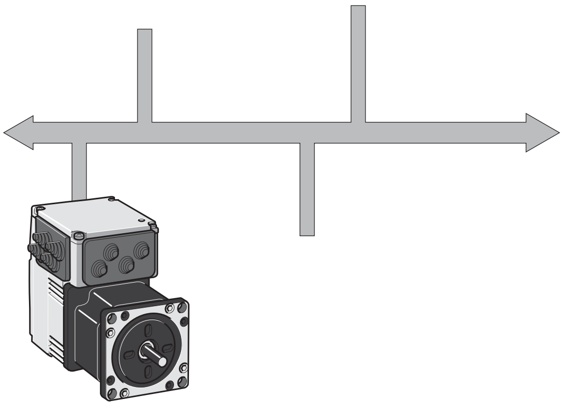
IL•1R RS485

Fieldbus interface
Fieldbus manual

V2.01, 11.2008



0198441113592, V2.01, 11.2008

BERGER LAHR

www.schneider-electric.com

Schneider Electric

Important information

This manual is part of the product.

Carefully read this manual and observe all instructions.

Keep this manual for future reference.

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

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

Some products are not available in all countries.
For information on the availability of products, please consult the catalog.

Subject to technical modifications without notice.

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

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

0198441113592, V2.01, 11.2008

Table of Contents

Important information . . . 2

Table of Contents . . . 3

Writing conventions and symbols . . . 5

1 Introduction . . . 7

1.1 About this manual . . . 7

1.2 RS485 transmission technology . . . 7

1.3 Directives and standards . . . 7

1.4 Documentation and literature references . . . 7

2 Before you begin - safety information . . . 9

3 Basics . . . 11

3.1 Network topology . . . 11

3.2 Access procedures . . . 11

3.3 Fieldbus devices on the RS485 network . . . 12

3.4 Data communication with the slave . . . 12
3.4.1 Addressing . . . 12
3.4.2 Communication via parameters . . . 13
3.4.3 Online command processing . . . 13
3.4.4 Commands . . . 14
3.4.5 Data structure . . . 14
3.4.6 Addressing command . . . 15
3.4.7 Coding and decoding of data . . . 15
3.4.8 Transmit data frame . . . 17
3.4.9 Receive data frame . . . 18
3.4.10 Short transmit command . . . 21

3.5 Monitoring and acknowledgement mechanism . . . 21
3.5.1 Connection monitoring . . . 21
3.5.2 Function of bits sf and rf . . . 22
3.5.3 Command error bit . . . 23
3.5.4 Example of a positioning command . . . 23

3.6 Control and action commands . . . 24

4 Installation . . . 29

5 Commissioning . . . 31

5.1 Commissioning the network . . . 31
5.1.1 Starting network operation . . . 31

0198441113592, V2.01, 11.2008

6 Operation. 33

6.1 Operating modes and functions 33

6.2 Examples of fieldbus operation. 34
6.2.1 Structure of the examples 34
6.2.2 Operating state 36
6.2.3 Operating mode independent parameters 39
6.2.4 Operating mode-dependent processing parameters 40
6.2.5 Reading device information 40
6.2.6 Using different operating modes 42
6.2.7 Examples of error processing 47

7 Diagnostics and troubleshooting 55

7.1 Troubleshooting 55

7.2 Error messages 55

7.3 Synchronous errors 56

7.4 Asynchronous error 57

8 Glossary 59

8.1 Units and conversion tables 59
8.1.1 Length 59
8.1.2 Mass 59
8.1.3 Force 59
8.1.4 Power 59
8.1.5 Rotation 60
8.1.6 Torque 60
8.1.7 Moment of inertia 60
8.1.8 Temperature 60
8.1.9 Conductor cross section 60

8.2 Terms and Abbreviations 61

9 Index 63

0198441113592, V2.01, 11.2008

Writing conventions and symbols

Work steps

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

■ Special prerequisites for the following work steps

▶ Step 1

◁ Specific response to this work step

▶ Step 2

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

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

Bulleted lists

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

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

Making work easier

Information on making work easier is highlighted by this symbol:

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

SI units

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

Example:
Minimum conductor cross section: 1.5 mm^2 (AWG 14)

0198441113592, V2.01, 11.2008

0198441113592, V2.01, 11.2008

1 Introduction

1.1 About this manual

This manual describes the fieldbus specifics for products in a fieldbus network addressed via RS485.

1.2 RS485 transmission technology

In the case of serial transmission with RS485 technology, one or more products and a master computer form a fieldbus system. The products are interconnected in parallel via an RS485 interface and execute the command that are sent from the master computer via the RS485 interface.

RS485 technology excels with a simple structure of the network and the possibility to use conventional computers with serial interfaces (e.g. PC). Even simple terminal programs can be used to send commands from the computer to the product.

1.3 Directives and standards

The following directives and standards apply to the fieldbus handling of products which are addressed via Modbus:

- RS485 standard,
- EIA RS485.2-4 serial interface

1.4 Documentation and literature references

The following manuals belong to this product:

- **Product manual**, describes the technical data, installation, commissioning and all operating modes and functions.
- **Fieldbus manual**, description required to integrate the product into a fieldbus.

Supported devices The product manual of the specific device describes whether the commissioning software can be operated with the device.

0198441113592, V2.01, 11.2008

0198441113592, V2.01, 11.2008

2 Before you begin - safety information

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

0198441113592, V2.01, 11.2008

0198441113592, V2.01, 11.2008

3 Basics

3.1 Network topology

The RS485 network consists of a computer as a master and the product as a slave.

Master Masters are active fieldbus devices which control the data communication within the network.

Examples of masters:

- Automation devices, e.g. PLCs
- PC

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

Examples of slave devices:

- Integrated drives
- Drive controllers

3.2 Access procedures

Master-Slave method Data is exchanged with the product with the master-slave method. The master sends a command to the slave and waits for the slave to acknowledge the command. The slave only sends after is has received a command from the master.

Prior to transmission, the master establishes a point-to-point connection with the slave using an addressing command. The other slaves on the network cannot be addressed as long as the connection is active

0198441113592, V2.01, 11.2008

3.3 Fieldbus devices on the RS485 network

The commands for products with the BL protocol are different from the commands for other devices.

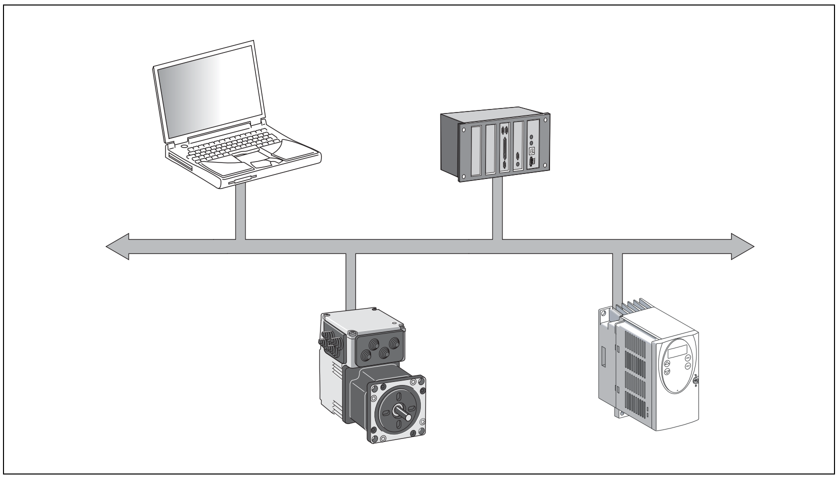


Figure 3.1 Fieldbus devices in the network

3.4 Data communication with the slave

3.4.1 Addressing

In order for the master to be able to communicate with a slave via the fieldbus, it must address the slave. When the addressed slave responds, the communication channel remains open until the connection is terminated or another device is addressed.

0198441113592, V2.01, 11.2008

3.4.2 Communication via parameters

The parameters of the slave serve as the basis of communication between the master and the slave.

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

Example The parameter `PTP.p_absPTP` is used to start absolute positioning in Profile Position mode:

- Parameter name: `PTP.p_absPTP`
- Index:Sub-index: 35:01
- Target position: 4650 increments

The parameter is activated by means of an RS485 command:

	Subindex	Index	Target position
…	01	35	4650

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

Example In the case of the parameter `Status.UDC_act, 31:20`, the factor for converting read values into voltage is: **1 V = 10**. A read value of 345 corresponds to a voltage of 34.5 V.

3.4.3 Online command processing

The master sends a command to the slave, or example:

- Trigger motion command
- Start operating mode
- Request information from slave

The slave executes the command and acknowledges its successful execution with a message.

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

Transmit data The master device places a command in the transmit data memory. The command is sent from this memory to the slave where it is executed.

Receive data The slave acknowledges the execution status of the command in the received data. If the command was not correctly executed, the acknowledgement contains an error message.

The master device can send a new commands as soon as it has received acknowledgement of the current command.

0198441113592, V2.01, 11.2008

3.4.4 Commands

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

Control commands After a control command the master expects a feedback message from the slave indicating whether the command could be successfully executed and processing be completed.

Action commands After an action command the slave simply reports whether it was possible to successfully start processing. The master must then continuously monitor for completion of the process command by requesting receive data from the slave and evaluating this data.

3.4.5 Data structure

Information contained in the transmit and receive data

- Action or control commands
- Administration data for operability of the network

The administration data is provided by the application program of the master device.

