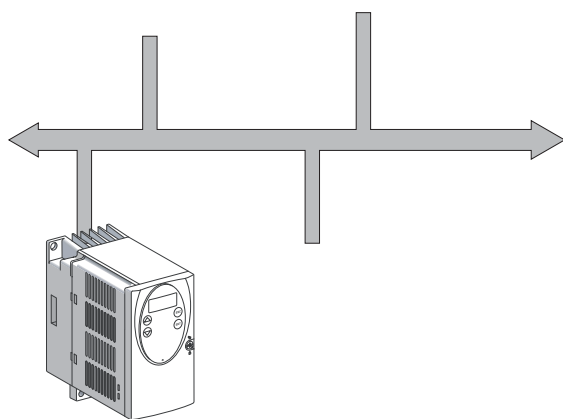


# SD328 Modbus RTU

Fieldbus interface

Fieldbus manual

V2.01, 11.2008



## **Important information**

This manual is part of the product.

Carefully read this manual and observe all instructions.

Keep this manual for future reference.

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

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

Some products are not available in all countries.

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

Subject to technical modifications without notice.

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

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

## Table of Contents

<b>Important information</b>	<b>2</b>
<b>Table of Contents</b>	<b>3</b>
<b>Writing conventions and symbols</b>	<b>5</b>
<b>1 Introduction</b>	<b>7</b>
1.1 Modbus	7
1.2 Directives and standards	7
1.3 Documentation and literature references	7
<b>2 Before you begin - safety information</b>	<b>9</b>
<b>3 Installation</b>	<b>11</b>
<b>4 Commissioning</b>	<b>13</b>
4.1 Fieldbus settings	14
4.2 Starting network operation	16
4.3 Running a function test	16
<b>5 Operation</b>	<b>17</b>
5.1 Basics	17
5.1.1 Modbus network	17
5.1.2 Modbus transmission technology	18
5.1.3 Modbus RTU protocol	19
5.2 Function codes	21
5.2.1 FC 3 (Read Multiple Registers)	21
5.2.2 FC 8 (Diagnostics)	22
5.2.3 FC 16 (Write Multiple Registers)	23
5.2.4 FC 23 (ReadWrite Multiple Registers)	24
5.2.5 FC 43 (Read Device Identification)	25
5.3 Examples of function codes FC	26
5.4 Examples for standardized operating modes	28
5.4.1 Operating mode Profile Position	28
5.4.2 Operating mode Profile velocity	29
5.4.3 Operating mode Homing	30
5.5 Examples of manufacturer-specific operating modes	31
5.5.1 Operating mode Electronic Gear	31
5.5.2 Operating mode Jog	32
5.6 Connection monitoring	33

<b>6</b>	<b>Diagnostics and troubleshooting</b>	<b>35</b>
6.1	Communication errors	35
6.2	Protocol errors	35
6.3	Handling errors	36
6.3.1	Synchronous errors	36
<b>7</b>	<b>Glossary</b>	<b>37</b>
7.1	Units and conversion tables	37
7.1.1	Length	37
7.1.2	Mass	37
7.1.3	Force	37
7.1.4	Power	37
7.1.5	Rotation	38
7.1.6	Torque	38
7.1.7	Moment of inertia	38
7.1.8	Temperature	38
7.1.9	Conductor cross section	38
7.2	Terms and Abbreviations	39
<b>8</b>	<b>Index</b>	<b>41</b>

## 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<sup>2</sup> (AWG 14)



# 1 Introduction

## 1.1 Modbus

This manual describes the fieldbus handling for products in the fieldbus network, which are addressed via Modbus RTU.

In order to be able to use a PC as a master in a Modbus network, the PC must be equipped with an RS485 interface card. In the case of a PC with an RS232 interface, a level shifter must be connected between the network connection and the PC interface.

Fieldbus devices by other manufacturers can be operated in the same RS485 network as long as they support the Modbus protocol.

## 1.2 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

## 1.3 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.
- *Literature* • Modicon Modbus Protocol Reference Guide, PI-MBUS-300 Rev. J, June 1996, MODICON, Inc., USA
- <http://www.modicon.com>



## 2 Before you begin - safety information

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



### 3 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.<sup>1)</sup>
- 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.



## 4 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.**

## 4.1 Fieldbus settings

*Transmission format* The data transmission format is factory-set to:

- Modbus RTU
- 19200 Baud
- 8 data bits (LSB is transmitted first)
- Even parity
- 1 stop bit

Communication between master and slave takes place in half-duplex mode.

*Node address* Up to 31 fieldbus devices can be connected to the bus. Each fieldbus device must have its own node address which must be unique in the network. The node address is factory-set to 1. The node address 0 is a broadcast address; all fieldbus devices in the network receive data from this address, but they do not respond to it.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
MBadr	Modbus address	-	UINT16	CANopen 3016:4 <sub>h</sub>
COM- - MBAD	Valid addresses: 1 to 247	1	UINT16	Modbus 5640
Цагн- - ИбРд		1	R/W	
		247	per.	
			-	

*Baud rate* The baud rate must be set to the same value for all fieldbus devices.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
MBbaud	Modbus Baud rate	-	UINT16	CANopen 3016:3 <sub>h</sub>
COM- - MBBD	<b>9600 / 9.6KB / 9.5:</b> 9600 Baud	9600	UINT16	Modbus 5638
Цагн- - Иббд	<b>19200 / 19.2KB / 19.2:</b> 19200 Baud	19200	R/W	
	<b>38400 / 38.4KB / 38.4:</b> 38400 Baud	38400	per.	
			-	
	NOTE: Changed settings do not become active until the unit is switched on the next time			

