4 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:

- Input the new position to Pos1 + Pos2. (`PZD2` has no meaning here)
- Input in `modeCtrl` operating mode 02_h (referencing, dimension setting).
- Change `modeCtrl, Bit 7`, so the data will be applied by the compact drive

Example: The motor stops at position "-100.000" ($FFFE7960_h$).

Position 200.000 is assigned to the motor ($00030D40_h$).

		Master					Compact drive
Compact drive reports position -100.000	Received data	«	driveStat $XXXX_h$	modeStat XX_h		Pos1 + Pos2 $FFFE7960_h$	«

			Master				Compact drive
Dimension setting at 200.000	Transmitted data	»	driveCtrl 02_h	modeCtrl 82_h	vel $XXXX_h$	Pos1 + Pos2 $00030D40_h$	»
Position applied x_err = 0, x_end = 1, x_info = 0	Received data	«	driveStat 4006_h	modeStat $A2_h$		Pos1 + Pos2 $00030D40_h$	«

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

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 controller manual for more information on setting parameters and on running a reference movement.

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

- Input in `PZD2` the type of the reference movement (`PZD3+PZD4` is meaningless here).

  The types of reference movement are described in the controller manual.

- Input in `modeCtrl` operating mode 12_h (referencing, reference movement).
- Change `modeCtrl, Bit 7`, so the data will be applied by the compact drive

Example

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

			Master				Compact drive
Trigger reference movement	Transmitted data	»	driveCtrl 02_h	modeCtrl 92_h	PZD2 0002_h	PZD3 + PZD4 $XXXXXXXX_h$	»
Reference movement running xerr=0, xend=0	Received data	«	driveStat 0006_h	modeStat8 02_h		Pos1 + Pos2 $XXXXXXXX_h$	«
Reference movement	Transmitted data	»	driveCtrl 02_h	modeCtrl 92_h	PZD2 0002_h	PZD3 + PZD4 $XXXXXXXX_h$	»

		Master					Compact drive
Reference movement complete xerr=0, xend=1	Received data	«	driveStat 4006_h	modeStat $A2_h$		Pos1 + Pos2 00000000_h	«

Table 7.12 Reference movement

7.5 Error signalling in process data channel

7.5.1 Synchronous errors

If a request for an operating mode sent by the compact drive via the transmitted log cannot be processed, the compact drive rejects it and places `modeStat, Bit 6` (ModeError) in the received log. This does not interrupt the current process. To find the cause of the error the master device can now read the error number from the object `Status.modeError, 30:11` with an access via the parameter channel.

Group.Name Index:Subindex dec. (hex.)	Meaning of bit assignment	Data type range (dec.)	Unit Default (dec.)	R/W/ rem. Info S.
Status.ModeError 30:11 (1E:0Bh)	Error code that results in setting the ME flag in a PDO Manufacturer-specific error code that results in setting the ModeError flag in a PDO.In general this is an error that was triggered by starting an operating mode.	UINT16	- 0	R/-/-

Table 7.13 Profibus parameter group

Example

The compact drive rotates in speed mode. An attempt is made to run a dimension setting.

		Master					Compact drive
Speed mode x_end = 0	Received data	«	driveStat 0006_h	modeStat 04_h		Pos1+Pos2 $XXXXXXXX_h$	«
Request: Dimension setting to 0	Transmitted data	»	driveCtrl 02_h	modeCtrl 82_h	PZD2 $XXXX_h$	PZD3+PZD4 00000000_h	»
Request rejected ModeError = 1	Received data	«	driveStat 0006_h	modeStat $C4_h$		Pos1+Pos2 XXXXXXXXh	«

Table 7.14 Synchronous error, Invalid request of an operating mode

Note: If a request for dimension setting is rejected, the compact drive continues to rotate in speed mode without change.

However, the compact drive sends diagnostic telegram to the master device with the corresponding error number.

7.5.2 Asynchronous errors

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

Asynchronous errors are displayed as follows:

- Switch to operating state 7 (Quick-Stop) or operating state 9 (Fault).

  The change is displayed in the received log `driveStat, Bits 0..3`.

- Setting `driveStat, Bit 5` (fault by internal monitoring) or `driveStat, Bit 6` (fault by external monitoring)
- In case of a fault message by the internal monitoring:

  set the bits corresponding to the fault in the object `?Status.FltSig_SR, 28:18`.

