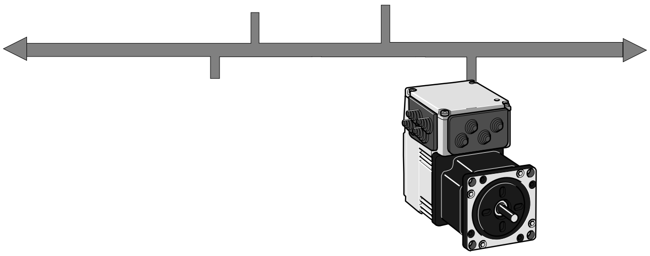
Technical documentation



Field bus protocol for the intelligent compact drive IclA IFx

RS485

Order no.: 0098441113187

Edition: -000, 12.02

BERGER LAHR

a company of Schneider Electric

Important information

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

For more information see chapter 2, "Safety".

Table of contents

5 Installation and setup

6 Examples for field bus operation

7 Error handling

8 Service

9 Glossaries

Index

Supplement

Written conventions and note symbols

Instructions for use

Layout and format:

Introduction to the following instruction steps

① This is the first step.

② This is the second step.

Explanation:

Instructions consist of an introduction and the actual instruction steps.

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

If an instruction step triggers a detectable reaction from the compact drive, the response is described after the step. In this way you can check that the step was correctly completed.

List symbol

Layout and format:

Note on the contents of the list

- 1. list item
- 2. list item
  - 1. list subitem
  - 2. list subitem
- 3. list item

Explanation:

The actual list follows a note on the contents of the list. It can consist of 1 or 2 levels.

The list items are sorted alphanumerically or by priority.

User notes

User notes contain general information, not safety information.

For an explanation of the safety instructions, refer to the chapter on "Safety".

ℹ *This shows additional information on the current subject.*

1 Introduction

1.1 Documentation and literature

Documentation

- Datasheets for IclA in the IclA Intelligent Compact Drives catalog
Order no. 005 9941 2010 001 D
Order no. 005 9941 2010 002 GB
- Controller manuals for IclA compact drives:
- Intelligent Compact Drive Field Bus Stepper Motor IclA IFS6x
Order no. 00 9844 1113 188 D
Order no. 00 9844 1113 189 GB

Literature

- Online command processing for Twin Line units via "Serial Interface" Field Bus
Order no. 9844 1113 135 D
Order no. 9844 1113 134 GB

1.2 Directives and standards

- RS485 Standard
- EIA RS-485

2 Safety

2.1 Qualifications of personnel

Only electrical and controller technicians qualified under IEV 826-09-01 (modified) are authorized to set parameters on, commission and operate the compact drive. The electrical and controller technicians must be familiar with the contents of this manual before starting work with and on the compact drive.

The electrical and controller technicians must have sufficient training, knowledge and experience to recognize and avoid dangers.

The technicians must be familiar with the current standards, regulations and work safety regulations that must be observed while working on and with the compact drive.

2.2 Intended use

2.2.1 Ambient conditions

See the approved environmental conditions described in the data sheet.

2.2.2 Intended use

The compact drive is a variable-speed drive with permanently excited synchronous motor (stepper motor), integrated controller and power electronics. As an option the compact drive can be fitted with a gearbox and a holding brake.

Another option is a Hall sensor, which sends an index pulse and can be used for blocking detection.

The compact drive may be used for industrial applications in the system configuration described with a fixed connection only.

The environment in which the compact drive is to be installed and operated must meet degree of protection IP54 as a minimum.

The compact drive must not be commissioned and operated until it has been installed in conformity with EMC requirements. The compact drive may only be used with the cables and accessories specified by your local dealer.

2.3 Safety instructions

2.3.1 Structure of the safety instructions

All safety instructions are structured to comply with the US standard **ANSI Z535.4**. Under this standard safety instructions are classified into four core elements. A pictogram before the text allows an initial danger classification.

The following general structure is derived from this:

Structure of the safety instructions under ANSI Z535.4

⚠

DANGER LEVEL

Description of the cause and source of the danger

Actions for avoiding the danger

Danger levels

⚠

DANGER

This indicates direct personal danger.

Can lead to serious injuries with fatal consequences if not observed.

⚠

WARNING

Indication of a recognizable danger.

Can result in serious injuries with fatal consequences and destruction of the unit or system component if not observed.

⚠

CAUTION

Indication of a danger.

If this is ignored, minor personal injury and light damage to the unit or system may result.

3 Basics

3.1 RS485 transmission technology

For serial transmission using RS485 technology, one or more compact drives with a higher-order processor form a field bus system. The compact drives are connected in parallel with one another via an RS485 interface and execute commands sent by the higher-order processor via the RS485 interface.

The advantages of RS485 technology include a simple network structure and the fact that a conventional processor with a serial interface (such as a PC) can be used. Commands can even be sent from the processor to the compact drive using simple terminal programs.

3.2 Network topology

The RS485 network comprises a processor as the master device and the compact drives as slave devices.

Master Masters are active bus devices that control the data traffic in the network.

The following are examples of master devices:

- automation devices, e.g. PLCs
- PCs

Slave Slaves are passive bus devices. They receive control commands and supply data to the master.

The following are examples of slave devices:

- compact drives, e.g. IclA IFx devices
- drive controllers, e.g. Twin Line units

3.3 Access methods

The Master-Slave method Data are exchanged with compact drives using the master-slave method. The master device sends a command to the slave, and waits for the slave to acknowledge the command. The slave only transmits if the master sent a command beforehand.

Before transmitting, the master establishes a logical point-to-point connection with the slave device with a polling command. The other slaves of the network cannot be polled while the connection exists.

3.4 Field bus devices in the RS485 network

Berger Lahr field bus devices can be operated on the same field bus. However, the commands for the compact drive are different from the commands of other Berger Lahr devices.

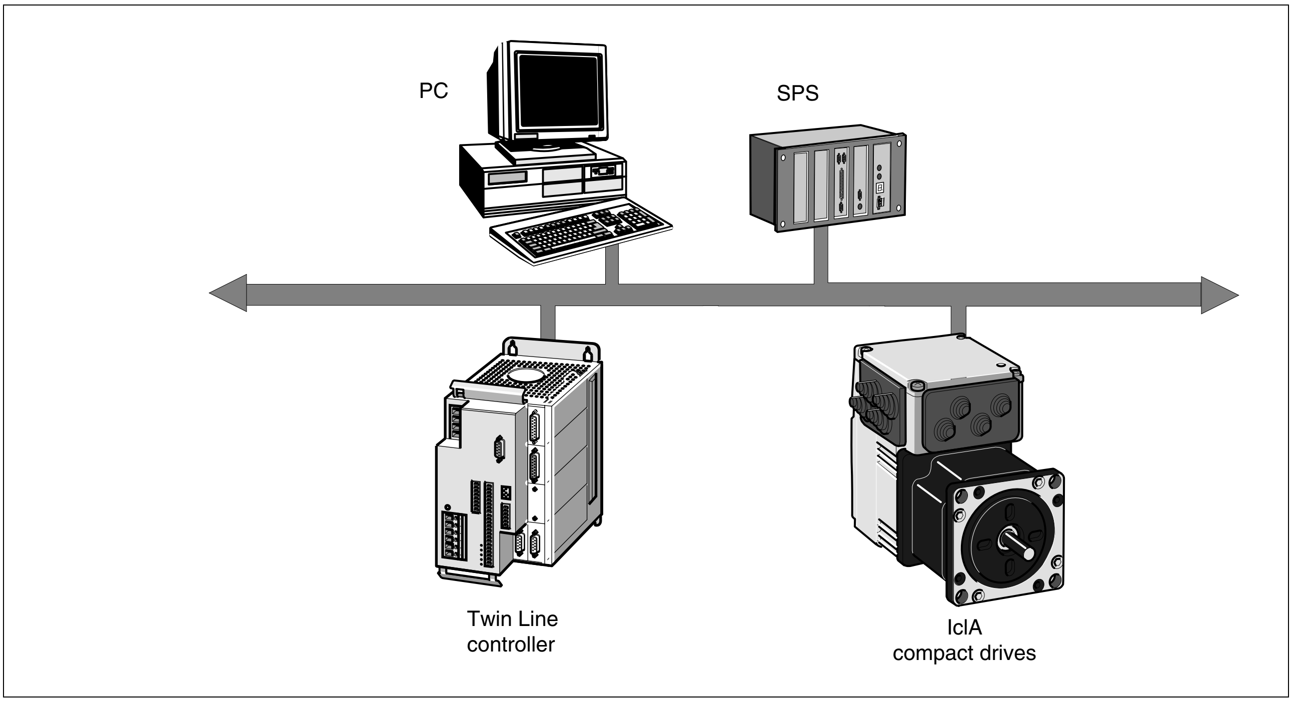


Fig. 3.1 Field bus devices in the network

3.5 Operating modes and functions in field bus operation

This manual only described the protocol for the compact drive. You can find the description of the operating modes, operating functions and all parameters in the controller manual for the compact drive in the "Operation" and "Parameters" chapters:

Operating modes

- speed mode
- point-to-point mode
- referencing

Operating functions

- definition of direction of rotation
- creating movement profile
- Quick-Stop
- fast position capture

Setting options

The following settings can be made over the field bus:

- reading and writing parameters
- monitoring inputs and outputs of the 24-V signal interface
- activating diagnostics and error monitoring functions
field bus mode

4 Communication in the field bus

4.1 Data exchange with the compact drive

4.1.1 Polling

For the master device to be able to communicate with a compact drive, it must poll the compact drive. When the compact drive responds, the communications channel remains open until the connection is broken or another device is polled.

4.1.2 Communication via parameters

The parameters of the compact drive provide the basis for communication between the master device and the compact drive.

An RS485 command polls every parameter via its index and subindex.

Example:

The parameter `PTP.p_absPTP` is used to initiate absolute positioning in point-to-point mode:

• Parameter name:	`PTP.p_absPTP`
• Index:Subindex:	35:01
• Target position:	4650 increments

The parameter is called via an RS485 command:

	Subindex	Index	Target position
…	01	35	4650

The compact drive controller manual contains a list of all parameters. The number format of the parameter values of an RS485 command can likewise be found in the parameter tables in the controller manual.

Example:

For the parameter `Status.UDC_act, 31:20` to convert readings into voltage: **1 V = 10**. A reading of 345 V equates to a voltage of 34.5 V.

4.1.3 Online command processing

The master device sends a command to the compact drive, e.g.:

- initiate movement
- enable operating function
- request information from compact drive

The compact drive executes the command and acknowledges its successful execution.