*Data bits, stop bits and parity* The following combinations of data bits, stop bits and parity can be set.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
MBformat	Modbus data format	-	UINT16	CANopen 3016:5 <sub>h</sub>
COM- - MBFo	<b>1 / 8Bit NoParity 1Stop / Bn 1:</b> 8 bits, no parity bit, 1 stop bit	1	UINT16	Modbus 5642
Com- - BbFo	<b>2 / 8Bit EvenParity 1Stop / BE 1:</b> 8 bits, even parity bit, 1 stop bit	2	R/W	
	<b>3 / 8Bit OddParity 1Stop / Bo 1:</b> 8 bits, odd parity bit, 1 stop bit	4	per.	
	<b>4 / 8Bit NoParity 2Stop / Bn2:</b> 8 bits, no parity bit, 2 stop bits		-	
	NOTE: Changed settings do not become active until the unit is switched on the next time			

*Node guarding* A monitoring time can be set for node guarding.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
MBnode_guard	Modbus node guard	ms	UINT16	CANopen 3016:6 <sub>h</sub>
-	Node guard	0	UINT16	Modbus 5644
-	0: Inactive (default)	0	R/W	
-	>0: Monitoring time	10000	-	

*Word sequence* This setting specifies the way the parameter data (2 words) is transmitted.

Example: parameter value = 1234 5678<sub>h</sub>

- HighWord-LowWord = 1234<sub>h</sub> , 5678<sub>h</sub>
- LowWord-HighWord = 5678<sub>h</sub> , 1234<sub>h</sub>

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
MBdword_order	Modbus word sequence for double words (32 bit values)	-	UINT16	CANopen 3016:7 <sub>h</sub>
COM- - MBWo	<b>0 / HighLow / h, Lo:</b> HighWord-LowWord	0	UINT16	Modbus 5646
Com- - BbLo	<b>1 / LowHigh / Lo, h:</b> LowWord-HighWord	0	R/W	
	High word first or low word first	1	per.	
	High word first -> Modicon Quantum		-	
	Low word first -> Premium, HMI (Telemecanique)			

*Detailed description* A detailed description of the settings is provided in the "Commissioning" chapter of the product manual.

## 4.2 Starting network operation

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

## 4.3 Running a function test

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

- Did you switch on the power supply and start the master for network mode?
- Did you properly connect the cables (mechanical connection)?
- Did you set the correct address?
- Did you set the same baud rate and the same interface parameters set (data bits, parity, stop bits)?

For more information on the cause of errors and on troubleshooting see chapter 6 "Diagnostics and troubleshooting" or the product manual.

## 5 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.**

### 5.1 Basics

#### 5.1.1 Modbus network

A Modbus network consists of one master and at least one slave.

*Master* Masters are active fieldbus devices which control the data communication within the network. Examples of masters include:

- Automation devices, e.g. PLCs
- PCs

*Slave* Slaves are passive fieldbus devices. They receive control commands and provide data to the master. Examples of slaves include programmable drive controllers, such as, for example, the device described in this manual.

A typical application for Modbus is communication between devices in automated manufacturing.

*Slave address* In order for the master to be able to communicate with a slave in the fieldbus, it must address the slave. The node address of the slave is contained in the data frame. Information on setting the address of the slave can be found on page 14.

*Parameter addresses* Communication between fieldbus master and slave is based on parameters to which a unique parameter address is assigned. There are write and read parameters. A description of all parameters available for this product (=slave) with the corresponding Modbus addresses can be found in the "Parameters" chapter of the product manual.

The Modbus address is indicated in decimal notation. It must be converted to hexadecimal to be entered. In this manual, hexadecimal values are shown with a subscript "h" at the end. The notation with "16#" before the number is also used for the examples with the Premium.

### 5.1.2 Modbus transmission technology

In a Modbus system, data is transmitted via a serial interface (RS485).

Data is interchanged between the fieldbus devices on the basis of the master-slave logic. Only the master can send commands (requests). The master can address each slave individually. Depending on the command, the response of a slave consists of sending the requested data or executing the requested function.

During data transmission, request and response alternate on an ongoing basis.

The master sends commands to the slave. The slave only sends data if requested by the master to do so.

The exchange of data follows a fixed pattern. The process is described from the perspective of the master.

The commands are embedded in the transmitted data frame in the form of function codes.

The request contains a function code which represents a command to be executed by the slave. The information required to execute the command is included in the transmitted data bytes.

The error bytes enable the slave to verify the integrity of the received data.

The response from the slave contains the function code of the request as an "echo". The data bytes of the response depend on the function code used and are provided by the slave. The error bytes enable the master to verify the validity of the received data.

The structure of the transmitted data is specified by the Modbus protocol.

*Modbus protocols* There are 3 versions of the Modbus protocol:

- Modbus-RTU: Master-slave-communication, binary coded
- Modbus-ASCII: Master-slave-communication, ASCII-coded
- Modbus-PLUS: Peer-to-peer communication

The device only supports the Modbus RTU protocol.

### 5.1.3 Modbus RTU protocol

#### 5.1.3.1 Modbus RTU message

A Modbus message is also referred to as a data frame. A message addressed to a slave is referred to as a transmit data frame or request. In response to this request, this device sends a receive data frame.

A Modbus RTU data frame consists of the following fields:

<SlaveAdr> <FC> <Data> <CRC>		
Field name	Meaning	No. of bytes
<SlaveAdr>	Slave address	1
<FC>	Function code	1
<Data>	Data	n (high byte, low byte)
<CRC>	Checksum	2 (low byte, high byte)

Table 5.1 Fields of a Modbus RTU message

The beginning and the end of a data frame are detected via a time condition. A pause of 3.5 characters means that the data frame is completed and that the next character is to be interpreted as a slave address. As a result, a data frame must be sent as a continuous data stream. In the event of an interruption of more than 1.5 characters the data is rejected by the receiving device.