  In the event of a fault message by the external monitoring: set the bits corresponding to the fault in the object `Status.Sign_SR, 28:15`

- An error number is also assigned to every error. With an asynchronous error the corresponding error number from the object `Status.StopFault (32:7)` is read.

Example: Fault message triggered by the external monitoring; approach the positive limit switch LIMP.

		Master					**Compact drive**
Trigger positioning	Transmitted data	»	driveCtrl 02_h	modeCtrl 03_h	vel $03E8_h$	Pos1 + Pos2 $0FFF8765_h$	»
Positioning running xerr=0, xend=0	Received data	«	driveStat 0006_h	modeStat 03_h		Pos1+Pos2 $XXXXXXXX_h$	«
Positioning	Transmitted data	»	driveCtrl 02_h	modeCtrl 03_h	vel $03E8_h$	Pos1 + Pos2 $0FFF8765_h$	»
Limit switch detected xerr=1, xend=0	Received data	«	driveStat 8047_h	modeStat 03_h		Pos1+Pos2 $XXXXXXXX_h$	«
Positioning	Transmitted data	»	driveCtrl 02_h	modeCtrl 03_h	vel $03E8_h$	Pos1 + Pos2 $0FFF8765_h$	»
Motor stopped xerr=1, xend=1	Received data	«	driveStat $C047_h$	modeStat 03_h		Pos1+Pos2 $XXXXXXXX_h$	«

Table 7.15 Asynchronous errors

Note: When a limit switch is detected, the motor brakes to a standstill with the emergency stop ramp and the bit `x_err` is set. After the motor is at a standstill bit `x_end` is set.

0098 441 113 193, -000, 09.03

8 Diagnostics and troubleshooting

8.1 Communication error diagnostics

Correctly functioning field bus operation is necessary for evaluation of operating and error messages.

Checking connections

If the compact drive cannot be addressed over the field bus, first check the connections. The controller manual contains the specifications for the compact drive and information on the network and device installation.

Check the following connections:

- ▶ System power supply
- ▶ Power connections
- ▶ Field bus cables and wiring
- ▶ Field bus connection

Operational error

When the compact drive is functioning correctly with the power amplifier switched off the LED in the connector housing flashes steadily at 0.5 Hz (1 second on, 1 second off). If this is not the case, the compact drive has an operational error. See the controller manual for information on causes of errors and troubleshooting.

Function test on the field bus

If the drive is connected correctly, check the DIP switch settings.

After correct configuration of the transmission data test the field bus operation.

Apart from the master device, which the drive detects by GSD and addressing, a bus monitor that displays messages as a passive device should be installed.

- ▶ Switch the drive system power supply off and on again.
- ▶ Observe the network messages shortly after switching on the drive system. The elapsed time between telegrams and the relevant information of the telegram contents can be read during recording with a bus monitor.

Possible errors: Addressing, setting parameters, configuration

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

- Addressing: The address of all network devices must be between "1" and "127". Every network device must have a different address.
- Setting parameters: The parameterised ID number and user-defined parameters must match the values saved in the GSD file.
- Configuration: The data length in the input and output direction must be identical with the length specified in the GSD file.

If network operation cannot be started, the network function of the drive system must be checked by your local dealer. In this case contact your local dealer.

0098 441 113 193, -000, 09.03

8.2 Error messages

The master device receives error messages over the field bus during operation.

The following error messages may be encountered:

- Synchronous error
- Asynchronous error
- Error during operating mode control over process data channel.

8.2.1 Synchronous error

If the compact drive cannot process a command in the parameter channel, the master device receives a synchronous error message directly from the compact drive.

Error message in the 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.

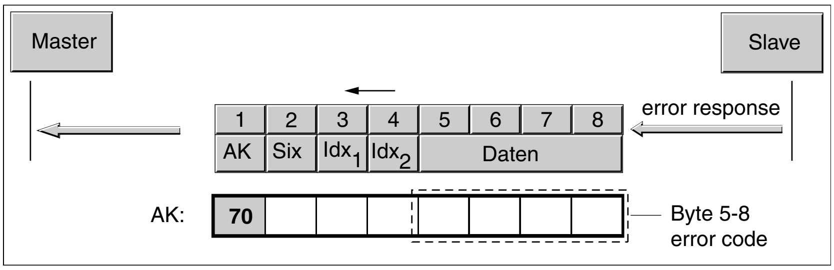


Figure 8.1 Error message in the parameter channel

Causes of a synchronous error

Possible causes of a synchronous error are:

- Error during execution of an action or control command.
- Parameter value outside the allowed value range
- Illegal action or control command during a running process
- Access to unknown object (Index/Subindex)

The table on page 8-3 shows all error messages that can occur with the compact drive.