The exchange of data follows a fixed routine. The process is always viewed from the point of view of the master device.

Data transmitted to the controller:

The master device places a command in the data transmission memory. From there the command is transmitted to the compact drive and executed.

Data received from the compact drive:

The compact drive acknowledges the execution status of the command in the received data. If the command is not executed correctly, the acknowledgement does not contain an error message.

A new command can be sent as soon as the master device receives acknowledgement of the current command.

4.1.4 Commands

The master device transmits control commands or action commands with the transmitted data.

Control commands After sending a control command, the master device expects an acknowledgment from the compact drive confirming whether the processing operation has been successfully executed and completed.

Action commands After an action command, the compact drive merely reports whether a processing operation has been successfully initiated. The master device must then watch continuously for completion of the processing task by requesting and evaluating data received from the compact drive.

4.1.5 Data structure

Information contained in the transmitted and received data:

- action commands or control commands
- administration data for monitoring network operation

The administration data is supplied by the master device user program.

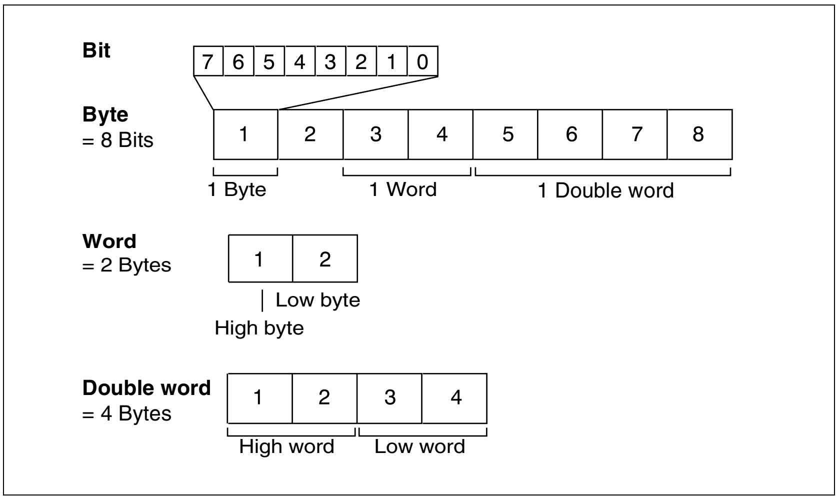


Fig. 4.1 Data structure of the RS485 data bus

Byte 1 contains the acknowledgement information for coordinating the data exchange.

The byte, word and double word values are given as hexadecimal values in this manual.

Hexadecimal values are indicated by an "h" after the number (for example: `31h`).

4.1.6 Polling command

The master device must poll the compact drive with a polling command, before communication can take place via the RS485 bus.

Aufbau `<Address> <CR>`

`<CR>` (CR = carriage return) is a control character that is attached to the transmitted command as a termination code.

Example of a polling command

`<#21> <CR>`

In ASCII code:
`'#','2''1'<CR>`

In hexadecimal notation:
`23h`, `32h`, `31h`, `0Dh`

The compact drive with the corresponding address returns the polling command to the master device as confirmation.

Communication between the master device and the polled compact drive continues until one of the following occurs:

- The connection is interrupted.
- A new slave is polled.

The polling command does not affect the controller status, but it does empty the internal device buffer of the communications interface.

Up to 30 compact drives can be connected and polled in the RS485 bus.

4.1.7 Coding and decoding data

The master device and compact drive transmit data in the RS485 network in ASCII format.

Since the initial data frame is 8 bytes in size, the initial data frame must be converted before transmission into an ASCII data frame of 16 bytes.

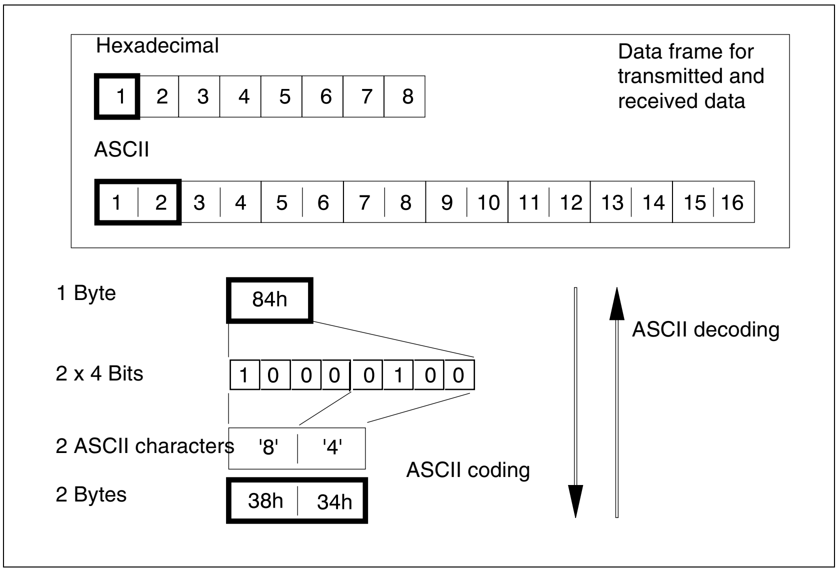


Fig. 4.2 Principle of ASCII coding on the first byte of the 8-byte data frame

IclA IFS6x-RS485 Edition -000 Version 12.02

A `<CR>` (`0Dh`) is attached to the 16-byte ASCII data frame before transmission as a termination code.

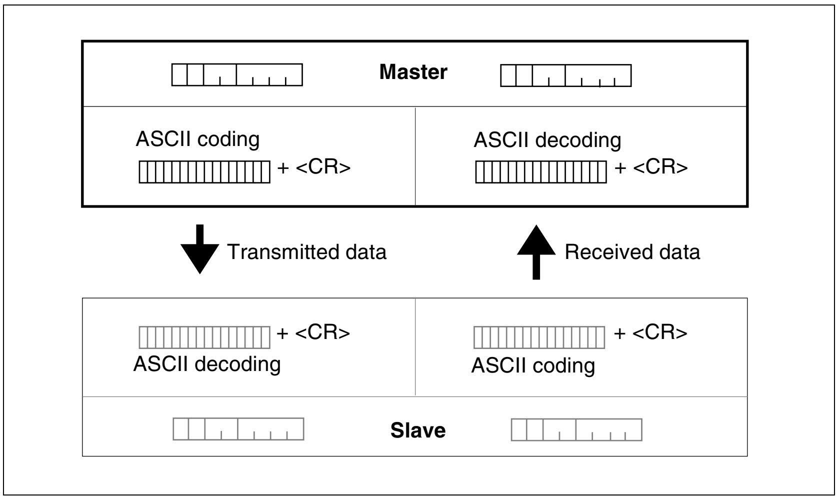


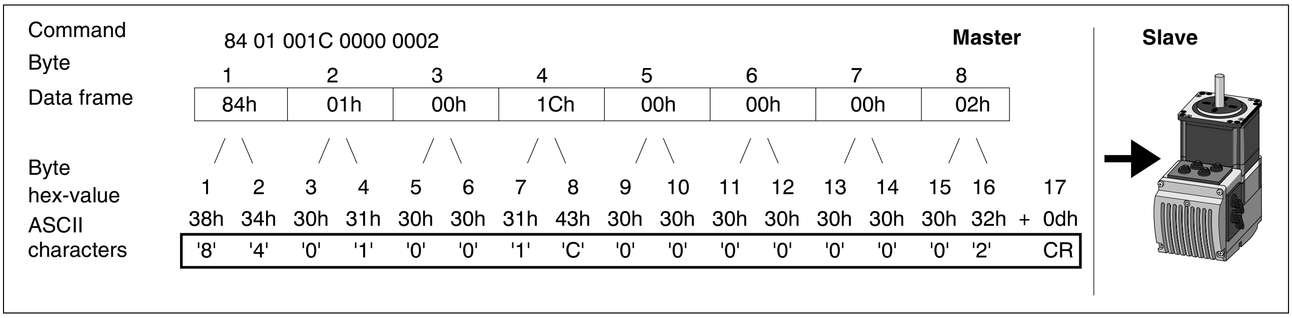
Fig. 4.3 ASCII coding and decoding of the data frame

Example of ASCII coding

A command is to be sent to activate the power amplifier for the compact drive.

The parameter `Commands.driveCtrl, 28:1, Bit 1` is used for this purpose.

The 8-byte data frame is converted into a 16-byte ASCII data frame and sent.



Command 84 01 001C 0000 0002

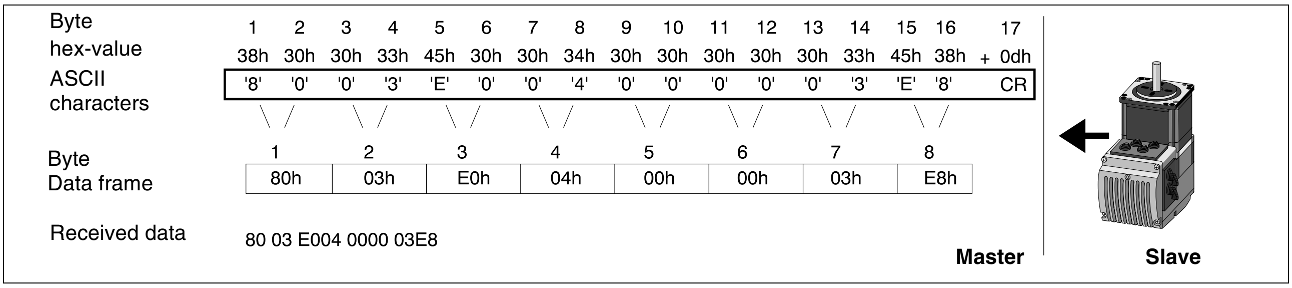
Byte	1	2	3	4	5	6	7	8
Data frame	84h	01h	00h	1Ch	00h	00h	00h	02h

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		17
hex-value	38h	34h	30h	31h	30h	30h	31h	43h	30h	30h	30h	30h	30h	30h	30h	32h	+	0dh
ASCII characters	'8'	'4'	'0'	'1'	'0'	'0'	'1'	'C'	'0'	'0'	'0'	'0'	'0'	'0'	'0'	'2'		CR

Fig. 4.4 Coding transmitted data

The compact drive acknowledges the command by returning the master's received data as a 16-byte ASCII data frame.

The master converts the received 16-byte ASCII data frame back into an 8-byte data frame.



Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
hex-value	38h	30h	30h	33h	45h	30h	30h	34h	30h	30h	30h	30h	30h	33h	45h	38h	+ 0dh
ASCII characters	'8'	'0'	'0'	'3'	'E'	'0'	'0'	'4'	'0'	'0'	'0'	'0'	'0'	'3'	'E'	'8'	CR

Byte	1	2	3	4	5	6	7	8
Data frame	80h	03h	E0h	04h	00h	00h	03h	E8h

Received data 80 03 E004 0000 03E8

Fig. 4.5 Decoding received data

Coding tables

ASCII	hexadecimal	ASCII	hexadecimal
0	30h	8	38h
1	31h	9	39h
2	32h	A	40h
3	33h	B	41h
4	34h	C	42h
5	35h	D	43h
6	36h	E	44h
7	37h	F	45h

ASCII control character	Hexadecimal character	Meaning
#	23h	Start character for polling command
`<CR>`	0Dh	Termination code

4.1.8 Transmitted data frame

With the transmitted data, the master device transmits an action command or a control command to the compact drive.

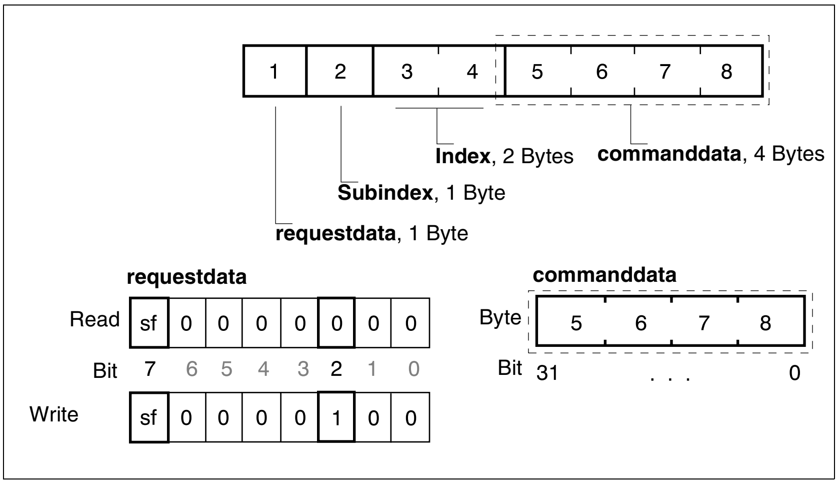


Fig. 4.6 Transmitted data frame

Byte 1: requestdata

Byte 1 contains control information for the following processes:

- acknowledgement of the command
- synchronization
- command as read or write access

Bit	Name	Meaning
2	–	0 = read value The compact drive reads a parameter value and sends the parameter value to the master device. 1 = write value The master device sends the compact drive a parameter value which is written to the controller.
7	sf	sf = sendflag The master device flags a new command for the compact drive by changing the signal of the sf bit. The rf bit is the counterpart bit in the received data.

Bytes 2..4: Subindex, Index

Within the command, the necessary parameter is polled via its index and subindex.

For further information on parameters and parameter values, see

Bytes 5..8: commanddata

Bytes 5 … 8 contain the parameter value (e.g. setpoint speed) sent with the command to the compact drive.

When a parameter of data type INT16 or UINT16 is transmitted, the parameter value is stored only in bytes 7 and 8. A "0" is stored in each of bytes 5 and 6.

4.1.9 Data frame for received data

Content of received data:

- responde of compact drive to a command of the master device
- operating status of compact drive

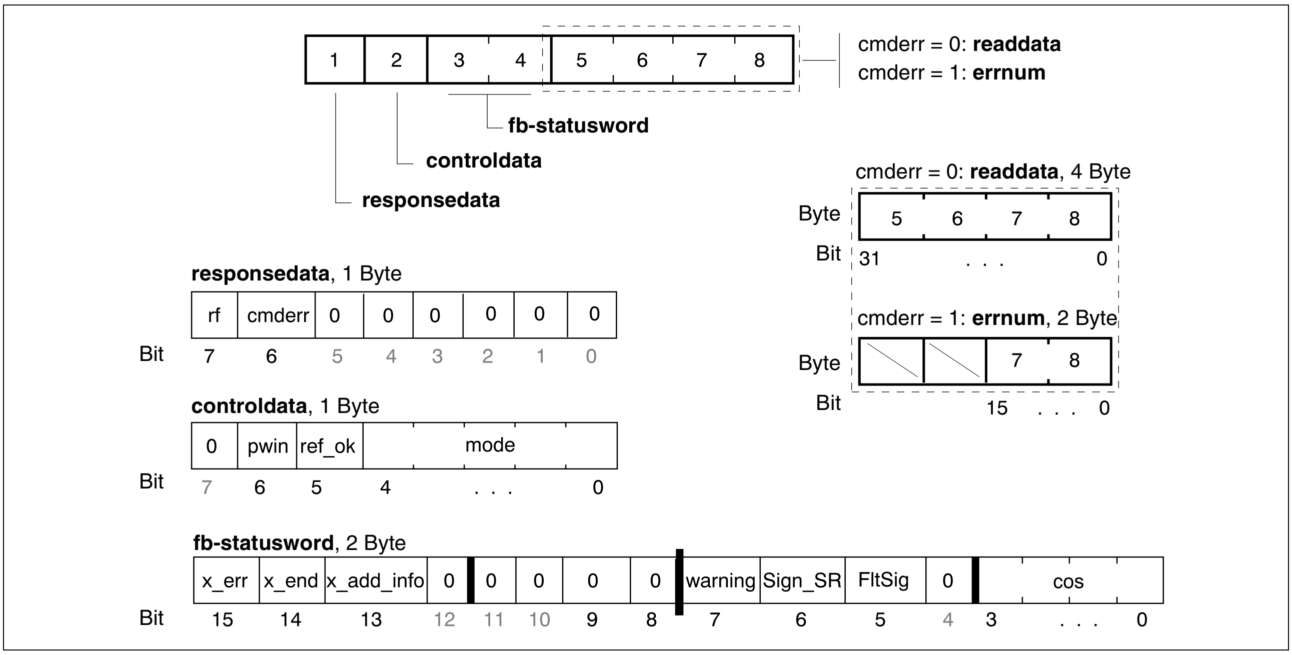


Fig. 4.7 Received data frame

Byte 1: responsedata

Byte 1 contains the following information:

- error detection
- identification of field bus service
- acknowledgement
- synchronization

Bit	Name	Meaning
6	cmderr	cmderr (command error) Signal is only valid after correct acknowledgment of a data package 0 = command was executed without error. 1 = command error, bytes 7 and 8 contain the error number errnum.
7	rf	rf = receiveflag Compact drive's acknowedgment of receipt by changing the rf bit. Thereafter, sf=rf again. As long as rf ≠sf: new command not yet processed.

Byte 2: controldata

The compact drive uses the control data to provide information on the set operating mode and also supplies axis data. The control data can also be read via the LOW byte of the parameter Status.xMode_act, 28:3.

Bit	Name	Meaning
0..4	mode	Current axis mode, bit-coded Example: 3 = 00011 - PTP mode 2: referencing 3: PTP mode 4: speed mode
5	ref_ok	0: no reference point specified 1: axis has been referenced

Bytes 3, 4: fb-statusword

The operating status of the compact drive is monitored with the fb-statusword. This information can also be queried via the LOW word of the Status.driveStat, 28:2 parameter.

Bit	Name	Meaning
0..3	cos	Operating status of the drive. Details on the display and detection of operating states are described in (see chapter 6.2, page 3) and in the controller manual, "Operation" chapter.
5	FltSig	Internal monitoring signals 0: no error detected 1: error detected, cause via parameter Status.FltSig_SR, 28:18
6	Sign_SR	External monitoring signals 0: no error detected 1: error detected, cause via parameter Status.Sign_SR, 28:15
7	Warning	Warning message 0: no warning message 1: warning message, cause via parameter Status.WarnSig,28:10
13	x_add_info	Mode-specific status bit for monitoring the process status
14	x_end	Mode-specific status bit for monitoring the process status
15	X_err	Mode-specific status bit for monitoring the process status

Bytes 5..8: readdata Read data (`readdata`) are control information (e.g. the current motor position) and are requested with a command from the master device. The control information last requested continues to be sent with the next received data until different control information is read.

If no read data have been requested, the compact drive sends the current axis position.

When a parameter of type `INT16` or `UINT16` is transmitted, the value `readdata` is stored only in bytes 7 and 8. In that case the content of bytes 5 and 6 has no significance.

Information on the data types used can be found in the controller manual, chapter 9, "Parameter".

Read data are only transmitted if it was possible to execute the command without errors.

Byte 7, 8: errnum If the commmand is not executed correctly, the compact drive reports an error via `byte 1, bit 6 cmderr`. The cause of the error is stored in bytes 7 and 8, `errnum`.

A list of error numbers can be found in the controller manual, chapter 7, "Diagnosis and troubleshooting".

4.1.10 Abbreviated transmission command

The content of the readdata value can also be requested via the abbreviated control command `<CR>` (`0Dh`).

The compact drive sends the current status information to the master device as acknowledgement. Bits `sf` and `rf` in the acknowledgment mechanism do not change.

4.2 Monitoring and acknowledgement mechanism

4.2.1 Connection monitoring

Two monitoring mechanisms are used continuously to ensure error-free exchange of data:

- timeout monitoring
- received character check

Timeout monitoring

If the master device sends a command or the control character `<CR>` to the compact drive, the master device expects an acknowledgement from the compact drive within 200 ms. If the acknowledgement does not arrive, the compact drive was unable to evaluate the command. The compact drive is then no longer polled.

The time interval for timeout monitoring can be set with parameter `RS485.timeout, 1:11`. Setting a time interval of "0" disables timeout monitoring.

Received character check

The compact drive checks an incoming command for transmission errors. If the compact drive was unable to receive the command correctly, no acknowledgement is sent.

The master device checks the received data for transmission errors and error messages. If a transmission error has occurred, the master device resends the data to the compact drive.

Two bits in the data frame control recognition of a new or resent command:

- bit `sf` in the transmitted data
- bit `rf` in the received data

An error message is recognized by the `cmderr` bit in the received data.

4.2.2 Function of bits `sf` and `rf`

With every polling command, bits `sf` and `rf` start at level "0".

New command from master device

The master device flags a new command by changing the bit `sf`.

Evaluation of a command by the compact drive

The compact drive compares the two bits:

- `sf` ≠ `rf`: Command is new.
- `sf` = `rf`: Command has already been processed.