#### 5.1.3.2 Modbus RTU request and response

Request and response have the same structure.

If an error occurs on receipt of the request or if the slave cannot execute the action, the slave sends an error message as the response.

#### 5.1.3.3 Fields of a Modbus RTU data frame

- <SlaveAdr> The device address identifies the target device. It is the same in both the request and the response.
- <FC> The function code determines the Modbus service the slave is to execute. The function code is the same in both the request and the response.
- <Data> Whether a data field is contained in the data frame and what its length is depends on the function code used. The data field of a request contains the control and action commands of the corresponding function code. The data field of a response contains the data requested by the master, depending on the function code. It can also contain an error message.

<CRC> The "Cyclic Redundancy Checksum" (CRC) is created on the basis of the the transmitted fields <SlaveAdr>, <FC> and <Data> for error checking under Modbus RTU. It is a CRC16 with the generator polynome  $A001_n$ , which is calculated using the algorithm shown in the following illustration.

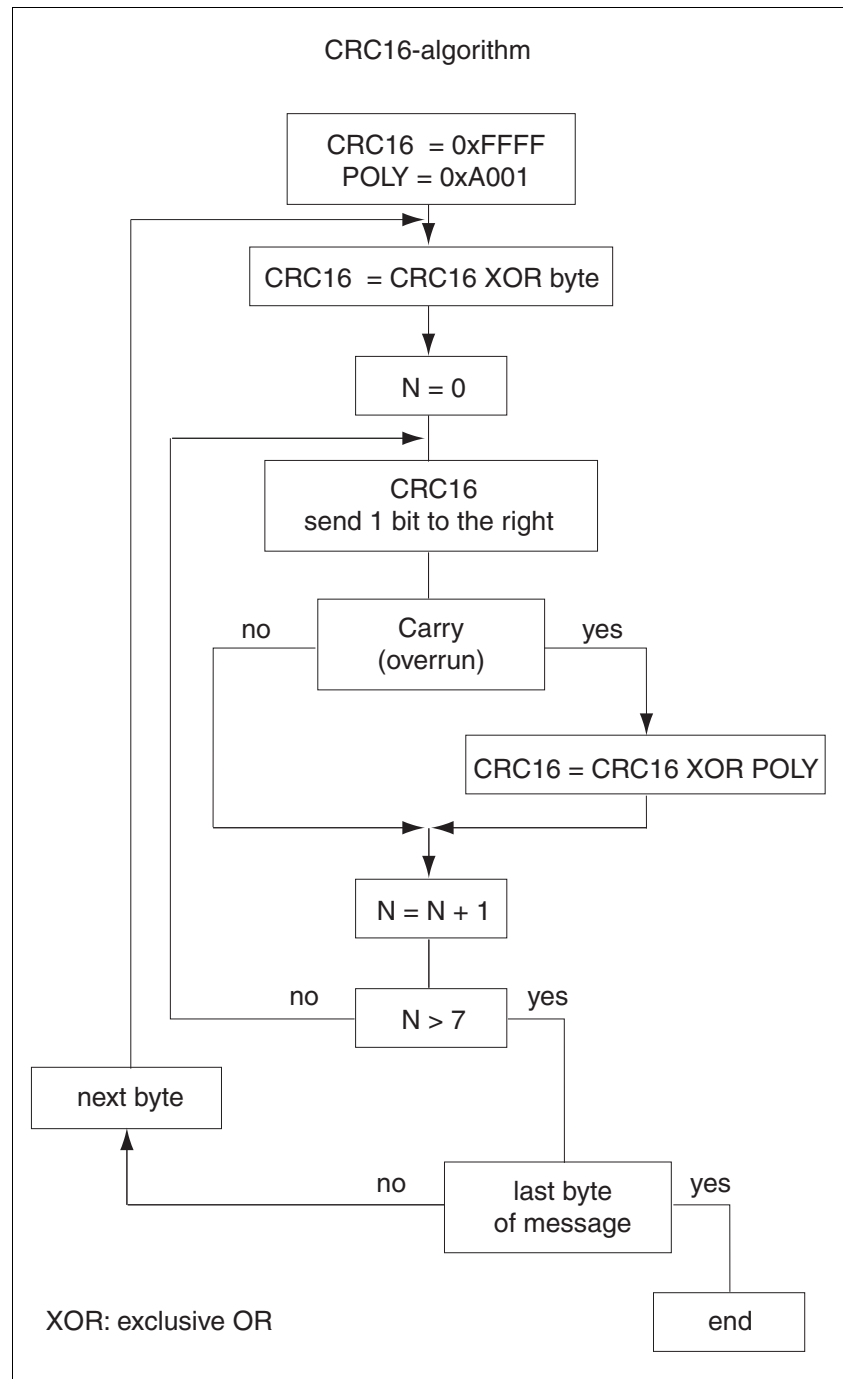


Figure 5.1 CRC16 algorithm

## 5.2 Function codes

The function codes (FC) allow for triggering various communication mechanisms (services) provided by the Modbus protocol. The following table provides an overview of the function codes implemented in the device.

FC	Modbus meaning	Device meaning
3	Read Multiple Registers	Read n Parameters
8	Diagnostics	Diagnostics
16	Write Multiple Registers	Write n Parameters
23	Read/Write Multiple Registers	Read Write n Parameters
43 Subcode14	Read Schneider Identification	–

### 5.2.1 FC 3 (Read Multiple Registers)

With this function code, it is possible to read "n" consecutive parameters, starting at any address.