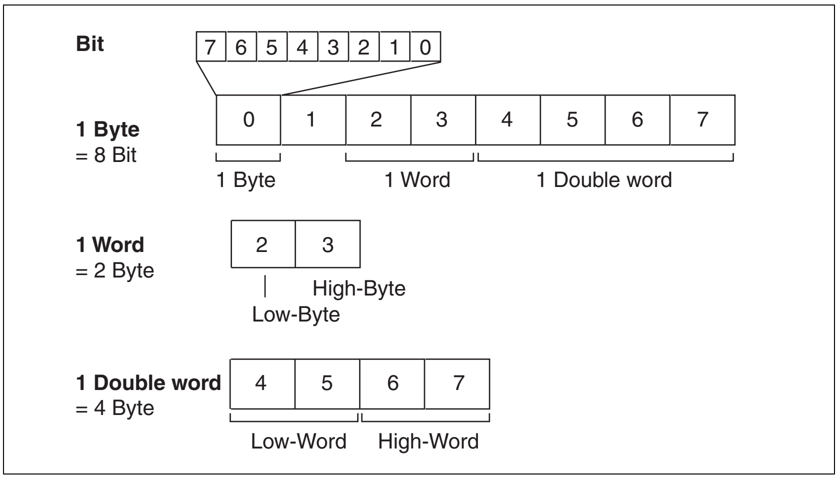


Figure 3.2 Data structure of the RS485 data bus

Byte 1 contains the acknowledgement information for coordination the exchange of data.

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

Hexadecimal values are indicated by an "$_h$" following the number (example: 31_h).

0198441113592, V2.01, 11.2008

3.4.6 Addressing command

The master device must address the slave by means of an addressing command before communication can take place via the RS485 bus.

Structure `<Adresse> <CR>`

`<CR>` (CR = carriage return) is a character that is suffixed to the transmitted command to identify the end of the command.

Example of addressing `<#21> <CR>`

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

Hexadecimal notation: 23_h, 32_h, 31_h, $0D_h$

The slave with the corresponding address returns the addressing command to the master as a confirmation.

Communication between the master and the addressed slave remains active until one of the following events occurs:

- Termination of the communication
- Addressing of another slave

The addressing command does not influence the controller status; however, it clears the device buffer of the communication interface.

Up to 30 devices can be connected and addressed via one RS485 bus.

3.4.7 Coding and decoding of data

In the RS485 network, master and slave transmit data in ASCII format.

Since the size of the initial data frame is 8 bytes, the initial data frame must be converted into an ASCII data frame of 16 bytes prior to transmission y the master.

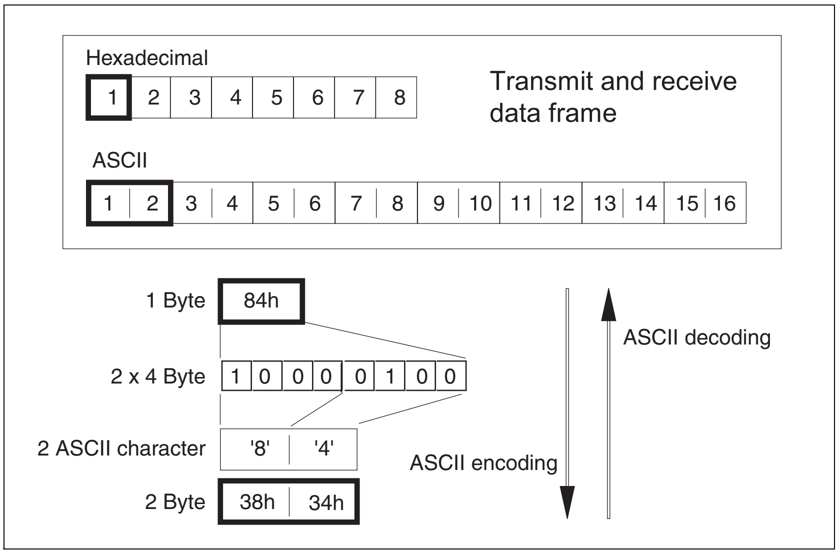


Figure 3.3 Principle of ASCII coding, the first byte of the 8 byte data frame

Prior to transmission, an `<CR>` ($0D_h$) is appended to the 16 byte ASCII data frame to identify its end.

0198441113592, V2.01, 11.2008

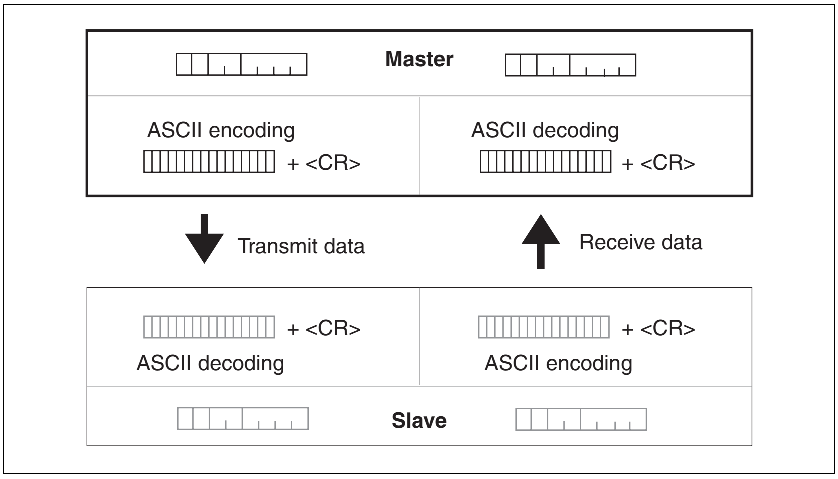


Figure 3.4 ASCII coding and decoding of the data frame

Example of ASCII coding

The command to activate the power stage is to be sent to the slave.

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.

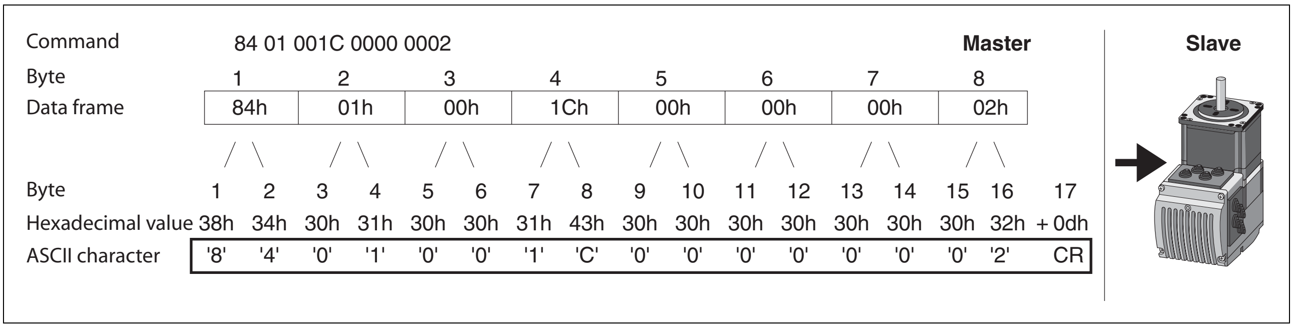


Figure 3.5 Coding of the transmit data

The slave acknowledges the command by returning the receive 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.

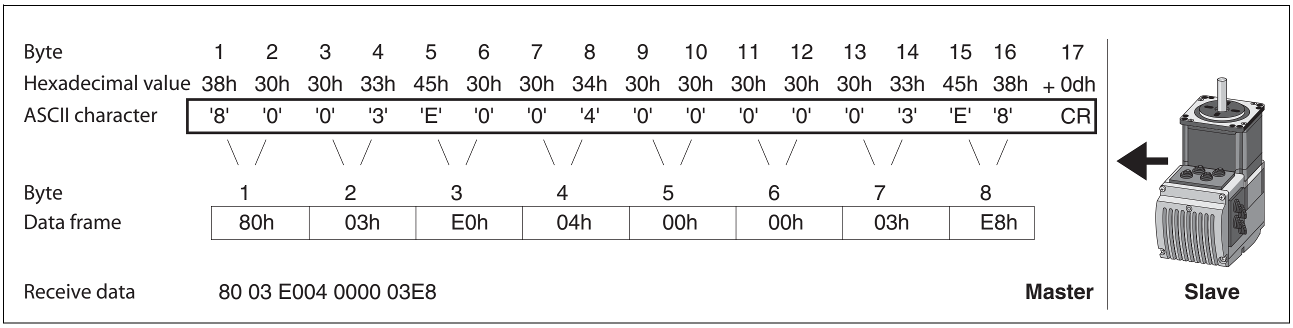


Figure 3.6 Decoding the receive data

0198441113592, V2.01, 11.2008

Coding tables

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

ASCII control character	Hexadecimal code	Meaning
#	23_h	Beginning of addressing command
<CR>	$0D_h$	End of command

3.4.8 Transmit data frame

The master sends an action or control command with the transmit data.

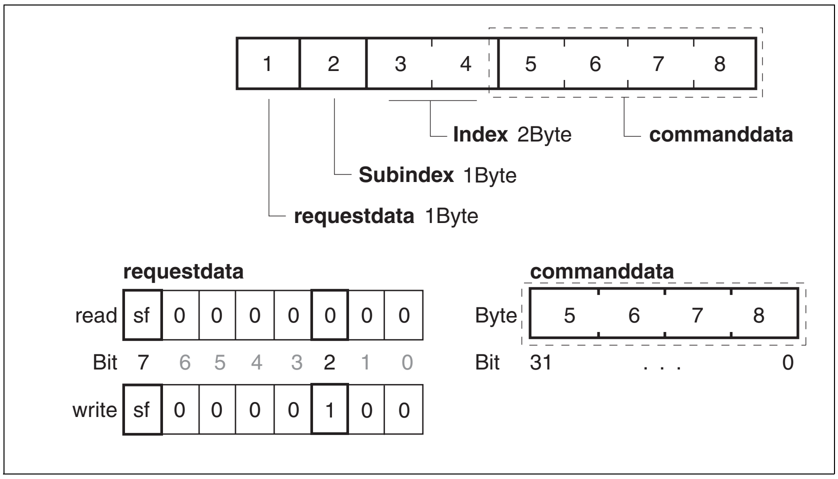


Figure 3.7 Transmit data frame

Byte 1: `requestdata`

Byte 1 contains the control information for the following processes:

- Acknowledgement of command
- Synchronization
- Command as read or write access

Bit	Name	Meaning
2	–	0 = read value: The slave reads a parameter value and sends the parameter value to the master. 1 = write value: The master sends a parameter value to the slave which is written to the controller.
7	`sf`	`sf` = sendflag: The master indicates a new command for the slave with a signal change of the `sf` bit. The `rf` bit is the corresponding bit in the receive data.