Error code	Meaning
$0504\ 0001_h$	Job identification (AK) incorrect or unknown
$0601\ 0002_h$	Write access not allowed, because read object (ro)
$0602\ 0000_h$	Object not in object directory
$0607\ 0010_h$	Data type and parameter length do not match
$0609\ 0011_h$	Subindex not supported
$0609\ 0030_h$	Parameter value too large or too small (only relevant for write access)

0098 441 113 193, -000, 09.03

Error code	Meaning
0800 $xxxx_h$	Manufacturer-specific error, "xxxx“ corresponds to the error number of the compact drive. The error number is listed in the error number table of the controller manual

Table 8.2 Error codes

8.2.2 Asynchronous error

If a device error occurs, an asynchronous error is sent by the monitoring instruments of the compact drive.

In the case of asynchronous errors that result in a movement interruption the compact drive sends a diagnostic telegram.

An asynchronous error is sent via different objects:

- In the received log in the parameter `Status.driveStat, 28:2` with the following bits:
  - `Bit 15, x_err`

    Error status during processing

    Cause via the `Bits 5 und 6` evaluate
  - `Bit 5`

    Error message of an internal monitoring signal (e.g. overtemperature)

    The error information is input bit-coded in the `Status.FltSig_SR, 28:18` parameter.
  - `Bit 6`

    Error message of an external monitoring signal

    (e.g. movement interruption by limit switch)

    The exact cause is input bit-coded in the `Status.Sign_SR, 28:15` parameter.
  - `Bit 7` Controller warning message (e.g. warning of overtemperature)

    The error information is input bit-coded in the `Status.WarnSig, 28:10` parameter.
- The last interruption cause is also input in the parameter `Status.StopFault, 32:7` as error number.

  The error numbers and their meaning are listed in Chapter 7, "Diagnostics and troubleshooting" of the controller manual.
- In the diagnostics telegram with the corresponding error number. The error number is identical with that in the parameter `Status.StopFault, 32:7`.

Error message

If the controller sets the `x_err` bit, it interrupts the movement mode immediately and reacts either with braking or immediate shut-down of the power amplifier in accordance with the error class. Together with the

Bit15, x_err is Bit 6 or Bit 7 set. The meaning of the error message can be found with the corresponding parameter.

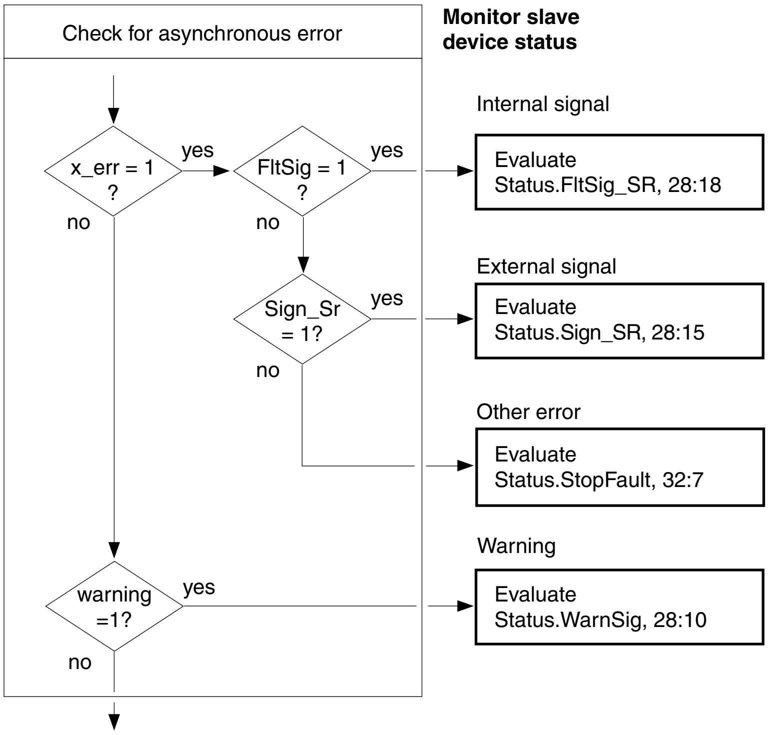


Figure 8.3 Evaluating asynchronous errors

As a simplified procedure the master device can also just evaluate the diagnostics telegram and react or visualise the error numbers accordingly.