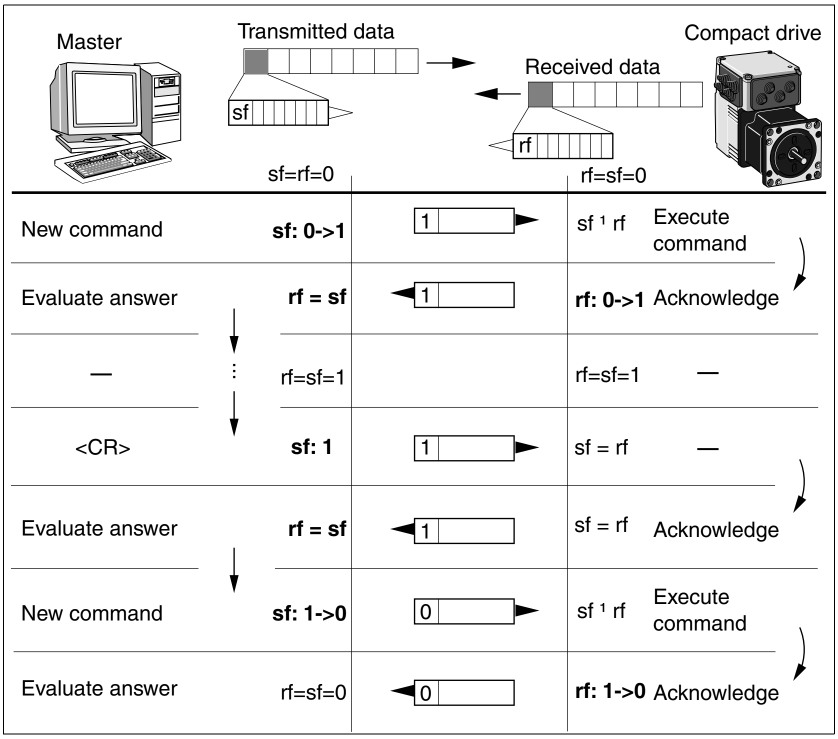


Fig. 4.8 Data exchange and synchronization with bits `sf` and `rf`

Response from the slave

When the command has been executed, the compact drive changes the bit `rf` and transmits the bit `rf` within the response data to the master device.

When the compact drive receives a command that has already been executed, it sends an acknowledgement with the current status information to the master device.

Evaluation of response by the master device

The master device compares the bits `sf` and `rf`.

`sf` = `rf` confirms the execution of the command.

4.2.3 Command error bit cmderr

The command error bit is valid when the command of the master device has been acknowledged:

- `cmderr` = 0: Command has been executed successfully.
- `cmderr` = 1: A synchronous error has occurred.

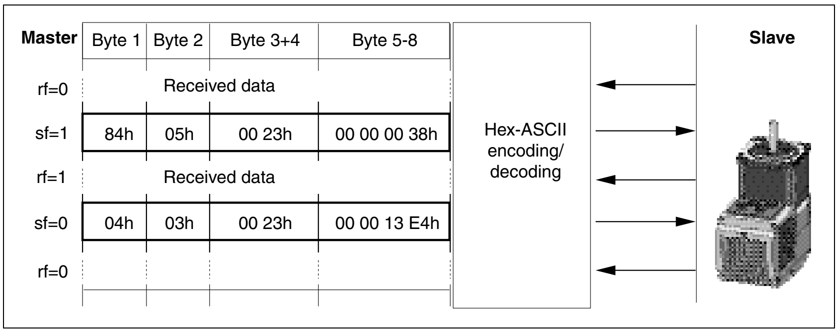
In case of a synchronous error, the cause of the error is stored as an error number `errnum` in bytes 7 and 8.

A list of error numbers can be found in the controller manual, chapter 7 "Diagnosis and troubleshooting".

For further information about synchronous errors, see chapter 7, "Error handling".

4.2.4 Example of a positioning error

A relative positioning command is sent to the compact drive.



Master	Byte 1	Byte 2	Byte 3+4	Byte 5-8
rf=0	Received data			
sf=1	84h	05h	00 23h	00 00 00 38h
rf=1	Received data			
sf=0	04h	03h	00 23h	00 00 13 E4h
rf=0				

Fig. 4.9 Example: Relative positioning command

The first command sets the setpoint speed.

- Byte 1: `requestdata`
  New command ⇒ change bit `sf`
  Write access with bit 2 = 0: 4h
- Bytes 2..4: `Subindex` and `Index`
  Parameter call:
  05h:00 23h: Parameter `PTP.v_tarPTP, 35:5`, setpoint speed
- Bytes 5..8: `commanddata`
  Set value of setpoint speed
  00 00 00 38h = 56 rpm

The second command starts the relative positioning.

- Byte 1: `requestdata`
  New command ⇒ change bit `sf`
  Write access with bit 2 = 0: 4h
- Bytes 2..4: `Subindex` and `Index`
  Parameter call:
  03h:00 23h: Parameter `PTP.p_relPTP, 35:3`, start relative positioning
- Bytes 5..8: `commanddata`
  Set value of positioning path
  00 00 13 E4h = 5092 increments

For further examples, see chapter 6, "Examples for field bus operation".

4.3 Control commands and action commands

The master device can send two types of commands:

- control commands
- action commands

The compact drive reacts differently depending on the type of command.

Control commands

Control commands are executed immediately and end with the sending of the received data to the master device.

The following settings, for example, can be made with control commands:

- switch outputs
- change parameters

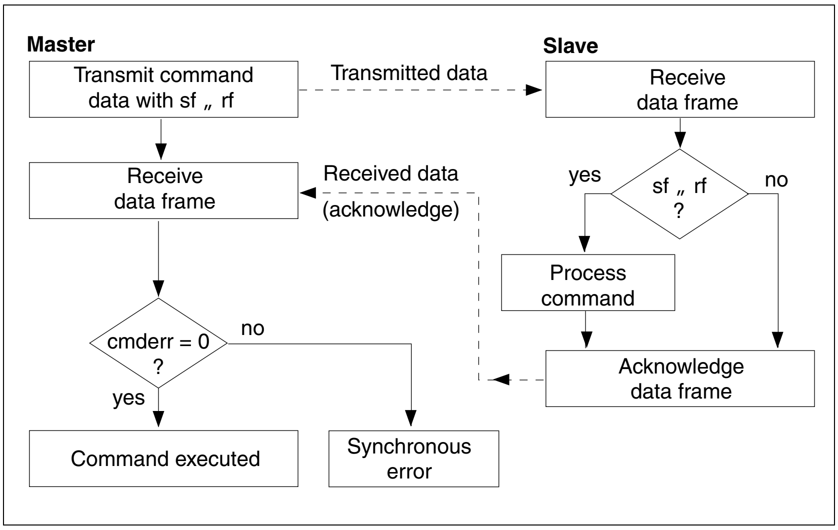


Fig. 4.10 Executing a control command

If a control command could not be correctly executed, the compact drive sets the command error bit `cmderr` to "1" and reports a synchronous error.

Action commands

Action commands start a movement. The compact drive activates the relevant operating mode and loads the required parameters. The start of the movement command is reported to the master device as a command confirmation.

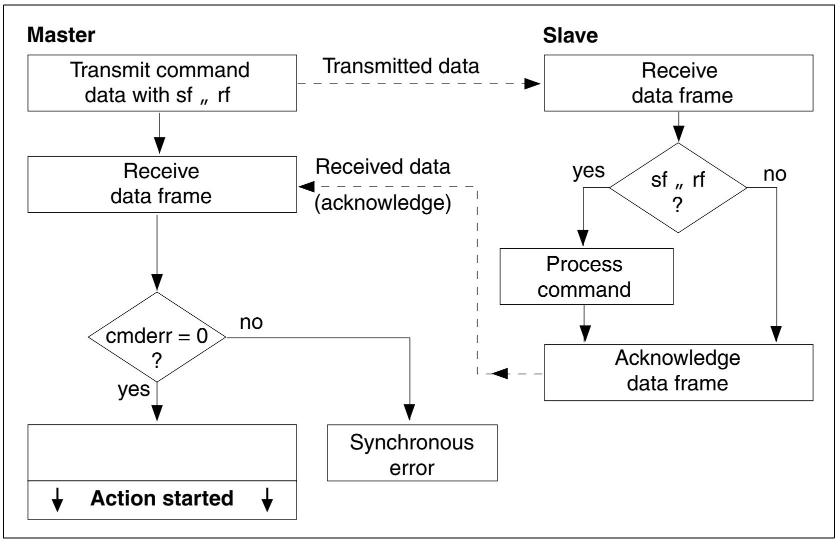


Fig. 4.11 Executing an action command

If an action command could not be correctly executed, the compact drive sets the command error bit `cmderr` to "1" and reports a synchronous error.

Monitoring the operating status

The operating status and the completion of a movement order must be continuously monitored by the master device with the statusword `fb-statusword` in the received data.

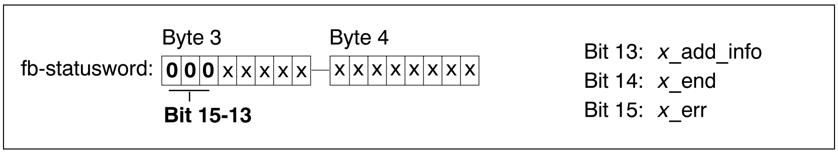


Fig. 4.12 Monitoring the operating status of the compact drive

Status bits The compact drive manages operating states internally for every mode with separate status bits. The compact drive reports only the operating state of the current operating mode via the RS485 bus.

Significance of status bits:

- Bit 13, `x_add_info`
  Mode-specific message
- Bit 14, `x_end`
  Processing status of operating mode
  0: process running
  1: process finished, motor stopped
- Bit 15, `x_err`
  Error status during processing:
  0: fault-free operation
  1. fault has occurred

Using the information on the current mode (parameter `Status.xMode_act, 28:3`), the status message can be evaluated in a mode-specific manner.

Switching operating	Mode	x_add_info	x_end	x_err
Referencing	2	0	ref_end	ref_err
PTP mode	3	Setpoint position reached	motion_end	motion_err
Speed mode	4	Setpoint speed reached	vel_end	vel_err

Table 4.1

As soon as processing is initiated by an action command, bit 14 `x_end` switches to "0". When the process is finished, bit 14 changes back to "1" thereby indicating that further process steps can now be executed.