Bytes 2 ... 4: `Subindex, Index` Within the command, the parameter is addressed using its index and sub-index.

For further information on parameters and parameter values, see

Bytes 5 ... 8: `commanddata` Bytes 5 … 8 contain the parameter value (e.g. the reference speed) sent to the slave with the command.

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

3.4.9 Receive data frame

Contents of the receive data:

- Response of slave to a command from the master
- Operating status of the slave

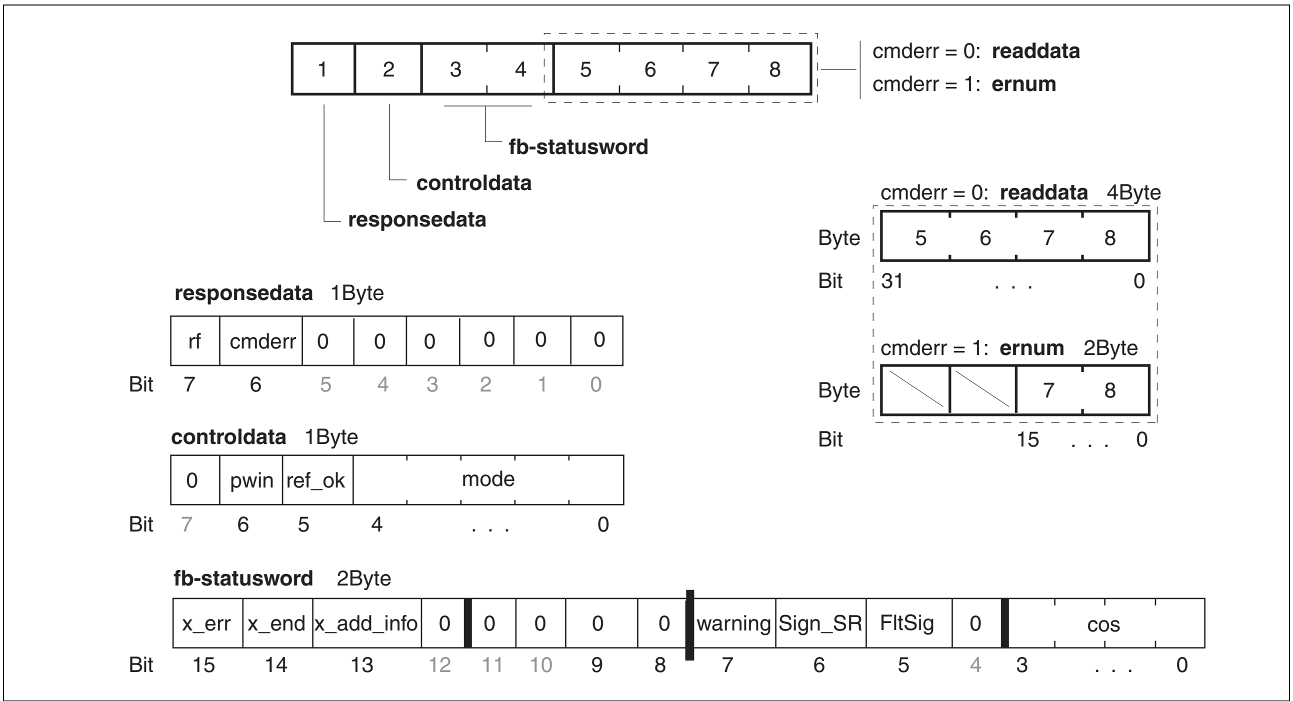


Figure 3.8 Receive data frame

0198441113592, V2.01, 11.2008

Byte 1: `responsedata`

Byte 1 contains the following information:

- Error detection
- Acknowledgment
- Synchronization

Bit	Name	Meaning
6	`cmderr`	(cmderr = command error) Signal is only valid after correct acknowledgment of a data packet 0 = command was executed without error. 1 = command error, bytes 7 and 8 contain the error number `errnum`.
7	`rf`	`rf` = receiveflag acknowledgement from the integrated drive by switching the `rf` bit. After that `sf=rf`. As long as rf #Variable:ungleich#sf: New command not yet processed.

Byte 2: `controldata`

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

Bit	Name	Meaning
0 ... 4	`mode`	Current operating mode, bit-coded example: 3 = 00011 - operating mode Profile Position 2: Homing Operating mode Profile Position 4: Profile Velocity
5	`ref_ok`	0: No reference point defined.1: Homing successful

0198441113592, V2.01, 11.2008

Bytes 3, 4: fb-statusword

The fb-statusword is used to monitor the status of the integrated drive. This information can also be read via the LOW word of parameter Status.driveStat, 28:2.

Bit	Name	Meaning
0 ... 3	cos	Operating state of the integrated drive. Details on the indication and detection of the operating states can can be found in the chapter "Operation" of the product manual.
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	0: No warning message 1: Warning message, cause via parameter Status.WarnSig,28:10
13	x_add_info	Operating mode-specific status bit for monitoring the processing status, see page 24.
14	x_end	Operating mode-specific status bit for monitoring the processing status, see page 24.
15	X_err	Operating mode-specific status bit for monitoring the processing status, see page 24.

Bytes 5 ... 8: readdata

Read data (readdata) represents control information (e.g. the current motor position) which is requested from the master by means of a command. The most recently requested control information continues to be sent with the next receive data until different control information is read.

If no read data have yet been requested, the slave sends the current axis position.

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

See the product manual for information on the data types used.

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

Bytes 7, 8: errnum

If the command was not executed correctly, the slave reports an error via Byte 1, Bit 6 cmderr. The cause of the error is contained in bytes 7 and 8, errnum.

See the product manual for a list of the error numbers.

0198441113592, V2.01, 11.2008

3.4.10 Short transmit command

The contents of the value readdata can also be requested via the short control command <CR> ($0D_h$).

The slave sends the current status information to the master as an acknowledgement. Bits `sf` and `rf` for the acknowledgement mechanism do not change.

3.5 Monitoring and acknowledgement mechanism

3.5.1 Connection monitoring

Two monitoring mechanisms are used on an ongoing basis to avoid data communication errors:

- Timeout monitoring
- Check of received characters

Timeout monitoring

If the master sends a command or the control character <CR> to the slave, the master expects an acknowledgement from the slave within 200 ms. If this acknowledgement is not received, the command could not be executed by slave. The slave is then no longer addressed.

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

Check of received characters

The slave checks an incoming command for transmission errors. If the slave was unable to receive the command correctly, it does not send an acknowledgement.

The master checks the receive data for transmission errors and error messages. If a transmission error has occurred, the master resends the data to the slave.

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

- Bit `sf` in the transmit data
- Bit `rf` in the receive data

An error message is recognized via bit `cmderr` in the receive data.

3.5.2 Function of bits sf and rf

New command from master

The master denotes a command a new by switching bit `sf`.

Evaluation of a command by the slave

The slave compares the two bits:

- `sf`#Variable:ungleich#`rf`: command is new.
- `sf`#variable:gleich#`rf`: command already processed.

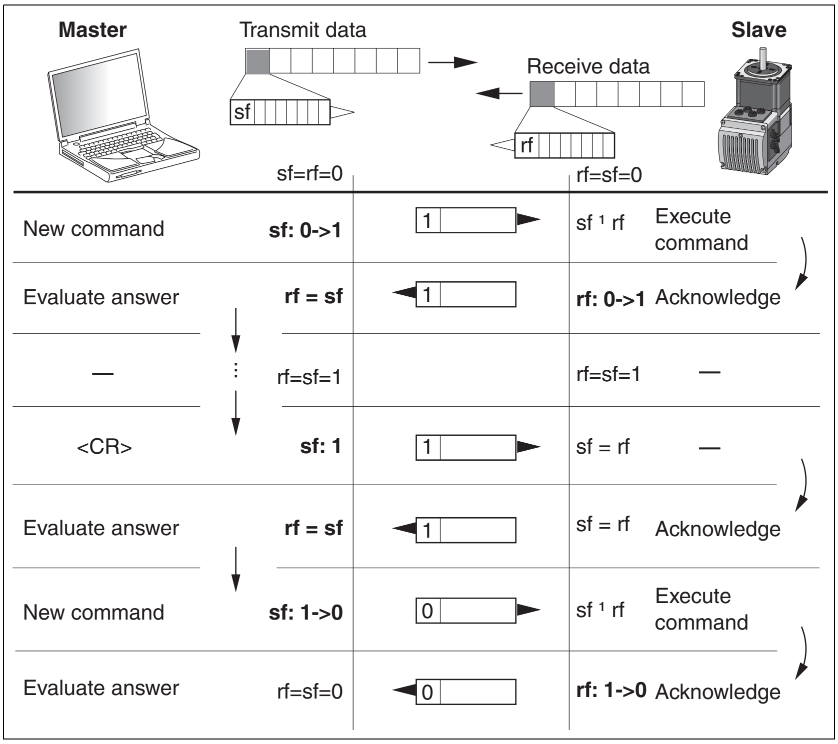


Figure 3.9 Data communication and synchronization of bits `sf` and `rf`

Response from slave

If the command has been executed, the slave switches bit `rf` and sends bit `rf` to the master with the response data.

If the slave receives a command that has already been executed, it sends an acknowledgement with the current status information to the master.