*Structure of the request:* <FC> <1. ReadAddress> <NumberOfParameters>

Field	Bytes	Value	Meaning
FC	1	3 = 03 <sub>h</sub>	Request code
1. ReadAddress	2	(various)	Address of the first parameter to be read
NumberOfParameters	2	2 * n	Number of the 16 bit parameters to be read

*Structure of the positive response* <FC> <NumberOfBytes> <Data>

Field	Bytes	Value	Meaning
FC	1	3 = 03 <sub>h</sub>	Response code
NumOfBytes	1	4 * n	Number of data bytes
Data	4 * n	(various)	n read parameter values

*Structure of the negative response* see chapter 6.2 "Protocol errors"

*Example of Premium in PL7* ► Read the current motor position:

The Modbus parameter address for the current position (`_p_act`) is 7700 (16#1E14). The following command calls the service "Read Multiple Registers" (FC 3). The current position is stored in %MW20, the result of the execution of the function is stored starting at %MW30.

```
// 2 Worte aus der Parameteradresse 16#1E14 vom Gerät
mit der Adresse (ADR) anfordern
READ_VAR (ADR, '%MW', 16#1E14, 2, %MW20:2, %MW30:4);
```



*All parameters are transmitted as 4 byte values (32Bit). A list of all the parameters can be found in the product manual in the chapter "Parameters".*

## 5.2.2 FC 8 (Diagnostics)

This function code allows diagnostics data of the slave to be read.

*Structure of request:* <FC> <Subfunction> <Data>

Field	Bytes	Value	Meaning
FC	1	8 = 08 <sub>h</sub>	Request code
Subfunction	2	(various)	Subfunction (see Table 5.2)
Data	2	(various)	Data (depending on subfunction)

*Structure of the positive response:* <FC> <Subfunction> <Data>

Field	Bytes	Value	Meaning
FC	1	8 = 08 <sub>h</sub>	Response code
Subfunction	2	(various)	Subfunction (see Table 5.2)
Data	2	(various)	Requested diagnostics data

*Structure of the negative response* see chapter 6.2 "Protocol errors"

*Subfunctions* The following subfunctions are provided by the Modbus protocol:

Code	Subfunction	Device-specific function
00	Return Query Data	Return request as a response
01	Restart Communication Option	Re-initialize the communication port
02	Return Diagnostic Register	Provide the error number in the event of synchronous errors
03	(reserved)	–
04	Force Listen Only Mode	Mute slave
05..09	(reserved)	–
10	Clear Counters and Diagnostic Register	Clear all statistical counters
11	Return Bus Message Count	Indicate the number of received messages
12	Return Bus Communication Error Count	Indicate number of detected LRC errors
13	Return Bus Exception Error Count	Indicate number of detected exception errors
14	(reserved)	–
15	(reserved)	–
16	Return Slave NAK Count	Indicate the number of detected "Not-Acknowledged" errors
17	Return Slave Busy Count	Indicate the number of "Slave Busy" errors
18	Return Bus Char Overrun Count	Indicate number of detected character overrun errors
>18	(reserved)	–

Table 5.2 Modbus subfunctions for FC 8

### 5.2.3 FC 16 (Write Multiple Registers)

With this function code, it is possible to write "m" consecutive parameters, starting at any address.

*Structure of request:* <FC> <1. WriteAddress> <NumberOfParameters> <NumberOfBytes> <Data>

Field	Bytes	Value	Meaning
FC	1	16 = 10 <sub>h</sub>	Request code
1. WriteAddress	2	(various)	Address of the first parameter to be written
NumberOfParameters	2	2 * m	Number of parameters to be written
NumOfBytes	1	4 * m	Number of data bytes
Data	2 * m	(various)	m parameter values to be written

*Structure of the positive response:* <FC> <ParamAddress> <NumberOfParameters>

Field	Bytes	Value	Meaning
FC	1	16 = 10 <sub>h</sub>	Response code
ParamAddress	2	(various)	Mirrored from request
NumberOfParameters	2	2	Mirrored from request

*Structure of the negative response* see chapter 6.2 "Protocol errors"

*Example of Premium in PL7* ► Write a target position (Profile Position operating mode):

The Modbus parameter address for the target position Ppp\_targetusr is 6940 (16#1B1C). The following command calls the Modbus service "Write Multiple Registers" (FC 16). The target position is stored in %MW25, the result of the execution of the function is stored starting at %MW35.

```
// Write two words to the parameteraddress 16#1B1C of
the device with the address (ADR)
WRITE_VAR (ADR, '%MW', 16#1B1C, 2, %MW25:2, %MW35:4);
```



*All parameters are transmitted as 4 byte values (32Bit). A list of all the parameters can be found in the product manual in the chapter "Parameters".*

### 5.2.4 FC 23 (ReadWrite Multiple Registers)

This function code allows data to be exchanged between master and slave in read and write modes.

*Structure of request:* <FC> <1. ReadAddress> <NumberOfParameters> <1. WriteAddress> <NumberOfParameters> <NumberOfBytes> <Data>

Field	Bytes	Value	Meaning
FC	1	23 = 17 <sub>h</sub>	Function code
1. ReadAddress	2	(various)	Address of the first parameter to be read
NumberOfParameters	2	2 * n	Number of the 16 bit parameters to be read
1. WriteAddress	2	(various)	Address of the first parameter to be written
NumberOfParameters	2	2 * m	Number of parameters to be written
NumOfBytes	1	4 * m	Number of data bytes
Data	4 * m	(various)	m parameter values to be written

*Structure of the positive response:* <FC> <NumberOfBytes> <Data>

Field	Bytes	Value	Meaning
FC	1	23 = 17 <sub>h</sub>	Response code
NumOfBytes	1	2 * n	Number of data bytes
Data	2 * n	n consecutive	n read parameter values

*Structure of the negative response* see chapter 6.2 "Protocol errors"



*All parameters are transmitted as 4 byte values (32Bit). A list of all the parameters can be found in the product manual in the chapter "Parameters".*

### 5.2.5 FC 43 (Read Device Identification)

This function code allows data for device identification to be read.