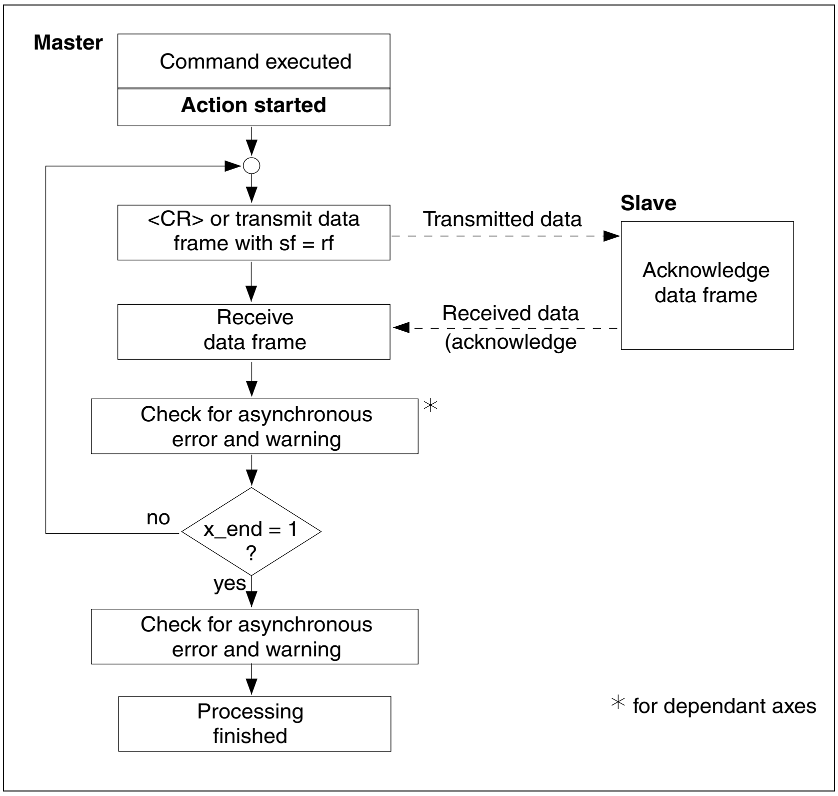


Fig. 4.13 Monitoring execution of the action command

If bit 15 `x_err` changes to "1", an error has occurred. The error must be corrected before further processing can take place.

For further troubleshooting information, see chapter 7, "Error handling".

A check during processing is only required if, for example, additional dependent drives must be stopped immediately.

Influencing ongoing execution

The compact drive can execute control commands during the execution of an action command, for example in order to change the setpoint speed of an ongoing positioning operation. The command error bit `cmderr` provides information on the successful execution of the control command.

5 Installation and setup

5.1 Installation

For information about installation, refer to the controller manual, "Installation" chapter.

Compact drives with DIP switches

Before installing the compact drive in the system, you must set the network address and the baud rate via the DIP switches in the plug housing.

For information about DIP switch settings, refer to the the controller manual, "Installation" chapter.

5.2 Commissioning the network connection

5.2.1 Initiating network operation

Network operation with the compact drive unit is started via a master device. This can be a PLC or a PC with the appropriate application software with which commands can be entered and received data read.

① Switch the compact drive on.

② Poll the compact drive. Use a PC with a terminal program for the initial test.
Enter the following command:
`#01 <CR>`
"01" is the address that has been set for the compact drive.

If the compact drive responds with "#01", the compact drive is ready for data communication.

5.2.2 Troubleshooting

If the compact drive does not respond, check the following settings:

① Has the compact drive been started and the master device switched on?

② Are the cable connections mechanically and electrically sound?

③ Has the address been correctly set on the compact drive? Check the DIP switch settings.

④ Are the same baud rate and the same interface parameters set on the master device and compact drive?

If the compact drive still does not respond:

① Open the plug cover.

② Compare the LED display with the information in the following table.

Error	Error class	Cause of error	Troubleshooting
LED off	–	No power supply.	Check power supply and fuses.
LED flashes at 0.5 Hz. (1 s on, 1 s off)	–	Firmware works without errors, power amplifier inactive.	Check cable connections. Check DIP switch settings.
LED flashes at 6 Hz.	4	Flash checksum wrong.	Reinstall firmware. Replace compact drive.
LED flashes at 10 Hz. Watchdog	4	Hardware error	Switch compact drive off and on again. Replace compact drive.

Refer to the controller manual for further information on the cause of errors and troubleshooting.

6 Examples for field bus operation

6.1 Structure of the examples

The programming examples show practical applications for network operation of compact drives and are designed to complement the function descriptions in the controller manual.

The controller manual contains a basic description of the operating modes and functions.

The programming examples are organized in the following way:

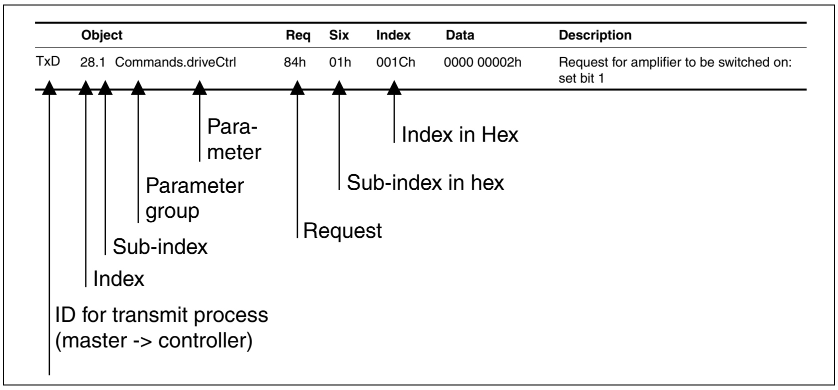
- description of task
- requirements
- commands required in transmitted data frame
- response of compact drive in received frame
- possible restrictions for command execution and other remarks

Transmitted and received data are shown in hexadecimal form.

Commands are shown from the point of view of the master device. The response of the compact drive is only shown if this is needed for further processing. Otherwise a positive acknowledgement is assumed.

Transmitted data

	Object	Req	Six	Index	Data	Description
TxD	28.1 Commands.driveCtrl	84h	01h	001Ch	0000 0002h	Request: Switch on power amplifier: set bit 1



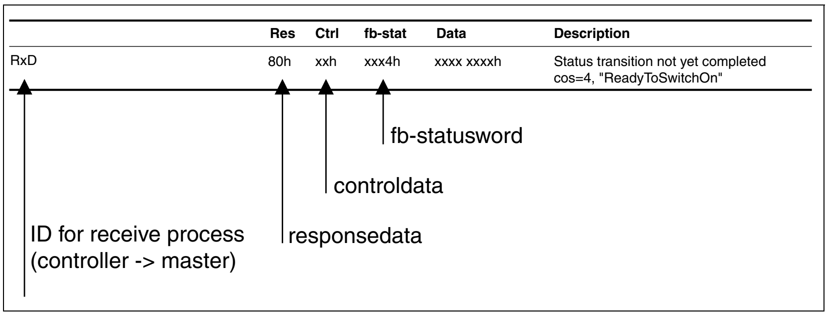
	Object	Req	Six	Index	Data	Description
TxD	28.1 Commands.driveCtrl	84h	01h	001Ch	0000 00002h	Request for amplifier to be switched on: set bit 1

Fig. 6.1 Coding transmitted data

The data type of the parameter values is indicated in the parameter description of the compact drive. When a value of data type `INT16` or `UINT16` is transmitted, this value is saved to bytes 7 and 8. You must enter "0" in bytes 5 and 6.

Received data

	Res	Ctrl	fb-statusword	Data	Description
RxD	80h	xxh	xxx4h	xxxx xxxxh	Status transition still unsuccessful; c0s = 4, "Ready to switch on"



	Res	Ctrl	fb-stat	Data	Description
RxD	80h	xxh	xxx4h	xxxx xxxxh	Status transition not yet completed cos=4, "ReadyToSwitchOn"

Fig. 6.2 Coding received data

The data type of the parameter values is indicated in the parameter description of the compact drive. When a value of data type `INT16` or `UINT16` is transmitted, this value is saved to bytes 7 and 8. "0" is entered in bytes 5 and 6.

In `responsedata`, an error is identified by `cmderr = 1`. The error number of a synchronous error message is saved as data type `UINT16`.

Values which are not relevant to the example are represented by "x".

Units of valueinformation

If read or write values are stated in user-defined units [usr], calibration factors must be applied to these user-defined units.

Example:

For the parameter `Status.UDC_act, 31:20` to convert readings into voltage: **1 V = 10**. For example, a reading of 345 V equates to a voltage of 34.5 V.

Acknowledgment bits

In all programming examples, the following signal level is assumed for the acknowledgement bits before the first transmit command:

- `rf` = 0
- `sf` = 0

The first command must therefore transmit with `sf` = 1. This also applies when the description of a subject is split across several examples. A new command is recognized from a change to the bit.

Data frame

In the programming examples, only the 8 byte frame for compact drives is shown. The control byte specific to the field bus must be added by the master device user program in accordance with the field bus protocol.

6.2 Operating status

The compact drive must be ready to start and be correctly initialized before an operating mode can be started.

Detailed information about operating states and a status diagram can be found in the controller manual, chapter 6, "Operation".

6.2.1 Check operating status

The operating status is evaluated via the RS485 bus with the first four bits in the statusword `fb-statusword`. The status word is transmitted with every received data set.

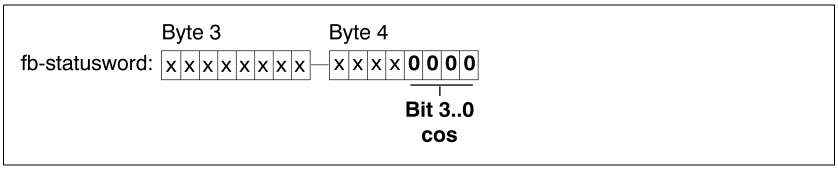


Fig. 6.3 Displaying the operating status of the compact drive

Bit 3..0	Operating state	Meaning
-	-	24 V switched on
...0001	1 - Start	Initialization of the unit electronics
...0010	2 - Not ready to switch on	Power amplifier not ready to switch on.
...0011	3 - Switch on disabled	Power amplifier cannot be switched on.
...0100	4 - Ready to switch on	Power amplifier is ready to switch on.
...0110	6 - Operation enable	Unit is working in the set operating mode.
...0111	7 – Quick Stop active	Quick Stop is executed.
...1000 ...1001	8 - Fault reaction active 9 - Fault	Fault reaction activated Fault display

Operating states 0..3, 8 and 9 are transitional states which the compact drive does not remain in if it is operating correctly.

If the compact drive remains in operating modes 1, 2 or 3 after the power supply is switched on, an error has occurred during the initialization of the compact drive.

Troubleshooting information can be found in the controller manual, chapter 7, page 1.