Evaluation by master

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

`sf`#Variable:gleich#`rf` confirm the execution of the command.

3.5.3 Command error bit

The command error bit is valid if the command from the master was acknowledged.

- `cmderr`#Variable:gleich#0: Command successfully executed.
- `cmderr`#Variable:gleich#1: A synchronous error occurred.

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

See the product manual for a list of the error numbers.

See 7.3 "Synchronous errors" for additional information on synchronous errors.

3.5.4 Example of a positioning command

A relative positioning command is sent to the slave.

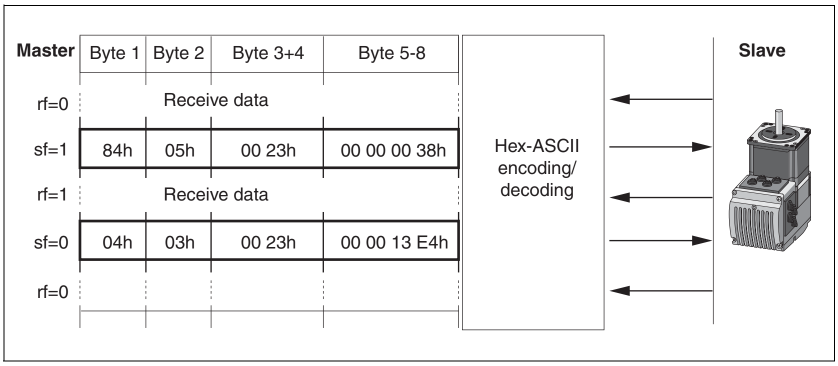


Figure 3.10 Example: Relative positioning command

The first command sets the reference speed.

- Byte 1: `requestdata`

  New command #Variable:daraus_folgt#switch bit `sf`

  Write access with bit 2 #Variable:gleich#0: 4_h

- Bytes 2 ... 4: `Subindex` and `Index`

  Parameter call:

  05_h:00 23_h: Parameter `PTP.v_tarPTP, 35:5`, reference speed

- Bytes 5 ... 8: `commanddata`

  Value reference speed

  00 00 00 38h = 56 1/min

The second command starts relative positioning.

- Byte 1: `requestdata`

  New command #Variable:daraus_folgt#switch bit `sf`

  Write access with bit 2 #Variable:gleich#0: 4_h

- Bytes 2 ... 4: `Subindex` and `Index`

  Parameter call:

  03_h:00 23_h: Parameter `PTP.p_relPTP, 35:3`, start relative positioning

- Bytes 5 ... 8: `commanddata`

  Value positioning distance

  00 00 13 $E4_h$ = 5092 increments

See 6.2 "Examples of fieldbus operation" for additional examples.

3.6 Control and action commands

The master can send two types of commands:

- Control commands
- Action commands

The slave responds differently, depending on the type of command.

Control commands

Control commands are executed immediately; they are completed when the receive data is transmitted to the master.

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

- Switching outputs
- Changing parameters

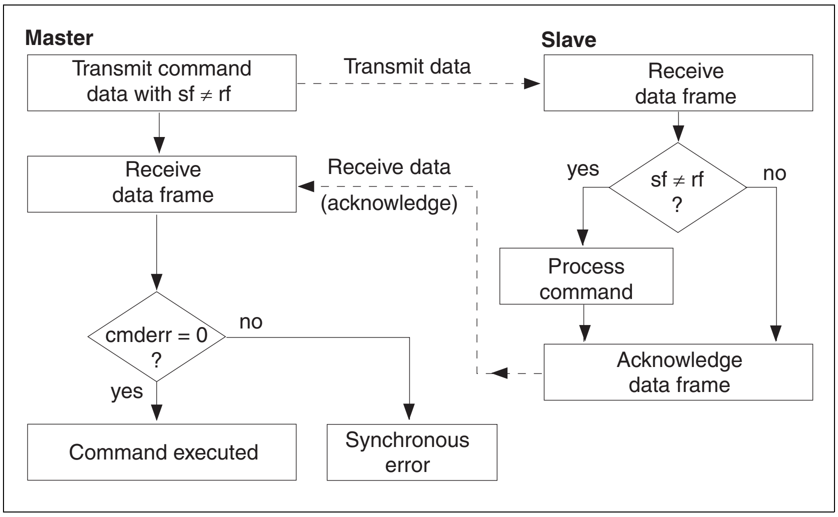


Figure 3.11 Executing control commands

If it was impossible to correctly execute a control command, the slave sets the error bit `cmderr` to "1" and signals a synchronous error.

0198441113592, V2.01, 11.2008

Action commands

Action commands start a motion command. The slave activates the corresponding operating mode and loads the required parameters. The start of the the motion command is returned to the master as a confirmation of the command.

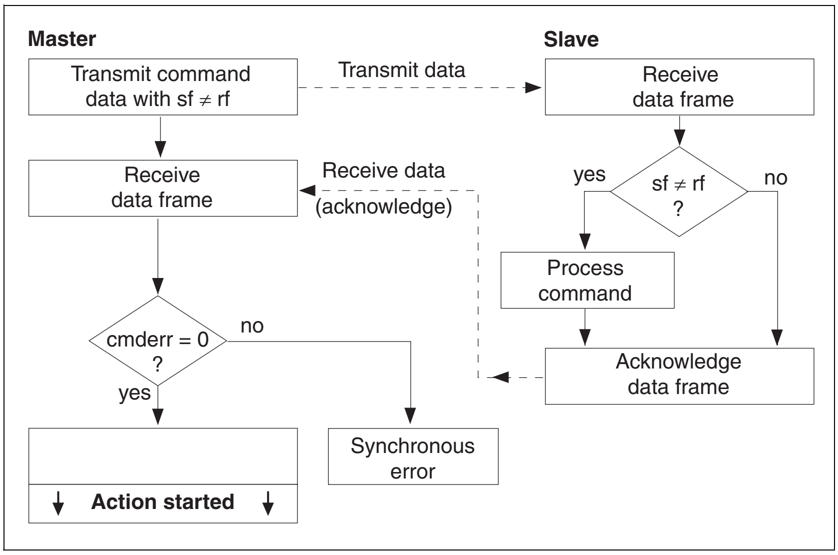


Figure 3.12 Executing an action command

If it was impossible to correctly execute an action command, the slave sets the command error bit `cmderr` to "1" and signals a synchronous error.

Monitoring operating status

The master must monitor the status of a motion command and its completion on an ongoing basis via the status word `fb-statusword` in the receive data.

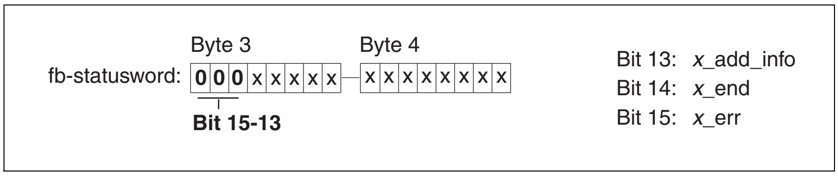


Figure 3.13 Monitoring the operating state of the integrated drive

0198441113592, V2.01, 11.2008

Status bits

The slave internally manages the operating status for each operating mode by means of separate status bits. The slave only signals the status of the current operating mode via the RS485 bus.

Meaning of the status bits:

- Bit 13, `x_add_info`

  Operating mode-specific message

- Bit 14, `x_end`

  Processing status of the operating mode

  0: Processing active

  1: Processing completed, motor stopped

- Bit 15, `x_err`

  Error status during processing

  0: No error

  1: Error

Using the information on the current operating mode (parameter `Status.xMode_act, 28:3`), it is possible to evaluate the status message in an operating mode-specific way.

Operating mode	mode	x_add_info	x_end	x_err
Homing	2	0	ref_end	ref_err
Operating mode Profile Position	3	Reference position reached	motion_end	motion_err
Profile Velocity	4	Reference speed reached	vel_end	vel_err

0198441113592, V2.01, 11.2008

As soon as processing is started via an action command, bit 14 x_end changes "0". When processing is complete, bit 14 switches back to "1" to signal that further steps can now be processed.

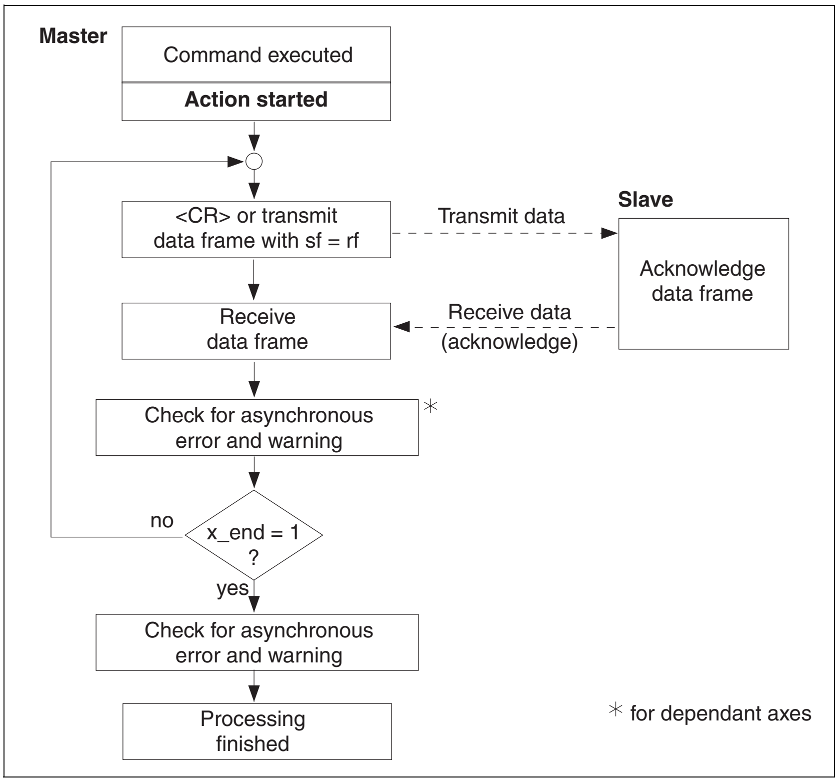


Figure 3.14 Monitoring the execution of the action command

If bit 15 x_err changes to "1", there is an error. The error must be remedied before further processing is possible.

