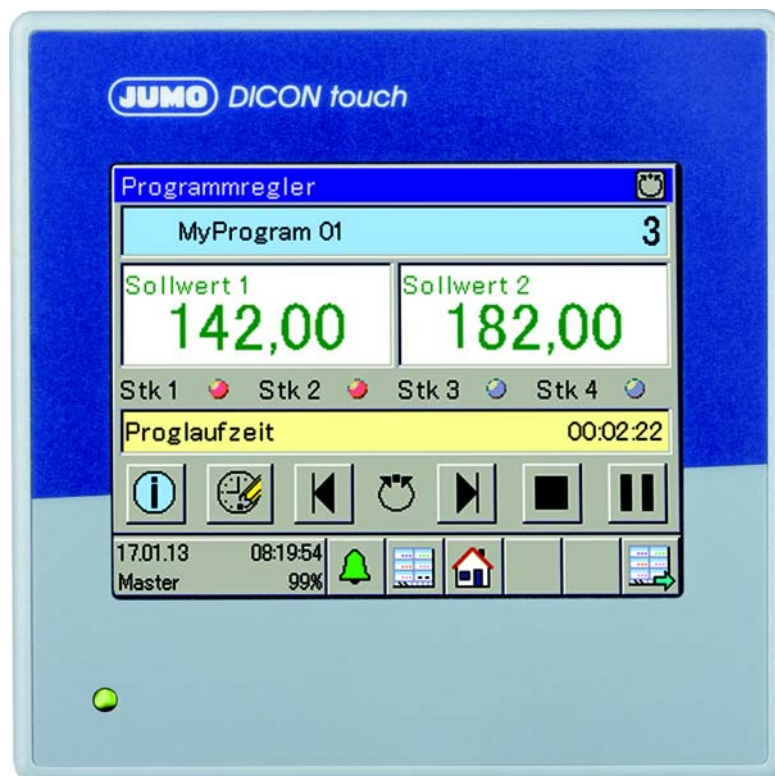


JUMO DICON touch

Two-channel process and program controller with
with paperless recorder and touchscreen



B 703571.2.0

Interface description Modbus



1	Safety information	7
1.1	Warning symbols	7
1.2	Note signs	7
2	Modbus protocol description	9
2.1	Master-slave principle	9
2.2	Transmission media for Modbus	10
2.3	Structure of a Modbus telegram	10
2.4	Function codes	11
2.4.1	Reading n words	12
2.4.2	Writing one word	13
2.4.3	Writing n words	14
2.5	Data types	15
2.6	Examples of data transmission	17
2.6.1	Integer values	17
2.6.2	Floating-point values	18
2.6.3	Character strings (texts)	19
2.7	Checksum (CRC16)	21
2.8	Error messages	22
2.8.1	Modbus error codes	22
2.8.2	Error messages for invalid values	23
2.8.3	Error codes as integer return values	24
3	Interfaces	29
3.1	Location of interfaces	29
3.2	Interfaces	30
3.2.1	USB device, USB host, and COM 1 interfaces as standard	30
3.2.2	COM 2 interface can be expanded using optional boards	30
3.2.3	Terminating resistors	31
3.3	Configuring interfaces	32
4	Modbus via a serial interface	37
4.1	Modbus timing for serial interface RS422/485	37
4.2	Modbus master operation	38
4.3	Modbus slave operation	40
5	Modbus and other protocols over Ethernet	41
5.1	Modbus/TCP protocol	41
5.2	Networking with Modbus/TCP	43
5.3	Modbus master with Modbus/TCP	43
5.4	Modbus slave with Modbus/TCP	44
5.5	HTTP	44