6.2.2 Changing operating status

The operating status of the compact drive is transmitted with bits cos in fb-statusword. The coding corresponds to the operating states of the compact drive.

Whether you can change an operating status depends on the current operating status.

Example

The operating status "Ready to switch on" can only be enabled if the following conditions are satisfied:

- The compact drive has been started up normally.
- No errors are present.

A change of operating status is enabled with the parameter Commands.driveCrtl, 28:1. The value is always "0", so that write access to a bit automatically triggers slope change 0 → 1.

Bit 3..0	Control word	Meaning
0 0 0 1	Disable	Switching off the power amplifier
0 0 1 0	Enable	Switching on the power amplifier
0 1 0 0	Quick Stop	Trigger stop with Quick Stop
1 0 0 0	Fault Reset	Acknowledging fault message

Example 1 Switching on the power amplifier

Requirements

- The compact drive must be in "Ready to switch on" mode.
- The parameter settings on the compact drive must be correct.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	28.1 Commands.driveCtrl	84h	01h	001Ch	0000 0002h	Request to switch on power amplifier: set bit 1

Received data

		Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	xxx4h	xxxx xxxxh	Status transition still unsuccessful; cos = 4, "Ready to switch on"

Once the status transition has been executed successfully, the compact drive reports:

		Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	xxx6h	xxxx xxxxh	Status transition complete: cos = 6, "Operation Enable"

Comments

As long as "Operation Enable" mode is not reached, bit 15 (`x_err`) is enabled in status word `fb-statusword`.

Operating state	fb-statusword `x_err` (bit 15)
≠ Operation Enable	1
= Operation Enable	0

Example 2 Switching off the power amplifier

Requirements

- The compact drive must be in "Operation Enable" mode.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	28.1 Commands.driveCtrl	84h	01h	001Ch	0000 0001h	Request to switch on power amplifier: set bit 0

		Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	xxx6h	xxxx xxxxh	Status transition still unsuccessful: cos = 6, "Operation Enable"

Once the status transition has been executed successfully, the compact drive reports:

		Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	xxx4h	xxxx xxxxh	Status transition still unsuccessful; cos = 4, "Ready to switch on"

Comments

As soon as the compact drive exits "Operation Enable" mode, bit 15 (`x_err`) is enabled in status word `fb-statusword`.

Operating state	fb-statusword `x_err` (bit 15)
≠ Operation Enable	1
= Operation Enable	0

Example 3 Movement interruption by software stop

Requirements

- The compact drive must be in "Operation Enable" mode.
- All the settings required for the operating mode must have been entered.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	28.1 Commands.driveCtrl	84h	01h	001Ch	0000 0004h	Request to switch on power amplifier: set bit 0

Message after "Quick Stop Active" mode has been activated successfully:

	Res	Ctrl	fb-statusword	Data	Description
RxD	80h	xxh	8047h	xxxx xxxxh	In fb-statusword: x_err=1, Sign_SR=1, cos=7: "QuickStopActive"

Message after compact drive stops:

	Res	Ctrl	fb-statusword	Data	Description
RxD	80h	xxh	C047h	xxxx xxxxh	In fb-statusword: x_err = 1, x_end = 1, Sign_SR = 1, cos=7: "Quick Stop Active"

Comments

If the compact drive is at a standstill, the interruption status can be canceled with "Fault reset".

For further information about "Fault Reset", see section 6.6.4, page 25.

6.3 Setting processing parameters

6.3.1 Setting mode-independent processing parameters

The mode-independent processing parameters apply to all operating modes.

The following parameter groups contain mode-independent processing parameters:

- Capture
- Commands
- ErrMem0
- I/O
- Motion
- RS485
- Settings

Example 4 Enable signal for monitoring parameter

Task

Enable limit switch monitoring.

Requirements

- The compact drive must be in "Ready to switch on" mode.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	28.13 Settings.SignEnabl	84h	0Dh	001Ch	0000 0003h	Enable monitoring inputs LIMP and LIMN

6.3.2 Setting mode-dependent processing parameters

The following parameter groups contain mode-dependent processing parameters:

- Referencing
  `Homing`
- Point-to-point mode
  `PTP`
- Speed mode
  `VEL`

Example 5 Set the setpoint speed for point-to-point operation

Task

Set a setpoint speed of 500 rpm for point-to-point operation.

Requirements

- The compact drive must be in "Operation Enable" mode.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	35:5 PTP.v_tarPTP	84h	05h	0023h	0000 01F4h	Setting the fast setpoint speed to 500 rpm = 1F4h

Comments

You can change the setpoint speed before and during positioning.

6.4 Reading device information

After a read access the following actions are executed:

- The current parameter value is output.
- The parameter value requested upon read access is output upon each further read or write access until such time as a different parameter value is requested.

Whether a parameter value can be read and for the data type of the parameter value, refer to chapter 9 of the manual, "Parameters". All values marked "R" in the "R/W" column of the parameter list can be read. The data type is shown in the "Value Range" column.

6.4.1 Reading parameter settings

The current value of a parameter can be read through read access.

Example 6 Reading the setpoint speed of the point-to-point operation

Requirements

- The parameter value must be available in the current operating status of the compact drive.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	35:5 PTP.v_tarPTP	80h	05h	0023h	0000 0000h	Reading the setpoint speed of the point-to-point operation

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	xxxxh	0000 00B4h	Setpoint speed in bytes 5 to 8: B4h = 180 rpm

Comments

In the event of a read access to a 16-bit value, the parameter value is in bytes 7 and 8; bytes 5 and 6 have a value of "0".

6.4.2 Reading status information

Various read values are available in the `status` parameter group relating to the processing status of the compact drive.

There is mode-dependent and mode-independent status information. Mode-independent status information is communicated in `controldata` and `fb-statusword`; the coding corresponds to the assignment of the `Status.driveStat, 28:2` parameter.

Example 7 Reading the actual motor speed

Task

Read the actual motor speed. The actual speed is stored in the `Status.n_act, 31:6` parameter.

Requirements

- The parameter value must be available in the current operating status of the compact drive.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	31:6 Status.n_act	80h	06h	001Fh	0000 0000h	Reading the actual motor speed

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	xxxxh	0000 03E8h	Actual speed in bytes 7 to 8: 3E8h = 1000 rpm

Comments

In the event of a read access to a 16-bit value, the parameter value is in bytes 7 and 8; bytes 5 and 6 have a value of "0".

6.5 Use of operating modes

Your compact drive can work in different operating modes.
Mode-dependent and mode-independent parameters are available for configuration.

6.5.1 Determining the operating mode and processing status

The current operating mode can be read from the parameter `Status.xMode_act, 28:3`.

The processing status can be determined from the bits `x_err` (bit 15) and `x_end` (bit 14) in `fb-statusword`.

x_err	x_end	Processing status
0	0	Process active and no fault
0	1	Process ended and no fault
1	0	Process active and fault detected
1	1	Process ended and fault detected

If an asynchronous error occurs during processing, `x_err` is set to `1` immediately and the compact drive is stopped. After the compact drive stops, `x_end` is set to `1`.

6.5.2 Point-to-point mode

A pallet must be moved two stations on an endless conveyor belt at a speed of 200 rpm. All units are stated in increments [inc].

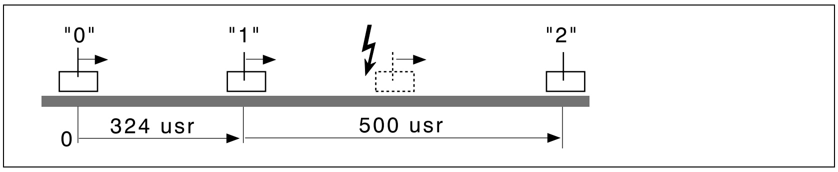


Fig. 6.4 Moving the pallet

The first station is situated 324 inc. from the reference position "0". The station is approached with absolute positioning. The second station is a further 500 inc. away and is reached by relative positioning.

During travel to the second station, an interruption is triggered by the STOP signal. After correction of the fault, the interrupted movement must be resumed and completed.

Requirements

- All settings must be have been entered via the operating functions.
- The reference position "0" must be defined.
  For information, see section 6.5.4, "Referencing", page 6-18.
- The compact drive must be at a standstill.

Example 8 Performing absolute positioning

Task

Move the pallet at the setpoint speed of 200 rpm to position 324 inc.

Requirements

- The compact drive must be in "Operation Enable" mode.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	35:5 PTP.v_tarPTP	84h	05h	0023h	0000 00C8h	Set the setpoint speed to 200 rpm = 00C8h rpm

	Object	Req	Six	Index	Data	Description
TxD	35:1 PTP.p_absPTP	04h	01h	0023h	0000 0144h	Start absolute positioning to 324 inc. = 0144h inc.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	x3h	6006h	xxxx xxxxh	Motor movement active

The processing status can be determined from the bit x_end in fb-statusword. When processing is ended, x_end is set to 1.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	2xh	6006h	xxxx xxxxh	Motor movement complete

Comments

The setpoint speed can be changed at any time during the movement through a write access to PTP.v_tarPTP, 35:5.

Example 9 Performing relative positioning

Task

Continue the movement in example 9, moving the pallet on by +500 inc.

Requirements

- The compact drive must be in "Operation Enable" mode.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	35.3 PTP.p_relPTP	84h	03h	0023h	0000 01F4h	Start relative positioning by +500 inc. = 0144h inc.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	x3h	0006h	xxxx xxxxh	Motor movement active

The processing status can be determined from the bit `x_end` in `fb-statusword`. When processing is ended, `x_end` is set to `1`.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	2xh	6006h	xxxx xxxxh	Motor movement complete

Comments

The setpoint speed can be changed at any time during the movement through a write access to `PTP.v_tarPTP, 35:5`.

Example 10 Continuing interrupted movement

Task

The movement in example 9 is interrupted by the signal `STOP`. Continue the movement.

Requirements

- Positioning must have been interrupted by the `STOP` signal.
- The cause of the error must have been corrected, i.e. the `STOP` signal is no longer active.
- The compact drive must have been set to "Operation Enable" mode by a "Fault Reset".

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	35:4 PTP.continue	84h	04h	0023h	0000 0000h	Initiate resumption of interrupted positioning

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	x3h	0006h	xxxx xxxxh	Motor movement active

The processing status can be determined from the bit `x_end` in `fb-statusword`. When processing is ended, `x_end` is set to `1`.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	23h	6006h	xxxx xxxxh	Motor movement complete

Comments

The setpoint speed can be changed at any time during the movement through a write access to `PTP.v_tarPTP, 35:5`.