*Structure of request:* <FC> <MEI> <ReadDevID> <ObjID>

Field	Bytes	Value	Meaning
FC	1	43 = 2B <sub>h</sub>	Function code
MEI	1	14 = 0E <sub>h</sub>	Modbus Encapsulated Interface Type(subfunction)
ReadDevID	1	01	Read Device ID Codeall objects
ObjID	1	0x00	Object ID vendor name, product code, revision

*Structure of the positive response:* <FC> <MEI> <ReadDevID> <ConfLev><MoreFoll><NextObjID><NumOfObj><Data>

Field	Bytes	Value	Meaning
FC	1	43 = 2B <sub>h</sub>	Function code
MEI	1	14 = 0E <sub>h</sub>	Modbus Encapsulated Interface Type(subfunction)
ReadDevID	1	01	Read Device ID Codeall objects
ConfLev	1	02	Conformity Level fixed value
MoreFoll	1	00	More Follows fixed value, since length of frame < 255
NextObjID	1	00	Next Object ID fixed value because MoreFoll = 00
NumOfObj	1	03	Number of objects
Data	1	(various)	Object ID (1 byte, see following table) Object length (1 byte) Object data (various)

The following identification data can be read:

Object ID	Object Name	Value
00 <sub>h</sub>	vendor name	Manufacturer name
01 <sub>h</sub>	product code	„xxxxxxxxxx“ (see type code)
03 <sub>h</sub>	revision	"Vxx.yyy" (e.g. "V02.001")

*Structure of the negative response* see chapter 6.2 "Protocol errors"

### 5.3 Examples of function codes FC

NOTE: Parameters are read or written separately.

Exception: If Modbus parameter addresses are consecutive (Modbus address, Modbus address+2), a single request is sufficient to transmit the values.

*Example 1* Reading an entry in the error memory -> FLT\_err\_num (15362) / FLT\_class (15364) / FLT\_Time (15366) / FLT\_Qual (15368). Since all Modbus error information has consecutive addresses in ascending order, a single read request as described below is sufficient:

Field	Bytes	Value	Meaning
FC (Request Code)	1	3	Request code (Multiple Register READ)
ParamAddress	2	15362 dec (3C02 <sub>h</sub> )	First parameter address to be read
NumberOfParameters	2	4 * 2 = 8	Number of 16 bit parameters to be read = 8, i.e. reading 16 bytes of data.

Table 5.3 Example 1, FC3 Request

Field	Bytes	Value	Meaning
FC (Request Code)	1	3	Request code: Multiple Register READ
NumOfBytes	1	16	Number of bytes: 8 bytes of data
Data	16	32 bit value 32 bit value 32 bit value 32 bit value	FLT_err_num, 15362 (error number) FLT_class, 15364 (error class) FLT_Time, 15366 (error time) FLT_Qual, 15368 (error code)

Table 5.4 Example 1, FC3 Positive Response

*Example 2* Writing the values for software limit switch -> SPVswLimPusr (1544) / SPVswLimNusr (1546).

Since these parameters also have consecutive addresses (Modbus address, Modbus address+2), a single write command is sufficient:

Field	Bytes	Value	Meaning
FC (Request Code)	1	16	Request code (Multiple Register WRITE)
ParamAddress	2	1544 dec (608 <sub>h</sub> )	First parameter address to be written
NumberOfParameters	2	2 * 2 = 4	Number of parameters = 4 (8 bytes of data)
NumOfBytes	1	8	Number of bytes: 8 bytes of data
Data	8	32 bit value 32 bit value	SPVswLimPusr, 1544 SPVswLimNusr, 1546

Table 5.5 Example 2, FC16 Request

Field	Bytes	Value	Meaning
FC (Request Code)	1	16	Response code (Multiple Register WRITE)
ParamAddress	2	1544 dec (608 <sub>h</sub> )	Modbus parameter address
NumberOfParameters	2	2 * 2 = 4	Number of parameters = 4 (8 bytes of data)

Table 5.6 Example 2, FC16 Positive Response

## 5.4 Examples for standardized operating modes

### 5.4.1 Operating mode Profile Position

*Example* Node address 1.

Description Fieldbus command / parameter name (address)	Value
► Acceleration ramp 2000 min <sup>-1</sup> *s FC16 / RAMPacc (1556)	0000 07D0 <sub>h</sub>
► Deceleration ramp 4000 min <sup>-1</sup> *s FC16 / RAMPdecel (1558)	0000 0FA0 <sub>h</sub>
► Limitation of reference speed 6000 min <sup>-1</sup> FC16 / RAMPn_max (1554)	0000 1770 <sub>h</sub>
► Reference speed 4000 min <sup>-1</sup> FC16 / PPn_target (6942)	0000 0FA0 <sub>h</sub>
► Disable Voltage FC16 / DCOMcontrol (6914)	0000 0000 <sub>h</sub>
► Shut Down FC16 / DCOMcontrol (6914)	0000 0006 <sub>h</sub>
► Operation Enable FC16 / DCOMcontrol (6914)	0000 000F <sub>h</sub>
► Check operating state <sup>1)</sup> FC 3 / DCOMstatus (6916)	
◁ Operating state active	0000 0007 <sub>h</sub>
► Start operating mode FC16 / DCOMopmode (6918)	0000 0001 <sub>h</sub>
► Check operating mode FC 3 / _DCOMopmd_act (6920)	
◁ Operating mode active	0000 0001 <sub>h</sub>
► Store new reference position FC16 / PPP_targetusr (6940)	0000 0030 <sub>h</sub>
► Start absolute positioning FC16 / DCOMcontrol (6914)	0000 005F <sub>h</sub>
► Check target position FC 3 / DCOMstatus (6916)	
◁ Target position reached (Bit 10 = 1)	xxxx x4xx <sub>h</sub>
► Reset startbit FC16 / DCOMcontrol (6914)	0000 000F <sub>h</sub>

1) Must be checked cyclically.