Contents

5.6	Browser connection and web server	45
5.7	E-mail (SMTP and POP3)	46
6	User frames	49
6.1	General information	49
6.2	Structure of the reading and writing processes	49
6.3	Compiling Modbus frames	51
6.3.1	Modbus frames for reading	51
6.3.2	Modbus frames for writing	54
6.4	Examples for the data transmission options with frames	57
7	Modbus address tables	59
7.1	Data types and access types	59
7.2	Configuration data and parameters	60
7.2.1	Device software version	60
7.2.2	Device name	60
7.2.3	Limit values for limit value monitoring 1 to 16	60
7.2.4	Ramp slope, controllers 1, 2	61
7.2.5	Controller 1, parameter block 1	61
7.2.6	Controller 1, parameter block 2	61
7.2.7	Controller 1, parameter block 3	62
7.2.8	Controller 1, parameter block 4	63
7.2.9	Controller 2, parameter block 1	63
7.2.10	Controller 2, parameter block 2	64
7.2.11	Controller 2, parameter block 3	65
7.2.12	Controller 2, parameter block 4	65
7.2.13	Setpoint values, controller 1	66
7.2.14	Setpoint values, controller 2	66
7.2.15	Filter time constant, offset	66
7.2.16	Analog and digital flags 1 to 8	67
7.2.17	Controller type, control direction	67
7.3	Process values	68
7.3.1	Transmission times of the frames	68
7.3.2	Date, time	68
7.3.3	Measuring inputs 1 to 4	68
7.3.4	Controller 1	69
7.3.5	Controller 2	69
7.3.6	Setpoint values, controller 1	69
7.3.7	Setpoint values, controller 2	70
7.3.8	Program controller	70
7.3.9	Math, logic	72
7.3.10	Digital inputs 1 to 7 and their alarms	73
7.3.11	Limit value monitoring, actual values and alarms	73
7.3.12	Digital control signals	74

7.3.13 Timers	75
7.3.14 Analog and digital flag	76
7.3.15 Device operation	76
7.3.16 Outputs	78
7.3.17 External analog and digital signals	78
7.3.18 Modbus error	80
7.3.19 Data, controller 1	80
7.3.20 Data, controller 2	80
7.3.21 Program editor memory	81
7.3.22 Program editor memory, section 1, 2 to 50	81
7.3.23 Program name	82
7.3.24 Compiled reading frames	82
7.3.25 Compiled writing frames	83

Contents

1.1 Warning symbols



DANGER!

This symbol indicates that **personal injury caused by electrical shock** may occur if the respective precautionary measures are not carried out.



WARNING!

This symbol in connection with the signal word indicates that personal injury may occur if the respective precautionary measures are not carried out.



CAUTION!

This symbol in connection with the signal word indicates that **damage to assets or data loss** will occur if the respective precautionary measures are not taken.



CAUTION!

This symbol indicates that **components could be destroyed** by electrostatic discharge (ESD = Electro Static Discharge) if the respective cautionary measures are not taken. Only use the ESD packages intended for this purpose to return device inserts, assembly groups, or assembly components.



READ DOCUMENTATION!

This symbol – placed on the device – indicates that the associated **device documentation has to be observed**. This is necessary to recognize the kind of the potential hazards as well as the measures to avoid them.

1.2 Note signs



TIP!

This symbol refers to **important information** about the product, its handling, or additional use.



REFERENCE!

This symbol refers to **further information** in other sections, chapters, or manuals.



FURTHER INFORMATION!

This symbol is used in the tables and refers to **further information** in connection with the table.



DISPOSAL!

This device and the batteries (if installed) must not be disposed in the garbage can after use! Please ensure that they are disposed properly and in an **environmentally friendly manner**.

1 Safety information

2.1 Master-slave principle

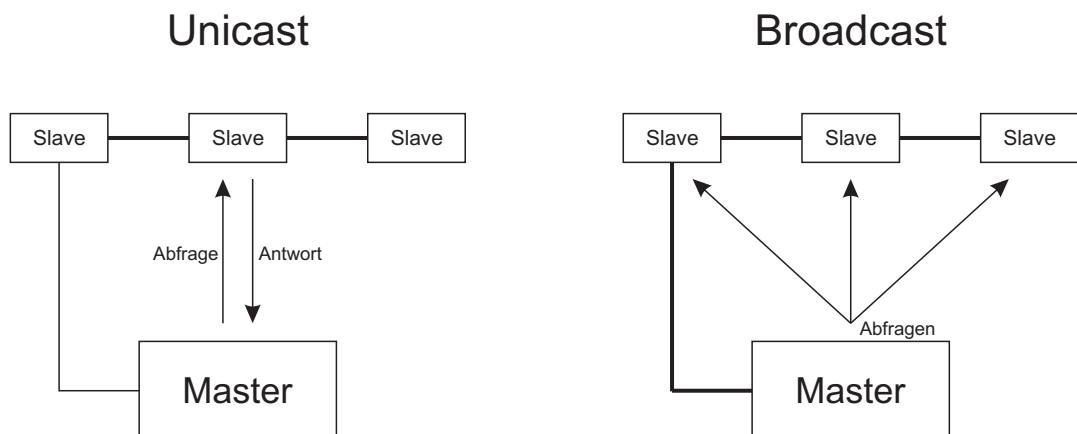
Communication between a master (e.g. a SCADA system or PLC) and a device type 703571 as slave takes place on the basis of the master-slave principle in the form of a data query – response. Depending on the transmission technology, the bus users (master and slaves) are addressed as follows:

- For **Modbus via a serial interface**, all slaves are identified by their device address (1 to 254). Master devices do not need an address.
- For **Modbus via Ethernet**, users are identified by their IP address. Slave responses are sent to the IP address for the master.