6.5.3 Speed mode

Example 11 Setting the setpoint speed

Task

Set the setpoint speed to 2000 rpm and monitor attainment of the setpoint speed.

Requirements

- All settings must have been entered via the operating functions.
- The compact drive must be in "Operation Enable" mode.
- The compact drive must be at a standstill.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	36:1 VEL.velocity	84h	01h	0024h	0000 07D0h	Set the setpoint speed: +2000 rpm = 07D0h rpm

Attainment of the setpoint speed can be checked via `x_add_info` in `fb-statusword`. If the setpoint speed has been reached, `x_add_info` is set to 1.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	x4h	2006h	xxxx xxxxh	Motor movement active, setpoint speed reached

Example 12 Bringing the compact drive to standstill

Requirements

- The compact drive must be running at a setpoint speed > 0.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	36:1 VEL.velocity	04h	01h	0024h	0000 0000h	Set the setpoint speed to 0 rpm = 0h rpm

The processing status can be determined from the bit `x_end` in `fb-statusword`. When processing is ended, `x_end` is set to `1`.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	x4h	6xx6h	xxxx xxxxh	Motor movement complete

6.5.4 Referencing

The following conditions must be met to be able to run the "referencing" mode examples:

- All necessary settings must have been entered via the functions.
- The compact drive must be in "Operation Enable" mode.
- The compact drive must be at a standstill.

Example 13 Dimension setting

Task

Set the current position of the compact drive to a value of 1000 inc. The new value serves as a reference for further movements.

Requirements

- The compact drive must be at a standstill.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	40:3 Homing.startSetp	84h	03h	0028h	0000 03E8h	Dimension setting position +1000 inc = 03E8h inc

Dimension setting is run immediately when the parameter is called. The success of the operation can be checked with the `x_end` and `x_err` bits in `fb-statusword`.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	x2h	4006h	xxxx xxxxh	Motor movement active, setpoint speed reached

After successful referencing, bit ref_ok is set to 1 in `controldata`.

Example 14 Performing a referencing movement

Task

Carry out a referencing movement in a negative direction to the reference switch REF. The search movement to the reference switch REF should be run at the setpoint speed of 500 rpm.

Requirements

- The compact drive must be at a standstill.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	40:4 Homing.v_Home	84h	04h	0028h	0000 01F4h	Setpoint speed for search of reference switch REF: 500 rpm = 01F4h rpm

	Object	Req	Six	Index	Data	Description
TxD	40:1 Homing.startHome	04h	01h	0028h	0000 0003h	Start referencing movement to the reference switch REF in a negative direction

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	x0h	0006h	xxxx xxxxh	Referencing movement active

Dimension setting is run immediately when the parameter is called. The success of the operation can be checked with the x_end and x_err bits in fb-statusword.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	x2h	4006h	xxxx xxxxh	Referencing movement complete.

After successful referencing, bit ref_ok is set to 1 in controldata.

6.6 Examples of fault processing

Detailed information on fault handling in field bus mode can be found in chapter 7, page 1.

6.6.1 Synchronous errors

Synchronous faults only occur in response to a command. When a command is sent, it is immediately checked to see whether it can be executed correctly. If not, the compact drive returns an error number in response to the command and bit `cmderr` is set to 1 in `responsedata`. This action does not change the operation status of the compact drive.

Example 15 Generating a synchronous error

Task

Execute a write access to a non-existent parameter 0:255.

Requirements

- The compact drive must be in "Operation Enable" mode.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	0:255 non-existent parameter	84h	FFh	0000h	xxxx xxxxh	Write access to a non-existent parameter

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		C0h	xxh	6006h	0000 0130h	cdmerr = The error number (errnum) is stored in bytes 7 and 8.

6.6.2 Asynchronous errors

Asynchronous errors are independent of the transmitted command. If the external and internal monitoring signals detect a fault, the compact drive switches to an error status. The error status depends on the error class and in the case of cyclical transmission is stored in `fb-statusword`.

Example 16 **Internal monitoring signals**

Task

Generate an internal error and subsequently determine cause of the error.

Requirements

- The compact drive must be in "Operation Enable" mode.

Transmitted and received data

○ Reduce the power supply of the compact drive to below 15 V.

The compact drive reports an error.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	xxh	E029h	xxxx xxxxh	In fb-statusword: x_err = 1, x_end = 1, SignSr = 1, cos = 9: "Fault"

Now determine the cause of the error.

	Object	Req	Six	Index	Data	Description
TxD	28:18 Status.FltSig_SR	80h	12h	001Ch	xxxx xxxxh	

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	E029h	0000 0001h	Internal monitoring signal: Bit 0 active: cause of error "Undervoltage 1 power supply"

If bit 7 in `fb-statusword` is enabled, you can read the warning messages with the parameter `Status.WarnSig, 28:10`.

If the compact drive is at a standstill, the interruption status can be canceled with "Fault reset".

For further information about "Fault Reset", see section 6.6.4, page 25.

Example 17 External monitoring signals

Task

A positioning operation was interrupted by a light barrier at the STOP input. Determine the cause of the error.

Requirements

- The STOP external monitoring signal must be enabled (parameter Settings.SignEnabl, 28:13).
- The compact drive must be in "Quick Stop Active" mode.

Transmitted and received data

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	xxh	E047h	xxxx xxxxh	In fb-statusword: x_err = 1, x_end = 1, SignSr = 1, cos = 7: "Quick Stop Active"

Now determine the cause of the error.

	Object	Req	Six	Index	Data	Description
TxD	28:15 Status.Sign_SR	80h	0Fh	001Ch	xxxx xxxxh	

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	E047hh	0000 0004h	External monitoring signal: Bit 2 active: cause of error STOP

If the compact drive is at a standstill, the interruption status can be canceled with "Fault reset".

For further information about "Fault Reset", see section 6.6.4, page 25.

6.6.3 Other faults

If `fb-statusword` has the following bit values, this indicates an internal controller error:

- bit 15: `x_err = 1`
- bit 6: `SignSr = 0`
- bit 5: `FltSig = 0`

The error number of the cause of the interruption can be determined with the parameter `Status.StopFault, 32:7`.

Errors which cause the compact drive to exit "Operation Enable" mode are entered as follows:

- in the bit line for external and/or internal monitoringsignals
- cause of last interruption
- in the error memory

You can access the cause of the last status change directly (parameter `Status.StopFault, 28:15`).

Example 18 Determining error number

Task

Determine the cause of the last interruption from the error memory of the compact drive.

Requirements

- As a result of activating the `STOP` input, the compact drive must be in "Quick Stop Active" mode.

Transmitted and received data

Now determine the cause of the error.

	Object	Req	Six	Index	Data	Description
TxD	28:15 Status.StopFault	80h	07h	0020h	0000 0000h	Query the cause of the last interruption

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	Cx47h	xxxx 015Ah	In fb-statusword: x_err = 1, x_end = 1, SignSr = 1, error number 015Ah in bytes 7 and 8

If the error has been reset or the 24 V power supply was switched off and on again, the last cause has been deleted. If more than one error was detected, only the error that resulted in the drive being interrupted, i.e. that caused the drive to exit "OperationEnable" status, is saved as the cause of the interruption. Any subsequently occurring faults are saved in the standard error memory in order of occurrence.

6.6.4 Fault Reset

If the cause of an error is no longer present, the "Quick-Stop" or "Fault" states can be exited with a "Fault Reset".

A "Fault Reset" can only be executed if the compact drive is at a standstill (`x_end = 1`).

Once "Fault Reset" has been successfully performed, the error number of the last interruption is deleted from the parameter `Status.StopFault, 28:15`.

The entry in the normal error memory is not deleted by a "Fault Reset".

Example 19 Fault reset

Task

Reset the interruption caused by the `STOP` switch in example 18.

Requirements

- As a result of activating the `STOP` input, the compact drive must be in "Quick Stop Active" mode.
- The cause of the error is no longer active, i.e. the `STOP` input is deactivated.
- The compact drive must be at a standstill (`x_end = 1`)

Transmitted and received data

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	xxh	E047h	xxxx xxxxh	In fb-statusword: x_err = 1, x_end = 1, SignSr = 1, cos = 7: "Quick Stop Active"

	Object	Req	Six	Index	Data	Description
TxD	28:1 Commands.driveCtrl	84h	01h	001Ch	0000 0008h	Request: Bit 3 "Fault Reset"

If the fault has been successfully reset, the compact drive exits error status. The compact drive enters "Operation Enable" status.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	6xx6h	xxxx xxxxh	In fb-statusword: x_err = 0, cos = 6 "Operation Enable"

6.6.5 Reading and deleting the error memory

All error messages are entered in the error memory of the compact drive starting with the oldest (ErrMem0). Up to 7 different error messages can be stored.

If an entry in the error memory is empty, the error number is "0".

Errors of error class 1 or higher are stored in non-volatile memory and are not deleted when the compact drive is switched off.

The following information is stored in the error memory:

- error number
- error class
- age of the error (= how many times the compact drive has been switched on since the error occurred)
- error frequency (= how many times in succession an error has occurred)
- additional information

The error memory can be cleared with the parameter `Commands.del_err, 32:2`.

Example 20 **Reading the error memory**

Task

Read all information concerning the oldest error from the error memory.

Requirements

- The oldest fault entry is a movement interruption caused by the `STOP` switch.
- The compact drive must be in "Quick Stop Active" mode.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	900:1 ErrMem0.ErrNum	80h	01h	0384h	0000 0000h	Request: error number in ErrMem0

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	Cx47h	0000 015Ah	In fb-statusword: x_err = 1, x_end = 1, SignSr = 1, error number 015Ah: "Interruption/ Quick Stop caused by STOP"

	Object	Req	Six	Index	Data	Description
TxD	900:2 ErrMem0.Class	00h	02h	0384h	0000 0000h	Request: error class

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	xxh	Cx47h	0000 0001h	Error class 1 (in bytes 7 and 8)

	Object	Req	Six	Index	Data	Description
TxD	900:3 ErrMem0.Age	80h	03h	0384h	0000 0000h	Request: age of error

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	Cx47h	0000 000h	Age of error: value 0 = compact drive has not been switched off since the error occurred.

	Object	Req	Six	Index	Data	Description
TxD	900:4 ErrMem0.Repeat	00h	04h	0384h	0000 0000h	Request: error frequency

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00h	xxh	Cx47h	0000 0005h	Value 5 = error has occurred 6 times in succession

	Object	Req	Six	Index	Data	Description
TxD	900:5 ErrMem0.Age	80h	05h	0384h	0000 0000h	Request: additional information for assessing fault

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		80h	xxh	Cx47h	0000 0000h	Value 0: no further information available

Comments

Access to the other entries in the error memory (`ErrMem1..6`) is similar.