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

8.2.3 Error during operating mode control via process data channel

You can trigger and modify movement jobs via the process control. If the compact drive cannot process the request, the compact drive sends a diagnostic telegram to the master device and sets one of the error bits in the received data.

0098 441 113 193, -000, 09.03

9 Parameters

9.1 Overview Parameters

Parameter groups

CAN	CAN bus settings
Capture	"Fast position capture“ function
Commands	Save parameter status change in EEPROM Initialise default parameters
Control	Controller settings
ErrMem0	Error memory
Homing	"Referencing“ operating mode
I/O	Status and definition of inputs and outputs
Motion	Operating function "Definition of direction of rotation“ Operating function "Quick-Stop“ Default setpoint speed Acceleration and deceleration
Profibus	Profibus settings
ProgIO0..3	Operating function "Programmable inputs/outputs“
PTP	"Point-to-point" operating mode
RS485	RS485 bus settings
Settings	User device names Phase currents Monitoring inputs
Status	Status information
VEL	"Speed mode" operating mode

The above table shows all parameter groups; this document only lists the parameters for the field bus described. See the relevant controller manual for the device-specific parameters.

Range of values

In the case of parameters with range of values specified the valid range of value depends on the data type.

Data type	Byte	Min value	Max value
INT16	2 byte / 16 bit	-32768	32767
UINT16	2 byte / 16 bit	0	65535
INT32	4 byte / 32 bit	-2.147.483.648	2.147.483.647
UINT32	4 byte / 32 bit	0	4.294.967.295

Table 9.1 Data types and ranges of values

0098 441 113 193, -000, 09.03

9.2 Parameter groups

9.2.1 "Profibus“ parameter group

Group.Name Index:Subindex dec. (hex.)	Meaning Bit assignment	Data type range (dec.)	Unit Default (dec.)	R/W/ rem. Info S.
Profibus.MapOut 24:2 ($18:02_h$)	Value in PZD5+6 to drive Index and subindex of the object that is mapped to the PPO2 during data transfer from the master device to the drive. The setpoint acceleration is mapped by default.Possible values: 00000000h: No mapping active 001A001Dh: setpoint acceleration (29:26) 00010021h: digital outputs (33:1)Low word: index mapped object high word: subindex mapped object	UINT32	- 0x001 A001D	R/W/rem. ?
Profibus.MapIn 24:3 ($18:03_h$)	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: 00000000h: No mapping active 00070020h: error number (32:7) 00010021h: Dig. inputs/outputs (33:1) 0019001Fh: temperature of power amplifier (31:25) 0014001Fh: power supply (31:20) 000C001Fh: current motor current (31:12)Low word: index mapped object high word: Subindex mapped object	UINT32	- 0	R/W/rem. ?
Profibus.PkInhibit 24:4 ($18:04_h$)	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/rem. ?
Profibus.SafeState 24:5 ($18:05_h$)	Reaction to safe status Reaction of the drive in status 'Clear' of the Profibus DP master device.0 = no reaction 1 = error of class 2, drive goes to FAULT status if power amplifier was enabled.	UINT32 0..1	- 1	R/W/rem. ?

Table 9.2 "Profibus" parameter group

0098 441 113 193, -000, 09.03

10 Accessories and spare parts

The following are available as accessories and spare parts:

Designation	Order number
IclA Ixx Installation Set	0062 501 521 001
IclA Ixx Cable Glands 2 units	0062 501 520 002
IclA Ixx Cable Glands 10 units	0062 501 520 001
IclA IDx Cable (supply, P/D), length: 3m, 5m, 15m, 10m, 20m	0062 501 464 xxx
IclA IFx Cable (supply, CAN), length: 3m	0062 501 462 030
IclA IFx Cable (Supply, RS485)	0062 501 463 xxx
IclA IFx Cable (Supply: STAK), length: 3m, 5m, 10m, 15m, 20m	0062 501 470 xxx
IclA IFx slide-in signal-connector 2 I/O	0062 501 524 001
IclA IFx connector-set 2 I/O	0062 501 523 001

Table 10.1 Accessories for the compact drives

Designation	Order number
Documentation	
IclA Ixx CD-ROM multilingual	0098 441 113 207
IclA IFE7x DE	0098 441 113 211
IclA IFE7x GB	0098 441 113 212
IclA IFx CANopen DE	0098 441 113 184
IclA IFx CANopen GB	0098 441 113 185
IclA IFx RS485 DE	0098 441 113 186
IclA IFx RS485 GB	0098 441 113 187
IclA IFx Profibus DE	0098 441 113 192
IclA IFx Profibus GB	0098 441 113 193