See chapter 7 "Diagnostics and troubleshooting" for additional information on troubleshooting.

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

Influencing the execution process

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

0198441113592, V2.01, 11.2008

4 Installation

⚠ WARNING

LOSS OF CONTROL

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

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

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

⚠ WARNING

SIGNAL AND DEVICE INTERFERENCE

Signal interference can cause unexpected responses of device.

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

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

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

Integrated drives with DIP switches

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

See the chapter "Installation" in the product manual for information on the DIP switch assignments.

0198441113592, V2.01, 11.2008

0198441113592, V2.01, 11.2008

5 Commissioning

⚠ DANGER

UNINTENDED CONSEQUENCES OF EQUIPMENT OPERATION

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

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

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

⚠ WARNING

UNINTENDED OPERATION

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

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

5.1 Commissioning the network

5.1.1 Starting network operation

Network communication with the slave is started via a master. This can be a PLC or a PC with application software that allows you to enter commands and read receive data.

▶ Switch on the slave.

▶ Address the slave. Use a PC with a terminal program for the first test.

▶ Enter the following command: `#01 <CR>`

"01" represents the address set for the integrated drive.

◁ If the slave responds with "#01", the slave is ready for data communication.

0198441113592, V2.01, 11.2008

0198441113592, V2.01, 11.2008

6 Operation

> **⚠ WARNING**
>
> **UNINTENDED OPERATION**
>
> - Do not write values to reserved parameters.
> - Do not write values to parameters unless you fully understand the function. For more information see the product manual.
> - Run initial tests without coupled loads.
> - Verify that the system is free and ready for the movement before changing parameters.
> - Verify the use of the bits with fieldbus communication: bit 0 is far right (least significant). Bit 15 is far left (most significant).
> - Verify the use of the word sequence with fieldbus communication.
> - Do not establish a fieldbus connection unless you have fully understood all communications principles.
>
> **Failure to follow these instructions can result in death, serious injury or equipment damage.**

6.1 Operating modes and functions

This manual only describes the protocol. See the chapters "Operation" and "Parameters" for descriptions of the operating modes, functions and all parameters of the integrated drive:

Operating modes

- Profile Velocity
- Profile Position
- Homing

Functions

- Definition of direction of rotation
- Motion profile
- Quick Stop
- Fast position capture

Settings

The following settings can be made via the fieldbus:

- Reading and writing parameters
- Monitoring the inputs and outputs of the 24 V signal interface
- Activating diagnostics and fault monitoring functions

0198441113592, V2.01, 11.2008

6.2 Examples of fieldbus operation

6.2.1 Structure of the examples

The programming examples show hands-on applications for network operation of integrated drives and complement the function descriptions in the device manual.

The basic function principles of the operating modes and functions are described in the device manual.

The programming examples are organized in the following way:

- Description of the task
- Prerequisites
- Required commands in the transmit data frame
- Response of the slave in the receive data frame
- Possible restrictions concerning the execution of the command and other remarks

Transmit data and receive data are shown in hexadecimal notation.

Commands are shown from the perspective of the master. The response of the slave is only shown if this is needed for further processing. If no response is shown, a positive acknowledgement is assumed.

Transmit data

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

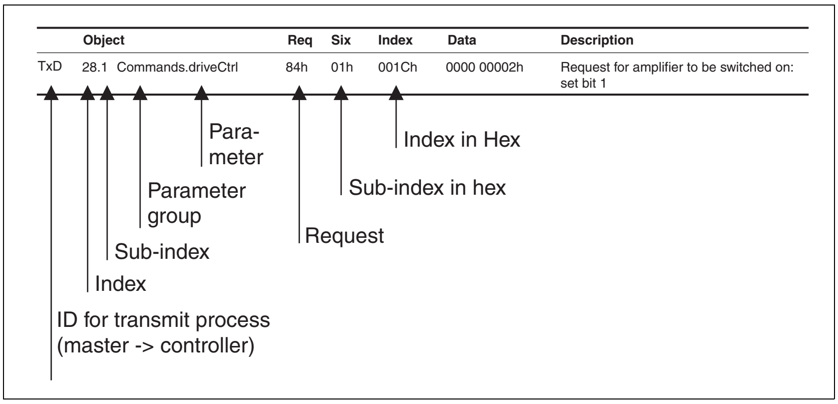


Figure 6.1 Coding of the transmit data

See the parameter description of the integrated drive for the data type of the parameter values. When a value of the data types `INT16` or `UINT16` is transmitted, this value is stored in bytes 7 and 8 . Bytes 5 and 6 must contain "0."

0198441113592, V2.01, 11.2008

Receive data

	Res	Ctrl	fb-status-word	Data	Description
RxD	80h	xxh	xxx4h	xxxx xxxxh	State transition still pending; cos = 4, "Ready to switch on"

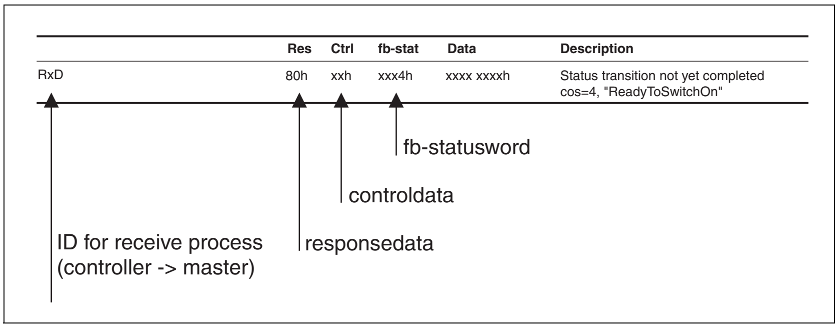


Figure 6.2 Coding of the receive data

See the parameter description of the slave for the data type of the parameter values. When a value of the data types `INT16` or `UINT16` is transmitted, this value is stored in bytes 7 and 8 . Bytes 5 and 6 contain "0".

An error is indicated in `responsedata` by `cmderr = 1`. The error number of a synchronous error message is stored as data type `UINT16`.

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

Units of the values

If read or write values are indicated in user-defined units [usr], scaling factors must be applied to such user-defined units.

Example

In the case of the parameter `Status.UDC_act, 31:20`, the factor for converting read values into voltage is: **1 V = 10**. A read value of 345 corresponds to a voltage of 34.5 V.

Acknowledgement bits

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

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

Therefore, the first command must be transmitted with `sf` = 1 . This also applies if the description of a topic is split across several examples. A new command is recognized as a result of a bit change.

Data frame

The programming examples only show the 8 byte data frame for the slaves. The fieldbus-specific control bytes must be added by the application on the master in accordance with fieldbus protocol.

6.2.2 Operating state

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

Detailed information on operating states and a state diagram can be found in the device manual.

6.2.2.1 Checking the operating status

The operating state is evaluated via the RS485 bus with the first 4 bits in the status word `fb-statusword`. The status word is transmitted with every record of receive data.

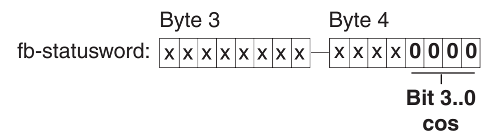


Figure 6.3 Operating state of the slave

Bit 3..0	Operating state	Meaning
-	-	24 V switched on
...0001	1 - Start	Initialization of the device electronics
...0010	2 - Not ready to switch on	Power stage is not ready to switch on
...0011	3 - Switch on disabled	Switching on the power stage is disabled.
...0100	4 - Ready to switch on	The power stage is ready to switch on.
...0110	6 - Operation enable	The device is working in the selected mode.
...0111	7 - Quick Stop active	Quick Stop is being executed.
...1000... 1001	8 - Fault reaction active9 - Fault	Fault reaction activatedFault indication

The operating states 0..3, 8 and 9 are transition states; the positioning mode does not remain in these states if it operates correctly.

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

See the device manual for information on troubleshooting.

6.2.2.2 Changing the operating state

The operating state of the slave is sent with the bits `cos` in `fb-statusword`. Coding corresponds to the operating states of the slave.

Whether or not you can change an operating state depends on the current operating state.

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

- The slave has started up without errors.
- There are no faults.

Use parameter `Commands.driveCrtl, 28:1` to change the operating state. The value is always "0" so that the write access to a bit automatically triggers an edge change 0 #Variable:Pfeil_nach_rechts#1.

Bit 3..0	Control word	Meaning
0 0 0 1	Disable	Switching off the power stage
0 0 1 0	Enable	Switching on the power stage
0 1 0 0	Quick Stop	Trigger Halt by Quick Stop
1 0 0 0	Fault Reset	Acknowledge error message

Switching on the power stage

Prerequisites:

- The slave is in the operating state "Ready to switch on".
- The parameter settings for the slave are correct.

Transmit and receive data

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

Receive data

	Res	Ctrl	fb-status-word	Data	Description
RxD	80h	xxh	xxx4h	xxxx xxxxh	State transition still pending; cos = 4, "Ready to switch on"

Once the state transition has been successfully completed, the slave signals the following:

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

Remarks As long as the operating state "Operation Enable" has not been reached, bit 15 (`x_err`) is activated in the status word `fb-statusword`.

Operating state	fb-statusword `x_err` (bit 15)
#Variable:ungleich# Operation Enable	1
= Operation Enable	0

Switching off the power stage

Prerequisites:

- The slave is in the operating state "Operation Enable".