### 5.4.2 Operating mode Profile velocity

*Example* Node address 1.

Description Fieldbus command / parameter name (address)	Value
► Acceleration ramp 2000 min <sup>-1</sup> *s FC16 / RAMPacc (1556)	0000 07D0 <sub>h</sub>
► Deceleration ramp 10000 min <sup>-1</sup> *s FC16 / RAMPdecel (1558)	0000 2710 <sub>h</sub>
► Limitation of the reference speed 10000 min <sup>-1</sup> FC16 / RAMPn_max (1554)	0000 2710 <sub>h</sub>
► Disable Voltage FC16 / DCOMcontrol (6914)	0000 0000 <sub>h</sub>
► Shut Down FC16 / DCOMcontrol (6914)	0000 0006 <sub>h</sub>
► Operation Enable FC16 / DCOMcontrol (6914)	0000 000F <sub>h</sub>
► Check operating state <sup>1)</sup> FC 3 / DCOMstatus (6916)	
◁ Operating state active	0000 0007 <sub>h</sub>
► Start operating mode FC16 / DCOMopmode (6918)	0000 0003 <sub>h</sub>
► Check operating mode FC 3 / _DCOMopmd_act (6920)	
◁ Operating mode active	0000 0003 <sub>h</sub>
► Set reference speed 1000 min <sup>-1</sup> FC16 / PVn_target (6938)	0000 03E8 <sub>h</sub>
► Check target speed FC 3 / DCOMstatus (6916)	
◁ Target speed reached (Bit 10 = 1)	xxxx x4xx <sub>h</sub>
► Set reference speed 0 min <sup>-1</sup> FC16 / PVn_target (6938)	0000 0000 <sub>h</sub>
► Check target speed FC 3 / DCOMstatus (6916)	
◁ Target speed reached (Bit 10 = 1)	xxxx x4xx <sub>h</sub>

1) Must be checked cyclically.

## 5.4.3 Operating mode Homing

*Example* Node address 1.

Description Fieldbus command / parameter name (address)	Value
▶ Reference speed for movement to limit switch 100 min <sup>-1</sup> FC16 / HMn (10248)	0000 0064 <sub>h</sub>
▶ Reference speed for clearance movement 10 min <sup>-1</sup> FC16 / HMn_out (10250)	0000 000A <sub>h</sub>
▶ Disable Voltage FC16 / DCOMcontrol (6914)	0000 0000 <sub>h</sub>
▶ Shut Down FC16 / DCOMcontrol (6914)	0000 0006 <sub>h</sub>
▶ Operation Enable FC16 / DCOMcontrol (6914)	0000 000F <sub>h</sub>
▶ Check operating state <sup>1)</sup> FC 3 / DCOMstatus (6916)	
◁ Operating state active	0000 0007 <sub>h</sub>
▶ Start operating mode FC16 / DCOMopmode (6918)	0000 0006 <sub>h</sub>
▶ Check operating mode FC 3 / _DCOMopmd_act (6920)	
◁ Operating mode active	0000 0006 <sub>h</sub>
▶ Select reference movement method, LimN (17) FC16 / HMmethod (6936)	0000 0011 <sub>h</sub>
▶ Start Homing FC16 / DCOMcontrol (6914)	0000 001F <sub>h</sub>
▶ Check Homing FC 3 / DCOMstatus (6916)	
◁ Drive has a valid reference point (Bit 12 = 1)	xxxx 1xxx <sub>h</sub>
▶ Reset startbit FC16 / DCOMcontrol (6914)	0000 000F <sub>h</sub>

1) Must be checked cyclically.

## 5.5 Examples of manufacturer-specific operating modes

### 5.5.1 Operating mode Electronic Gear

*Example* Node address 1.

Description	
Fieldbus command / parameter name (address)	Value
► Signal selection position interface FC16 / IOposInterfac (1284)	0000 0001 <sub>h</sub>
► Disable Voltage FC16 / DCOMcontrol (6914)	0000 0000 <sub>h</sub>
► Shut Down FC16 / DCOMcontrol (6914)	0000 0006 <sub>h</sub>
► Operation Enable FC16 / DCOMcontrol (6914)	0000 000F <sub>h</sub>
► Check operating state <sup>1)</sup> FC 3 / DCOMstatus (6916)	
◁ Operating state active	0000 0007 <sub>h</sub>
► Start operating mode (-2) FC16 / DCOMopmode (6918)	FFFF FFFE <sub>h</sub>
► Check operating mode FC 3 / _DCOMopmd_act (6920)	
◁ Operating mode active	FFFF FFFE <sub>h</sub>
► Activate gear with immediate synchronization FC16 / GEARreference (6948)	0000 0001 <sub>h</sub>
► Set denominator FC16 / GEARdenom (9734)	0000 0003 <sub>h</sub>
► Set numerator FC16 / GEARnum (9736)	0000 0002 <sub>h</sub>

1) Must be checked cyclically.

## 5.5.2 Operating mode Jog

*Example* Node address 1.