Table 10.2 Documentation for the compact drives

Recommended suppliers for Profibus cables:

- Profibus Cable (M12-M12) xxm: Profibus signal line, prepared both ends with M12 plug connector- M12 coupling, 5-pin B-coded. Supplier: Lumberg, www.lumberg.de Order no.: 0975 254 101 / ... M
- Profibus Cable (M12 SubD) xxm: Profibus signal line prepared both ends with M12 coupling, 5-pin B-coded, 9-pin SubD plug connector with switchable terminator. Supplier: Lumberg, www.lumberg.de Order no.: 0975 254 104 / ... M
- Profibus Cable (M12 SubD) xxm: Profibus signal line prepared both ends with M12 plug connector, 5-pin B-coded, 9-pin SubD plug connector with switchable terminator. Supplier: Lumberg, www.lumberg.de Order no.: 0975 254 105 / ... M

0098 441 113 193, -000, 09.03

The crimping pliers required for preparing cables are obtained directly from the manufacturer.

- Crimping pliers for power supply: AMP 654174-1
- Crimping pliers for multifunction interface and 24-V signal interface: Molex 69008-0982
- Crimping pliers for field bus interface: Molex 69008-0724

0098 441 113 193, -000, 09.03

11 Service, maintenance and disposal

11.1 Service address

Please contact your local dealer if you have any questions or problems. Your dealer will be happy to give you the name of a customer service outlet in your area.

11.2 Replacing units

After a slave unit has been replaced, the new unit should operate in exactly the same way as the old one. The new unit must have the same parameter settings to ensure this.

If the default values of other parameters need to be changed, these values can be stored on the master controller. They must be sent every time the compact drive is started, e.g. in the `ReadyToSwitchOn` status.

Your local dealer offers drivers for control by Siemens PLC controllers. If you need these drivers, contact your local dealer's technical service.

0098 441 113 193, -000, 09.03

0098 441 113 193, -000, 09.03

12 Supplement

0098 441 113 193, -000, 09.03

0098 441 113 193, -000, 09.03

13 Glossaries

13.1 Terms and abbreviations

Address	Memory location which can be accessed by its unique number. See also slave address.
AK	Job/answer identification
Broadcast	Type of data transmission in the network, one device sends a message to all devices on the network
Default values	Preset values for the parameters before initial commissioning, factory settings.
DIP switch	Small switches positioned side by side. They must be set during installation.
DP	**D**ecentralized **P**eriphery
Direction of rotation	Rotation of the motor shaft in a clockwise or anticlockwise direction. If the face plate of the extended motor shaft is observed and the motor is rotating in a clockwise direction, the direction of rotation is positive.
I/O	Inputs/Outputs
EMC	Electromagnetic compatibility
Limit switch	Switches that signal an overrun of the permissible travel range.
Error class	Classification of possible operating faults of the drive system that result in an error status.
FMS	Fieldbus-Message-Specification
GSD file	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..
HEX switch	Small rotary switch with 16 positions. It must be set during installation.
Idx	Index value of a parameter
Inhibit time	A PDO can be assigned a minimum waiting time for repeat transmissions in order to relieve the data transfer volume on the field bus. After the first transmission, the PDO is not re-sent until the delay has expired.
LED	Light-Emitting Diode
LWL	Optic fiber
Master	Active bus user that controls the data traffic in the network.
MT	**M**ode**T**oggle, bit change 0 » 1 or 1 » 0
Node-Guarding	Monitoring function by slave at an interface for cyclic communication.
Parameter	Device functions and values that can be set and called by the user.
PKE	Parameter code
PNO	Profibus User Organisation

0098 441 113 193, -000, 09.03

Profibus Standardised open field bus compliant with EN 50254-2 over which drives and other devices from different manufacturers communicate with one another.

PWE Parameter value

PZD Process data

Quick-Stop This function is used in the event of faults, the `STOP` command or for fast braking of the motor.

Six Subindex value of a parameter

Slave address Direct communication between master and slave devices is only possibly after assignment of addresses.

Slave Passive bus user that receives control commands and sends data to the master.

PLC Programmable Logic Controller

0098 441 113 193, -000, 09.03

14 INDEX

0098 441 113 193, -000, 09.03

0098 441 113 193, -000, 09.03

0098 441 113 193, -000, 09.03