Transmit and receive data

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

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

Once the state transition has been successfully completed, the slave signals the following:

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

Remarks Once the operating state "Operation Enable" is left, bit 15 (`x_err`) is activated in the status word `fb-statusword`.

Operating state	fb-statusword `x_err` (bit 15)
#Variable:ungleich# Operation Enable	1
= Operation Enable	0

Movement interruption by software stop

Prerequisites

- The slave is in the operating state "Operation Enable".
- All settings required for the operating mode must have been made.

0198441113592, V2.01, 11.2008

Transmit and receive data

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

Message after successful activation of the operating state "Quick Stop Active":

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

Message after slave has come to a standstill:

	Res	Ctrl	fb-status-word	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"

Remarks When the slave has come to a stop, you can clear the interruption by means of a "Fault Reset".

See chapter 6.2.7.4 "Resetting a fault ("Fault Reset")" for additional information on "Fault Reset".

6.2.3 Operating mode independent parameters

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

The following parameter groups contain operating mode-independent processing parameters:

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

Enabling signal for monitoring parameters

Task Activate limit switch monitoring.

Prerequisites
- The slave is in the operating state "Ready to switch on".

Transmit and receive data

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

0198441113592, V2.01, 11.2008

6.2.4 Operating mode-dependent processing parameters

The following parameter groups contain operating mode-dependent processing parameters:

- Homing `Homing`
- Profile position operating mode `PTP`
- Profile Velocity `VEL`

Setting the reference speed for Profile Position operating mode

Task Set a reference speed of 500 1/min for Profile Position operating mode.

Prerequisites
- The slave is in the operating state "Operation Enable".

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	35:5 PTP.v_tarPTP	84h	05h	0023h	0000 01F4h	Sets the fast reference speed to 500 1/min = 1F4h

Remarks You can change the reference speed prior to and during positioning.

6.2.5 Reading device information

The following actions are performed after a read access:

- The current parameter value is output.
- The parameter value requested with the read access continues to be output with each additional read or write access until another parameter value is requested.

See the chapter "Parameters" in the device manual for information on whether or not a parameter value can be read and on the data type of the parameter value. All values marked "R" in the "R/W" column of the parameter list can be read. The data type is indicated in the "Value range" column.

0198441113592, V2.01, 11.2008

6.2.5.1 Reading parameter settings

A read access allows you to read the current value of a parameter.

Reading the reference speed in Profile Position operating mode

Prerequisites

- The parameter value is available in the current operating state of the slave.

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	35:5 PTP.v_tarPTP	80h	05h	0023_h	$0000\ 0000_h$	Reads the reference speed in Profile Position operating mode

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80h	xxh	$xxxx_h$	0000 00B4h	Reference speed in bytes 5 to 8: $B4_h$ = 180 1/min

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

6.2.5.2 Reading status information

The `Status` parameter group provides various read value relating to the processing status of the slave.

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

Reading the actual motor speed

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

Prerequisites

- The parameter value is available in the current operating state of the slave.

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	31:6 Status.n_act	80h	06_h	$001F_h$	$0000\ 0000_h$	Reads the actual speed of the motor

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80h	xx_h	$xxxx_h$	0000 03E8h	Actual speed in bytes 7 to 8: 3E8h = 1000 1/min

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

0198441113592, V2.01, 11.2008

6.2.6 Using different operating modes

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

6.2.6.1 Determining the operating mode and the processing status

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

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

x_err	x_end	Processing status
0	0	Processing active and no error
0	1	Processing terminated and no error
1	0	Processing active and error detected
1	1	Processing terminated and error detected

If an asynchronous error occurs during processing, `x_err` is set `= 1` and the slave is stopped. After the integrated drive has come to a standstill, `x_end = 1` is set.

6.2.6.2 Operating mode Profile Position

A pallet is to be moved to 2 stations on an endless conveyor belt at a speed of 200 1/min. All units are increments [inc].

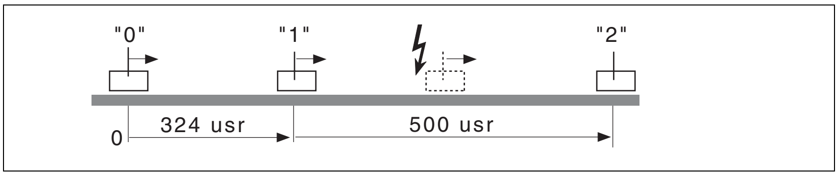


Figure 6.4 Moving the pallet

The first station is 324 inc away from position "0". Absolute positioning is used to move the pallet to the first station. The second station is another 500 inc away and reached by relative positioning.

During the movement to the second station, an interruption is triggered by the signal STOP. The interrupted movement is to be continued after the error is remedied.

Prerequisites

- All settings must have been made via the functions.
- The homing position "0" must have been defined. See chapter 6.2.6.4 "Homing" for additional information.
- The integrated drive is at a standstill.

Performing absolute positioning

Task Move the pallet to position 324 Inc at a speed of 200 1/min.

Prerequisites

- The slave is in the operating state "Operation Enable".

0198441113592, V2.01, 11.2008

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	35:5 PTP.v_tarPTP	84h	05_h	0023_h	0000 $00C8_h$	Sets the reference speed to 200 1/min = 00C8h 1/min

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

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		00h	$x3_h$	6006_h	xxxx $xxxx_h$	Motor moving

The processing status is contained in bit `x_end` in `fb-statusword`. Upon termination of processing, `x_end = 1`.

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		00h	$2x_h$	6006h	xxxx xxxxh	Motor movement terminated

Remarks The reference speed can be changed at any time during the movement by means of a a write access to `PTP.v_tarPTP, 35:5`.

Performing relative positioning

Task Continue the movement from example 9 and move the pallet by +500 increments.

Prerequisites
- The slave is in the operating state "Operation Enable".

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	35.3 PTP.p_relPTP	84_h	03_h	0023_h	0000 $01F4_h$	Starts relative positioning by +500 Inc = 01F4 Inc

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		00_h	$x3_h$	0006_h	xxxx $xxxx_h$	Motor moving

The processing status is contained in bit `x_end` in `fb-statusword`. Upon termination of processing, `x_end = 1`.

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		00_h	$2x_h$	6006_h	xxxx $xxxx_h$	Motor movement terminated

0198441113592, V2.01, 11.2008

Remarks The reference speed can be changed at any time during the movement by means of a write access to PTP.v_tarPTP, 35:5.

Continuing interrupted movement

Task The movement from example 9 is interrupted by the signal STOP. Continue the movement.

Prerequisites
- Positioning has been interrupted by the signal STOP.
- The cause of the error has been remedied, i.e. the signal STOP is no longer active.
- By means of a "Fault Reset", the slave has been set to the operating state "Operation Enable".

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	35:4 PTP.continue	84_h	04_h	0023_h	$0000\ 0000_h$	Triggers continuation of the interrupted positioning

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00_h	$x3_h$	0006_h	$xxxx\ xxxx_h$	Motor moving

The processing status is contained in bit x_end in fb-statusword. Upon termination of processing, x_end = 1.

	Object	Res	Ctrl	fb-statusword	Data	Description
RxD		00_h	23_h	6006_h	$xxxx\ xxxx_h$	Motor movement terminated

Remarks The reference speed can be changed at any time during the movement by means of a a write access to PTP.v_tarPTP, 35:5.

0198441113592, V2.01, 11.2008

6.2.6.3 Profile Velocity

Setting the reference speed

Task Set the reference speed to 2000 1/min and check for reaching of the reference speed.

Prerequisites
- All settings must have been made via the functions.
- The slave is in the operating state "Operation Enable".
- The integrated drive is at a standstill.

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	36:1 VEL.velocity	84_h	01_h	0024_h	$0000\ 07D0_h$	Sets the reference speed: +2000 1/min = $07D0_h$ 1/min

You can check whether the reference speed has been reached via `x_add_info` in `fb-statusword`. If the reference speed is reached, `x_add_info = 1`.

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80_h	$x4_h$	2006_h	xxxx xxxxh	Motor movement active, reference speed reached

Stopping the integrated drive

Prerequisites
- The integrated drive runs at a reference speed of > 0.

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	36:1 VEL.velocity	04_h	01_h	0024_h	0000 0000h	Sets the reference speed to 0 1/min = 0h 1/min

The processing status is contained in bit `x_end` in `fb-statusword`. Upon termination of processing, `x_end = 1`.

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		00_h	$x4_h$	$6xx6_h$	xxxx $xxxx_h$	Motor movement terminated

0198441113592, V2.01, 11.2008

6.2.6.4 Homing

The following conditions must be satisfied before you can run the "Homing" operating mode examples.

- All settings must have been made via the functions.
- The slave is in the operating state "Operation Enable".
- The integrated drive is at a standstill.

Position setting

Task Set the current position of the slave to a value of 1000 increments. The new value serves as the reference point for further movements.

Prerequisites
- The integrated drive is at a standstill.

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	40:3 Homing.startSetp	84_h	03_h	0028_h	0000 03E8h	Position setting position +1000 inc = $03E8_h$ inc

Position setting is run immediately when the parameter is called. You can check for successful completion via bits `x_end` and `x_err` in `fb-statusword`.

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80_h	$x2_h$	4006_h	xxxx $xxxx_h$	Motor movement active, reference speed reached

After successful homing, the following bit settings apply in `controldata` ref_ok = 1.

Reference movement

Task Run a reference movement in negative direction to reference switch REF. The search movement to reference switch REF is to be performed at a reference speed of 500 1/min.

Prerequisites
- The integrated drive is at a standstill.