Description Fieldbus command / parameter name (address)	Value
▶ Speed slow movement to 100 min <sup>-1</sup> FC16 / JOGn_slow (10504)	0000 0064 <sub>h</sub>
▶ Speed slow movement to 250 min <sup>-1</sup> FC16 / JOGn_fast (10506)	0000 00FA <sub>h</sub>
▶ Disable Voltage FC16 / DCOMcontrol (6914)	0000 0000 <sub>h</sub>
▶ Shut Down FC16 / DCOMcontrol (6914)	0000 0006 <sub>h</sub>
▶ Operation Enable FC16 / DCOMcontrol (6914)	0000 000F <sub>h</sub>
▶ Check operating state <sup>1)</sup> FC 3 / DCOMstatus (6916)	
◁ Operating state active	0000 0007 <sub>h</sub>
▶ Start operating mode (-1) FC16 / DCOMopmode (6918)	FFFF FFFF <sub>h</sub>
▶ Check operating mode FC 3 / _DCOMopmd_act (6920)	
◁ Operating mode active	FFFF FFFF <sub>h</sub>
▶ Jog (clockwise rotation, slow) FC16 / JOGactivate (6930)	0000 0001 <sub>h</sub>
▶ Jog (clockwise rotation, fast) FC16 / JOGactivate (6930)	0000 0005 <sub>h</sub>

1) Must be checked cyclically.

5.6 Connection monitoring

⚠ WARNING

**LOSS OF CONTROL**

- Activate the time-out function. Without time-out, the system will not detect the interruption of the communication link.
- The shorter the time-out period the faster the detection of the interruption.

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

Two monitoring mechanisms are used on an ongoing basis on the master and slave sides to monitor the exchange of data for errors:

- Timeout monitoring
- Check of received characters

Timeout monitoring

A timeout period can be set on the slave side; during this time, a message must be received from the master (node guarding). If no message is received from the master within this period, the device will stop the process triggered via the fieldbus and perform a "Quick Stop".

The time interval for the timeout message can be set via the parameter `MBnode_guard` (5644). Value range:

- =0: Node Guarding off (factory setting)
- >0: Node guarding in ms (milliseconds)

If timeout monitoring is not used, an interruption of the communication link does not generate an error message. The controller keeps executing the most recently transmitted command.

Use the timeout function during normal network operation. Timeout monitoring can be switched off during troubleshooting. The manufacturer of the system must then provide alternative control paths for stopping and controlling the motor.

Check of received characters

The device checks fieldbus commands for transmission errors. If the transmit data could not be correctly received, e.g. because of a parity error, no acknowledgement is generated. This allows the master to detect a timeout.



## 6 Diagnostics and troubleshooting

For error handling purposes, two error types are to be distinguished:

- Communication errors (serial transmission errors),
- Protocol errors (Modbus-specific errors),

### 6.1 Communication errors

Communication errors include:

- Character timeout (time exceeded during transmission of characters),
- Parity errors,
- Framing errors (error in data frame),
- Overrun errors (overflow in receive register of the serial module).

If one of these errors occurs, the device no longer responds. The master generates a timeout error.

### 6.2 Protocol errors

The response to all protocol errors is an exception code. The response has the same function code as the normal response, but, in addition, the "MSB" is set. The function code is followed by a 1 byte exception code.

*Structure of the negative response* For FC3, FC8, FC16, FC23: <FC> <ExceptCode>

For FC43: <FC> <MEI> <ExceptCode>

Field	Bytes	Value	Meaning
FC	1	FC + 128 (80 <sub>h</sub> ) 03 <sub>h</sub> + 80 <sub>h</sub> = 83 <sub>h</sub> 08 <sub>h</sub> + 80 <sub>h</sub> = 88 <sub>h</sub> 10 <sub>h</sub> + 80 <sub>h</sub> = 90 <sub>h</sub> 17 <sub>h</sub> + 80 <sub>h</sub> = 97 <sub>h</sub> 2B <sub>h</sub> + 80 <sub>h</sub> = AB <sub>h</sub>	Response code for errors for: FC3 FC8 FC16 FC23 FC43
MEI (only FC43)	1	14	Modbus Encapsulated Interface Type(subfunction)
ExceptCode	1	01 <sub>h</sub> .. 04 <sub>h</sub>	01 <sub>h</sub> = invalid function 02 <sub>h</sub> = invalid data addresses 03 <sub>h</sub> = invalid data 04 <sub>h</sub> = slave device error

## 6.3 Handling errors

### 6.3.1 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 this is not the case, the device returns an exception code in response to the command. The error which has occurred can be read using the diagnostics function, see page 22.

*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 table with the error numbers can be found in the device manual in the chapter on diagnostics and troubleshooting.

## 7 Glossary

### 7.1 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

#### 7.1.1 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	-

#### 7.1.2 Mass

	lb	oz	slug	kg	g
lb	-	* 16	* 0.03108095	* 0.4535924	* 453.5924
oz	/ 16	-	* 1.942559*10 <sup>-3</sup>	* 0.02834952	* 28.34952
slug	/ 0.03108095	/ 1.942559*10 <sup>-3</sup>	-	* 14.5939	* 14593.9
kg	/ 0.45359237	/ 0.02834952	/ 14.5939	-	* 1000
g	/ 453.59237	/ 28.34952	/ 14593.9	/ 1000	-

#### 7.1.3 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 <sup>-3</sup>
dyne	/ 444822.2	/ 27801	/ 980.7	-	/ 100*10 <sup>3</sup>
N	/ 4.448222	/ 0.27801	/ 9.807*10 <sup>-3</sup>	* 100*10 <sup>3</sup>	-

#### 7.1.4 Power

	HP	W
HP	-	* 746
W	/ 746	-

## 7.1.5 Rotation

	$\text{min}^{-1}$ (RPM)	rad/s	deg./s
$\text{min}^{-1}$ (RPM)	-	$\ast \pi / 30$	$\ast 6$
rad/s	$\ast 30 / \pi$	-	$\ast 57.295$
deg./s	/ 6	/ 57.295	-

## 7.1.6 Torque

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

## 7.1.7 Moment of inertia

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

## 7.1.8 Temperature

	°F	°C	K
°F	-	$(\text{°F} - 32) \ast 5/9$	$(\text{°F} - 32) \ast 5/9 + 273.15$
°C	$\text{°C} \ast 9/5 + 32$	-	$\text{°C} + 273.15$
K	$(\text{K} - 273.15) \ast 9/5 + 32$	$\text{K} - 273.15$	-