To do so, change the index of the parameter (e.g.: for `ErrMem5` ⇒ Index = 905 = 389h)

Example 21 Clearing the error memory

Task

Clear all entries from the error memory.

Transmitted and received data

	Object	Req	Six	Index	Data	Description
TxD	32:2 Commands.del_err	00h	02h	0020h	0000 0001h	Request: clear contents of error memory

Comments

Selective clearing of the error memory is not possible.

The error memory will be cleared even if the cause of the error is still active.

7 Error handling

7.1 Error messages

The master device receives error messages with the received data during network operation.

2 types of errors are distinguished:

- synchronous errors
- asynchronous errors

Synchronous errors The master receives the message regarding a synchronous error directly from the drive if the transmitted command could not be executed. A synchronous error is detected by the `cmderr` bit.

Asynchronous errors Asynchronous errors are reported by the monitoring devices in the compact drive as soon as a fault in the device occurs. The master continuously monitors the status word `fb_statusword` to detect asynchronous errors.

The compact drive supplies the status information cyclically in bus frequency.

7.2 Synchronous errors

A synchronous error is evaluated with the `cmderr` command error bit in the first byte of the received data:

- `cmderr = 0`: Command has been successfully carried out.
- `cmderr = 1`: Error has occurred.

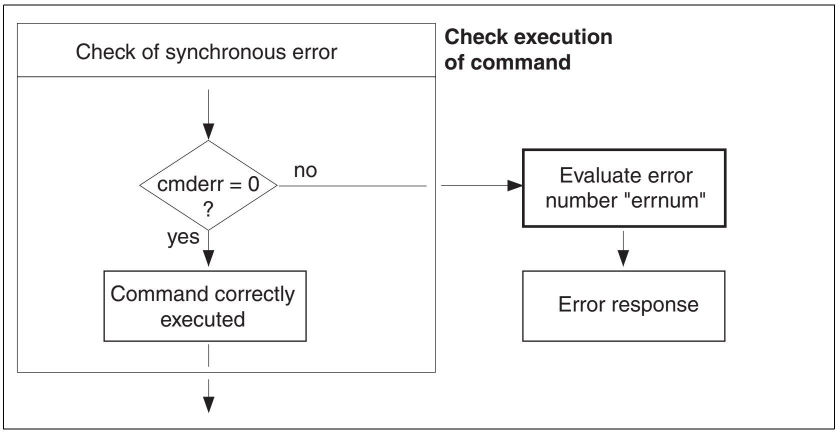

Check of synchronous error
Check execution of command
cmderr = 0 ?
no
yes
Command correctly executed
Evaluate error number "errnum"
Error response

Fig. 7.1 Evaluation of synchronous errors

Causes of errors

Possible causes of a synchronous error are:

- unknown command, syntax error or incorrect transmitted data frame
- parameter value outside the permissible value range
- illegal action or control command during a running process
- error while executing an action or control command

The `cmderr` bit is only valid when the command has been acknowledged. The drive returns an error number `errnum` in bytes 7 and 8 of the received data. The cause of the error can be determined from this number.

A table of error numbers appears in the controller manual, in the "Diagnostics and Troubleshooting" chapter.

7.3 Asynchronous error

Error bits must be monitored in the `fb_statusword` status word to detect an asynchronous error.

- Bit 15, `x_err`
  Error status during processing
  Evaluate cause via bit 5 and bit 6
- Bit 7, `warning`
  Warning message from compact drive
  e.g. power amplifier temperature warning
- Bit 6, `Sign_SR`
  Message from external monitoring signal
  e.g. movement interruption caused by `STOP` input
- Bit 5, `FltSig`
  Message from internal monitoring signal
  e.g. power amplifier overheating.

Signal status "1" indicates an error or warning message.

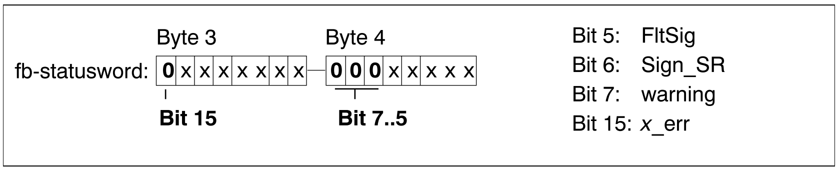


Fig. 7.2 Status word for evaluating asynchronous errors

Warning message In a warning message the movement command processing continues and the error information is input to the `Status.WarnSig, 28:10` parameter.

Error message If the compact drive sets the `x_err` signal, it interrupts the movement immediately and responds either by braking or by immediately shutting off the power amplifier, depending on the error class. The `FltSig` or `Sign_Sr` bits are set together with the `x_err` bit. The meaning of the error message must be determined via the corresponding parameter.

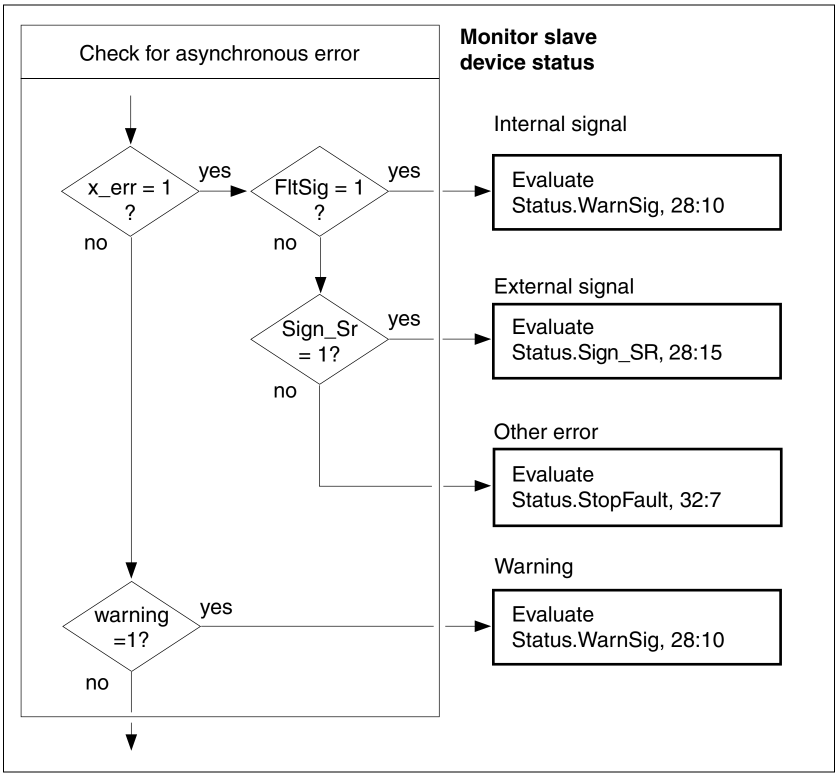

Check for asynchronous error
Monitor slave device status
x_err = 1 ?
yes
no
FltSig = 1 ?
yes
no
Sign_Sr = 1?
yes
no
warning =1?
yes
no
Internal signal
Evaluate Status.WarnSig, 28:10
External signal
Evaluate Status.Sign_SR, 28:15
Other error
Evaluate Status.StopFault, 32:7
Warning
Evaluate Status.WarnSig, 28:10

Fig. 7.3 Evaluation of asynchronous errors

For information about parameters, error classes and troubleshooting, refer to the controller manual, chapter 7, "Diagnosis and troubleshooting".

8 Service

8.1 Service address

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

9 Glossaries

9.1 Abbreviations

Abbreviation	Meaning
AC	Alternating current
ASCII	American Standard Code for Information Interchange Standard for coding text characters
DC	Direct current
DP	Decentralized Periphery
E	Incremental encoder
EC	European Community
EMC	Electromagnetic compatibility
EU	European Union
HMI	Human-Machine Interface, plug-in hand-held operating unit
I/O	Inputs/Outputs
Inc	Increment
IT system	I: isolated; T: terre (fr.), ground. System without potential to ground, not grounded
LED	Light Emitting Diode
M	Motor
PC	Personal Computer
PELV	Protective Extra-Low Voltage
PLC	Programmable Logic Controller
RC	Residual current
SM	Stepper motor
ZK	DC link

9.2 Glossary

Blocking detection

Blocking detection monitors whether the optional index pulse is always correctly triggered at the same angular position of the motor axis.

CAN

Standardized open field bus over which the drives and other devices from different manufacturers communicate with one another.

Default values

Preset values for the parameters of the compact drive before initial commissioning, factory settings.

DIP switch

Small switches adjacent to one another in the plug housing. Must be set during installation.

Direction of rotation

Rotation of the motor shaft in a clockwise or anticlockwise direction. A clockwise direction of rotation is defined as the motor shaft rotating clockwise as the observer faces the end of the protruding shaft.

Encoder

Sensor for recording the angular position of a rotating element. The encoder is mounted on the motor and signals the angular position of the rotor.

Limit switch

Switches that signal an overrun of the permissible travel range.

Power amplifier

This is the unit that controls the motor. The power amplifier generates currents for controlling the motor in accordance with the positioning signals from the control unit.

Error class

Response of the compact drive to an operational malfunction corresponding to one of five error classes

HEX switch

Small rotary switch with 16 positions in the plug housing. Must be set during installation.

HIGH/OPEN

Signal status of an input or output signal. The signal voltage is high, high level, in rest status.

Index pulse

Encoder signal for referencing the rotor position in the motor. The encoder sends one index pulse per revolution.

IT system

Power system with no ground potential reference, since it is not grounded.
I: isolation: T: terre (fr.), ground

Power controller

See Power amplifier

LOW/OPEN

Signal status of an input or output signal. The signal voltage is low, low level, in rest status.

Motor phase current

In a stepper motor the available torque is specified by the motor phase current. The higher the motor phase current, the higher the torque.

Node guarding

Connection monitoring at the field bus RS484 or CAN interface

Parameter

Device data and values that can be set by the user.

Quick Stop

This function is used in the event of faults, the STOP command or for fast braking of the motor in an emergency.

RS485

Field bus module which enables the field bus to be used over a multipoint connection with serial data transmission. A multipoint connection - in contrast to a point-to-point connection - can swap data with several devices on the bus.

Watchdog	Device which detects faults in the compact drive. In the event of failure the compact drive switches power amplifier and outputs off.
Angular position of the motor	The angular position of the motor corresponds to the angular position of the rotor in the motor housing, and refers to the zero point. Also referred to as the index position of the position sensor.

Index

Supplement