0198441113592, V2.01, 11.2008

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	40:4 Homing.v_Home	84_h	04_h	0028_h	$0000\ 01F4_h$	Reference speed for search of reference switch REF: 500 1/min = 01F4h 1/min

	Object	Req	Six	Index	Data	Description
TxD	40:1 Homing.startHome	04_h	01_h	0028_h	$0000\ 0003_h$	Starts the reference movement to reference switch REF in negative direction

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80_h	$x0_h$	0006_h	$xxxx\ xxxx_h$	Reference movement active

Position setting is run immediately when the parameter is called. You can check for successful completion via bits `x_end` and `x_err` in `fb-statusword`.

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80_h	$x2_h$	4006_h	$xxxx\ xxxx_h$	Reference movement complete.

After successful homing, the following bit settings apply in `controldata` ref_ok = 1.

6.2.7 Examples of error processing

6.2.7.1 Synchronous errors

Synchronous errors only occur in response to a command. When a command is sent, an immediate check is performed as to whether the command can be correctly executed. If not, the slave returns an error number in response to the command and sets the bit `cmderr = 1` in `responsedata`. This does not change the operating state of the slave.

Generating a synchronous error

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

Prerequisites
- The slave is in the operating state "Operation Enable".

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	0:255 non-existent parameter	84_h	FF_h	0000_h	$xxxx\ xxxx_h$	Access to a non-existent parameter

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		$C0_h$	xx_h	6006_h	$0000\ 0130_h$	`cdmerr` =The error number (`errnum`) is stored in bytes 7 and 8.

0198441113592, V2.01, 11.2008

6.2.7.2 Asynchronous errors

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

Internal monitoring signals

Task Generate and internal error and determine the cause of the error

Prerequisites
- The slave is in the operating state "Operation Enable".

Transmit and receive data
▶ Reduce the power supply of the slave to less than 15 V.

The slave signals an error.

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		00_h	xx_h	$E029_h$	$xxxx\ xxxx_h$	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	80_h	12_h	$001C_h$	$xxxx\ xxxx_h$	

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80_h	xx_h	$E029_h$	$0000\ 0001_h$	Internal monitoring signal:Bit 0 active: undervoltage power supply

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

Remarks When the slave has come to a stop, you can clear the interruption by means of a "Fault Reset".

See chapter 6.2.7.4 "Resetting a fault ("Fault Reset")" for additional information on "Fault Reset".

External monitoring signals

Task A movement was interrupted by a light barrier at the input `STOP`. Determine the cause of the error.

Prerequisites
- The external monitoring signal `STOP` is enabled (parameter `Settings.SignEnabl, 28:13`).
- The slave is in the operating state "Quick Stop Active".

Transmit and receive data

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		00_h	xx_h	$E047_h$	xxxx $xxxx_h$	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	80_h	$0F_h$	$001C_h$	xxxx $xxxx_h$	

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80_h	xx_h	$E047_h$	0000 0004_h	External monitoring signal:Bit 2 active: cause of error STOP

Remarks When the slave has come to a stop, you can clear the interruption by means of a "Fault Reset".

See chapter 6.2.7.4 "Resetting a fault ("Fault Reset")" for additional information on "Fault Reset".

6.2.7.3 Other errors

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

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

The error number of the last stop fault can be determined via parameter `Status.StopFault, 32:7`.

Errors that cause the integrated drive to exit the State "Operation Enable" are entered as described below:

- In the bit string of the external and/or internal monitoring signals
- Cause of last stop
- In the error memory

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

Determining an error number

Task Determine the last cause of stop from the error memory of the integrated drive.

Prerequisites
- Due to the activation of the input `STOP` the integrated drive is in the operating state "Quick Stop Active".

Transmit and receive data Now determine the cause of the error.

	Object	Req	Six	Index	Data	Description
TxD	28:15 Status.StopFault	80_h	07_h	0020_h	$0000\ 0000_h$	Requests the cause of the last stop fault

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80_h	xx_h	$Cx47_h$	$xxxx\ 015A_h$	In fb-statusword:x_err = 1, x_end = 1, SignSr = 1, error number $015A_h$ in bytes 7 and 8

Resetting the error or switching the 24 V power supply off and on again clears the cause of the last stop. If more than one error was detected, only the error that resulted in the slave being stopped, i.e. that caused the drive to exit the "OperationEnable" state, is saved as the cause of the stop. Any subsequently occurring faults are saved in the standard error memory in order of occurrence.

6.2.7.4 Resetting a fault ("Fault Reset")

Once you have remedied the cause of a fault, you can exit the error states "Quick Stop" and "Fault" using a "Fault Reset".

You can only perform a "Fault Reset" if the integrated drive is at a standstill (`x_end = 1`).

After a successful "Fault Reset", the error number of the last cause of the stop fault is cleared from the parameter `Status.StopFault, 28:15`.

A "Fault Reset" does not clear the entry in the standard error memory.

Resetting faults

Task Reset the stop fault caused by caused by the switch `STOP` in the previous example.

Prerequisites

- Due to the activation of the input `STOP` the integrated drive is in the operating state "Quick Stop Active".
- The cause of the fault no longer persists, i.e. the input `STOP` is deactivated.
- The integrated drive is at a standstill (`x_end = 1`)

0198441113592, V2.01, 11.2008

Transmit and receive data

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		00_h	xx_h	$E047_h$	xxxx $xxxx_h$	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	84_h	01_h	$001C_h$	0000 0008_h	Request: Bit 3 "Fault Reset"

After the error has been successfully rest, the drive exits the Fault state. The integrated drive switches to the operating state "Operation Enable" ein.

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80_h	xx_h	$6xx6_h$	xxxx $xxxx_h$	In fb-statusword:x_err = 0,cos = 6 "Operation Enable"

6.2.7.5 Reading and deleting the contents of the error memory

All error messages are written to the error memory of the slave 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 integrated drive is switched off.

The following information is stored in the error memory:

- Error number
- Error class
- Age of the error (= number of times the integrated drive has been switched on since the error occurred)
- Error frequency (= number of times in succession an error has occurred)
- Additional information

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

Reading the error memory

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

Prerequisites

- The oldest fault entry is a stop fault caused by the switch `STOP`.
- The slave is in the operating state "Quick Stop Active".

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	900:1 ErrMem0.ErrNum	80_h	01_h	0384_h	$0000\ 0000_h$	Request: Error number in ErrMem0

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80_h	xx_h	$Cx47_h$	$0000\ 015A_h$	In fb-statusword:x_err = 1, x_end = 1, SignSr = 1, error number 015Ah: "Stop/Quick Stop through STOP"

	Object	Req	Six	Index	Data	Description
TxD	900:2 ErrMem0.Class	00_h	02_h	0384_h	$0000\ 0000_h$	Request: Error class

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		00_h	xx_h	$Cx47_h$	$0000\ 0001_h$	Error class 1 (in bytes 7 and 8)

	Object	Req	Six	Index	Data	Description
TxD	900:3 ErrMem0.Age	80_h	03_h	0384_h	$0000\ 0000_h$	Request: Age of the error

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80_h	xx_h	$Cx47_h$	$0000\ 000_h$	Age of the error: Value 0=slave has not been switched off since the error occurred.

	Object	Req	Six	Index	Data	Description
TxD	900:4 ErrMem0.Repeat	00_h	04_h	0384_h	$0000\ 0000_h$	Request: Error frequency

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		00_h	xx_h	$Cx47_h$	$0000\ 0005_h$	Value 5=Error has occurred 6 times in succession

	Object	Req	Six	Index	Data	Description
TxD	900:5 ErrMem0.Age	80_h	05_h	0384_h	$0000\ 0000_h$	Request: Additional information for evaluation of the error

	Object	Res	Ctrl	fb-status-word	Data	Description
RxD		80_h	xx_h	$Cx47_h$	$0000\ 0000_h$	Value 0: no additional information available

0198441113592, V2.01, 11.2008

Remarks Accessing the other entries in the error memory (`ErrMem1..6`) works in the same way.

For access, change the index of the parameter (for example: For `ErrMem5`#Variable:Implikation# Index = 905 = 389h)

Delete error memory

Task Delete all entries from the error memory.

Transmit and receive data

	Object	Req	Six	Index	Data	Description
TxD	32:2 Commands.del_err	00_h	02_h	0020_h	$0000\ 0001_h$	Request: Delete contents of the error memory

Remarks Selective deletion of individual entries in the error memory is impossible. The error memory is deleted even if the error is still active.

0198441113592, V2.01, 11.2008

7 Diagnostics and troubleshooting

7.1 Troubleshooting

Check the following settings if the slave does not respond:

- Did you start the slave and switch on the master?
- Are all cable connections ok (electrically and mechanically)?
- Did you set the correct address at the slave? Check the DIP switch settings.
- Did you set the same baud rate and the same interface parameters set for the master and compact drive?

If the slave still does not respond:

▶ Open the cover of the connector housing.

▶ Compare the behavior of LED with the information in the table below.

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

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

7.2 Error messages

During network operation, the master receives error messages along with the receive data.

2 types of errors are distinguished:

- Synchronous errors
- Asynchronous errors

Synchronous errors

The master receives messages regarding a synchronous error directly from the drive if the transmitted command could not be processed. A synchronous error is detected via the `cmderr` bit.

Asynchronous errors

Asynchronous errors are signaled by the monitoring units in the slave as soon as a device fault occurs. The master monitors the status word `fb_statusword` on an ongoing basis to detect asynchronous errors.

The slave supplies the status information cyclically (bus cycle).