## 7.1.9 Conductor cross section

AWG	1	2	3	4	5	6	7	8	9	10	11	12	13
mm <sup>2</sup>	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 <sup>2</sup>	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

## 7.2 Terms and Abbreviations

<i>Actual position</i>	Current position of moving components in the drive system.
<i>Address</i>	Storage which can be accessed by its unique number. See also Slave address.
<i>ASCII</i>	American Standard Code for Information Interchange. Standard for coding of characters
<i>Asynchronous error</i>	Error which is detected and signaled by the internal monitoring unit of the controller.
<i>CRC</i>	Cyclical Redundancy Check, used to check for errors
<i>Data frame</i>	Serially transmitted data packet with a unique identifier for the beginning and the end of the packet; the structure depends on the protocol used.
<i>Default value</i>	Factory setting.
<i>Direction of rotation</i>	Rotation of the motor shaft in a clockwise or counterclockwise direction of rotation. Clockwise rotation is when the motor shaft rotates clockwise as you look at the end of the protruding motor shaft.
<i>Drive system</i>	System consisting of controller, power stage and motor.
<i>Electronic gear</i>	Calculation of a new output speed for the motor movement based on the input speed and the values of an adjustable gear ratio; calculated by the drive system.
<i>Error</i>	Discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition.
<i>Error class</i>	Classification of errors into groups. The different error classes allow for specific responses to faults, for example by severity.
<i>Fatal error</i>	In the case of fatal error, the product is not longer able to control the motor, so that an immediate deactivation of the power stage is necessary.
<i>Fault</i>	Operating state of the drive caused as a result of a discrepancy between a detected (computed, measured or signaled) value or condition and the specified or theoretically correct value or condition.
<i>Fault reset</i>	A function used to restore the drive to an operational state after a detected error is cleared by removing the cause of the error so that the error is no longer active (transition from operating state "Fault" to state "Operation Enable").
<i>Fieldbus</i>	A bus optimized for data transmission between field devices. A fieldbus is "open", i.e. it is not proprietary (not supported by only one manufacturer). The parameter settings of the drive system can be requested and modified via the fieldbus, inputs can be monitored and outputs controlled and diagnostics and error monitoring functions activated.
<i>Half-duplex</i>	Bi-directional data transmission, but only in one direction at a time (not simultaneously).
<i>I/O</i>	Inputs/outputs
<i>Inc</i>	Increments
<i>Index pulse</i>	Signal of an encoder to reference the rotor position in the motor. The encoder returns one index pulse per revolution.
<i>Limit switch</i>	Switches that signal overtravel of the permissible range of travel.

<i>LRC</i>	Longitudinal Redundancy Check, used to check for errors
<i>LSB</i>	Least Significant Bit in a sequence of bits, e.g. a byte.
<i>Master</i>	Active bus device that controls the data traffic on the network.
<i>Node guarding</i>	Monitoring of the connection to the slave at an interface for cyclic data traffic.
<i>Parameter</i>	Device data and values that can be set by the user.
<i>Power stage</i>	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.
<i>Protocol</i>	A protocol specifies the format that data must have for transmission.
<i>PWM</i>	Pulse width modulation
<i>Quick Stop</i>	Function used to enable fast deceleration of the motor via a command or in the event of an error.
<i>RTU</i>	Remote Terminal Unit
<i>Slave</i>	Passive bus device that receives control commands and provides data to the master.
<i>Slave address</i>	Communication between master and slave is only possible after the assignment of unique addresses.
<i>Synchronous error</i>	Error signaled by the controller if it is unable to execute a command received from the master.
<i>Timeout</i>	Error which occurs when the maximum permissible period of time between a request and the response from a device is exceeded.
<i>Warning</i>	If the term is used outside the context of safety instructions, a warning alerts to a potential problem that was detected by a monitoring function. A warning is not an error and does not cause a transition of the operating state.
<i>Watchdog</i>	Unit that monitors cyclic basic functions in the product. The power stage is disabled and the outputs are switched off in the event of errors.

## 8 Index

### A

Abbreviations 39

### B

Baud rate 14

Before you begin

Safety information 9

### C

Character check 33

Commissioning 13

Communication errors 35

Connection monitoring 33

### D

Data bits 15

Data frame 19

Device address 17

Diagnostics 22, 35

Directives and standards 7

Documentation and literature references 7

### E

Electronic Gear

example 31

Example

Electronic Gear 31

Function codes FC 26

Homing 30

Jog 32

Profile Position 28

Profile Velocity 29

### F

Factory settings 14

Fields, data frame 19

Function codes 21

Function tests 16

### G

Glossary 37

### H

Half-duplex operation 14

Homing

example 30

### I

Introduction 7

**J**

Jog  
    example 32

**L**

Literature 7

**M**

Master 17  
Modbus message 19  
Modbus network 17  
Modbus transmission technology 18

**N**

Network operation 16  
Node address 14  
Node guarding 15

**O**

Operation 17

**P**

Parameter addresses, hexadecimal, decimal 18  
parity 15  
Profile Position  
    example 28  
Profile Velocity  
    example 29  
Protocol errors 35  
Protocols for Modbus, in general 18

**R**

Read Device Identification 25  
Read Multiple Registers 21  
Read Write Multiple Registers 24  
Request 18, 19  
Response 18, 19

**S**

Settings 14  
Slave 17  
stop bits 15  
Synchronous errors 36

**T**

Telegram 19  
Terms 39  
Timeout 33  
Transmission format 14  
Troubleshooting 35

**U**

Units and conversion tables 37

**W**

Word sequence 15

Write Multiple Registers 23