7.3 Synchronous errors

A synchronous error is evaluated on the basis of the error bit `cmderr` in the first byte of the receive data:

- `cmderr = 0`: Command successfully executed

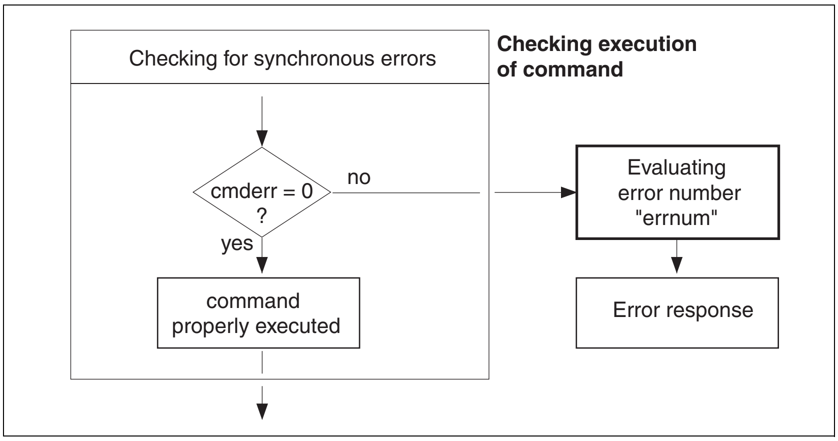


Figure 7.1 Evaluation, synchronous error

Causes of errors

Possible causes of a synchronous error are:

- Unknown command, syntax error or incorrect transmit data frame
- Parameter value outside the permissible value range
- Invalid action command or control command during processing
- Error during execution of an action command or control command

The bit `cmderr` is not valid unless the command was acknowledged. Along with the receive data, the slave returns an error number `errnum` in bytes 7 and 8 of the receive data; this number can be used to determine the cause of the error.

A table of error numbers is provided in the "Diagnostics and Troubleshooting" chapter of the product manual.

0198441113592, V2.01, 11.2008

7.4 Asynchronous error

For detection of asynchronous errors, the error bits in the status word `fb_statusword` must be monitored.

- Bit 15, `x_err` Fault state during processing
  determine cause via bit 5 and bit 6
- Bit 7, `warning` warning message of slave
  for example, temperature warning power stage
- Bit 6, `Sign_SR` message from external monitoring signal
  for example, movement interrupted by `STOP`
- Bit 5, `FltSig` message from internal monitoring signal
  for example, power stage overtemperature.

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

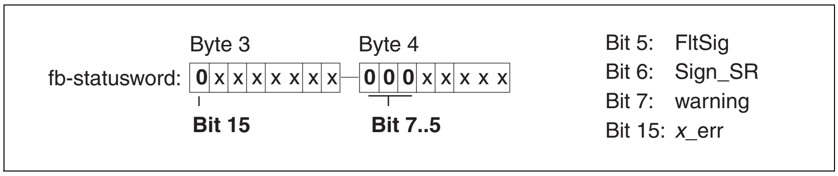


Figure 7.2 Status word for evaluation of asynchronous errors

Warning message

In the case of a warning message, the motion command continues to be processed; the error information is written to parameter `Status.WarnSig, 28:10`.

0198441113592, V2.01, 11.2008

Error message If the slave sets the x_err signal, it immediately stops the motion and responds either by braking or by immediately shutting off the power stage, depending on the error class. Together with bit x_err, bits FltSig or Sign_Sr are set. The meaning of the error message must be determined via the corresponding parameter.

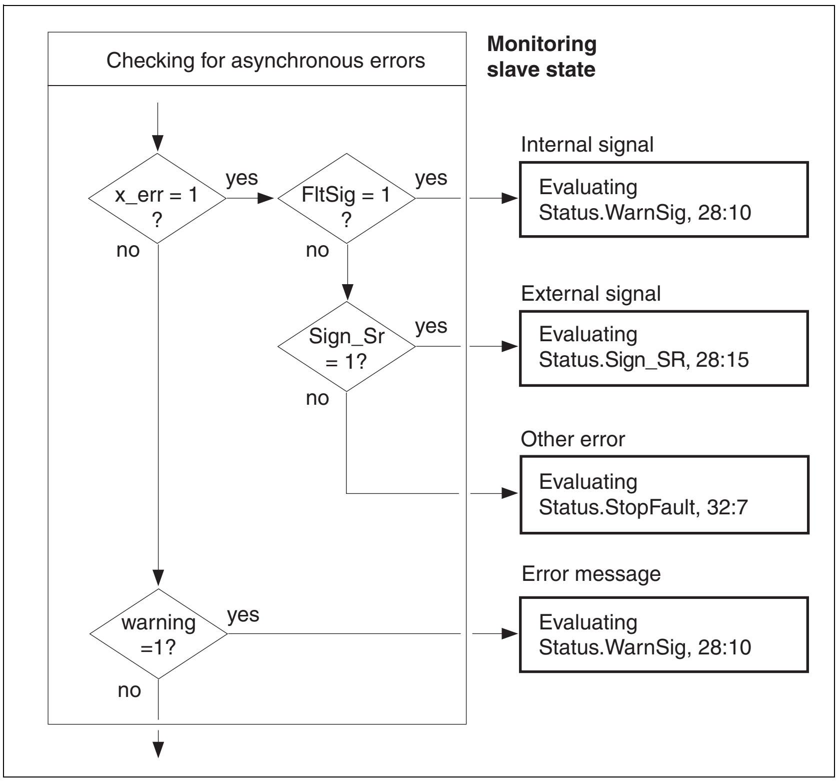


Figure 7.3 Evaluation of asynchronous errors

Information on parameters, error classes and troubleshooting can be found in the product manual.

0198441113592, V2.01, 11.2008

8 Glossary

8.1 Units and conversion tables

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

Example: conversion of 5 meters [m] to yards [yd]
5 m / 0.9144 = 5.468 yd

8.1.1 Length

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

8.1.2 Mass

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

8.1.3 Force

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

8.1.4 Power

	HP	W
HP	-	* 746
W	/ 746	-

0198441113592, V2.01, 11.2008

8.1.5 Rotation

	min^{-1} (RPM)	rad/s	deg./s
min^{-1} (RPM)	-	* π / 30	* 6
rad/s	* 30 / π	-	* 57.295
deg./s	/ 6	/ 57.295	-

8.1.6 Torque

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

8.1.7 Moment of inertia

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

8.1.8 Temperature

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

8.1.9 Conductor cross section

AWG	1	2	3	4	5	6	7	8	9	10	11	12	13
mm^2	42.4	33.6	26.7	21.2	16.8	13.3	10.5	8.4	6.6	5.3	4.2	3.3	2.6

AWG	14	15	16	17	18	19	20	21	22	23	24	25	26
mm^2	2.1	1.7	1.3	1.0	0.82	0.65	0.52	0.41	0.33	0.26	0.20	0.16	0.13

0198441113592, V2.01, 11.2008

8.2 Terms and Abbreviations

ASCII American Standard Code for Information Interchange. Standard for coding of characters

Blocking detection Blocking detection monitors the max. current, the time and the rotation of the motor shaft. If, in spite of maximum current, the motor shaft does not move for a parameterized time, the monitoring system signals a blocking error.

Default value Factory setting.

Direction of rotation Rotation of the motor shaft in a clockwise or counterclockwise direction of rotation. Clockwise rotation is when the motor shaft rotates clockwise as you look at the end of the protruding motor shaft.

EMC Electromagnetic compatibility

Encoder Sensor for detection of the angular position of a rotating component. Installed in a motor, the encoder shows the angular position of the rotor.

Error Discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition.

Error class Classification of errors into groups. The different error classes allow for specific responses to faults, for example by severity.

Fatal error In the case of fatal error, the product is not longer able to control the motor, so that an immediate deactivation of the power stage is necessary.

Fault Operating state of the drive caused as a result of a discrepancy between a detected (computed, measured or signaled) value or condition and the specified or theoretically correct value or condition.

Fault reset A function used to restore the drive to an operational state after a detected error is cleared by removing the cause of the error so that the error is no longer active (transition from operating state "Fault" to state "Operation Enable").

Forcing Forcing switching states of inputs/outputs.

I/O Inputs/outputs

Inc Increments

Limit switch Switches that signal overtravel of the permissible range of travel.

Motor phase current The available torque of a stepper motor is determined by the motor phase current. The higher the motor phase current the higher the torque.

Node guarding Monitoring of the connection to the slave at an interface for cyclic data traffic.

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

Persistent Indicates whether the value of the parameter is persistent, i.e. whether it remains in the memory after the device is switched off . When changing a value via commissioning software or fieldbus, the user must explicitly store the changed value in the persistent memory. When a value is entered via the HMI, the device stores the value of the parameter automatically each time it is changed.

PLC Programmable logic controller

0198441113592, V2.01, 11.2008

Power stage The power stage controls the motor. The power stage generates current for controlling the motor on the basis of the positioning signals from the controller.

PWM Pulse width modulation

Quick Stop Function used to enable fast deceleration of the motor via a command or in the event of an error.

Torque ramp Deceleration of the motor with the maximum possible deceleration, which is only limited by the maximum permissible current. The higher the permissible braking current, the stronger the deceleration. Because energy is recovered up depending on the coupled load, the voltage may increase to excessively high values. In this case the maximum permissible current must be reduced.

Virtual index pulse At every motor revolution, the virtual index pulse is at the same angle position of the motor. The virtual index pulse can be shifted with a parameter.

Warning If the term is used outside the context of safety instructions, a warning alerts to a potential problem that was detected by a monitoring function. A warning is not an error and does not cause a transition of the operating state.

Watchdog Unit that monitors cyclic basic functions in the product. The power stage is disabled and the outputs are switched off in the event of errors.

Zero voltage window Voltage range that is interpreted as 0 V.

0198441113592, V2.01, 11.2008

9 Index

A

Abbreviations 61

B

Before you begin
Safety information 9

C

Commissioning 31

D

Diagnostics 55
Directives and standards 7
Documentation and literature references 7

G

Glossary 59

I

Introduction 7

O

Operation 33

S

Supported devices 7

T

Terms 61
Troubleshooting 55

U

Units and conversion tables 59

0198441113592, V2.01, 11.2008

0198441113592, V2.01, 11.2008