⇒ Chapter 4 "Modbus via a serial interface", page 37

⇒ Chapter 5 "Modbus and other protocols over Ethernet", page 41

The master controls data exchange by cyclically querying the slaves on the overall bus. The slaves (for example, device type 703571) have a response function only. The master has write and read access to the slaves. This enables data to be communicated between master and slave devices in real-time. Slaves cannot communicate with one another directly. In order to transmit data from slave to slave, the master must extract data from one slave and transfer it to the next.



As a rule, the master directs its queries to individual slaves. For this, it must communicate with the individual slaves by means of their Unicast address. Queries can also be directed to all slaves on the bus in the form of a broadcast. In this case, the broadcast address is used as the slave address. To avoid data collisions, slaves do not respond to broadcast queries. For this reason, function codes are only used for writing data. Broadcasts cannot be used with function codes for reading data.



TIP!

Device type 703571 can be operated as a master or slave.

2 Modbus protocol description

2.2 Transmission media for Modbus

Serial interface

For data communication via a **serial interface**, the Modbus specification provides the transmission modes **RTU mode (Remote Terminal Unit)** and ASCII mode (transmission of data in ASCII format). Device type 703571 supports the **RTU mode** only. Here, the data are transmitted via the serial bus in binary format (RS422/485).

⇒ Chapter 4 "Modbus via a serial interface", page 37

Ethernet

Use of an **Ethernet network** is possible with the aid of **Modbus/TCP**. The Modbus data are encapsulated in the form of Modbus/TCP telegrams in TCP segments of the TCP/IP protocol family. In this way, it is possible to transmit Modbus/TCP telegrams via Ethernet.

⇒ Chapter 5 "Modbus and other protocols over Ethernet", page 41

2.3 Structure of a Modbus telegram

Modbus telegrams have the following structure:

Slave address	Function code	Data field	Checksum CRC
1 byte	1 byte	x bytes	2 bytes

Every telegram has four fields:

Slave address	Device address of a specific slave
Function code	Function selection (read/write words)
Data field	Contains the information (depending on the function code) - word address/bit address - number of words/number of bits - word value(s)/bit value(s)
Checksum	Detection of transmission errors

2.4 Function codes

Function overview

The functions described in the following (from the Modbus Standard) are available for extracting measured values, device and process data, and for writing data.

Function code		Function	Limit
Hex.	Dec.		
03 or 04	3 or 4	Reading n words	Max. 127 words (254 bytes)
06	6	Writing one word	Max. 1 word (2 bytes)
10	16	Writing n words	Max. 127 words (254 bytes)

**TIP!**

If the device does not respond to these functions (see requirements for communication) or outputs an error code (see error codes):

The information in Chapter 2.8.1 "Modbus error codes", page 22 may help during troubleshooting.

2 Modbus protocol description

2.4.1 Reading n words

This function is used to read n words, starting from a specific address.

Data query

Slave address	Function code 0x03 or 0x04	Address of first word	Number of words	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave address	Function code 0x03 or 0x04	Number of bytes read	Word value(s)	Checksum CRC
1 byte	1 byte	1 byte	x bytes	2 bytes

Example

This example is about reading the time (hours, minutes, and seconds). Starting from address 0x11E6, it is stored in three words.

⇒ Chapter 7 "Modbus address tables", page 59

Hex code for the data query:

01	03	11 E6	00 03	E1 00
Slave	Function code	Address of 1st word	Number of words	CRC

Hex code for the response:

01	03	06	00 0A	00 03	00 01	88 B4
Slave	Function code	Bytes read	10	03	01	CRC
			Hours	Minutes	Seconds	

2 Modbus protocol description

2.4.2 Writing one word

The data blocks for the data query and response are identical when writing a word.

Data query

Slave address	Function code 0x06	Word address	Word value	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave address	Function code 0x06	Word address	Word value	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example

In this example, the digital control signal 1 is to be set in the device. Here, the slave address of the device is 1, the word address is 0x12EF, and the value to write should be the logical value "1".

⇒ Chapter 7.3.12 "Digital control signals", page 74

Hex code for the data query:

01	06	12 EF	00 01	7C 87
Slave	Function code	Word address	Value	CRC

Hex code for the response:

01	06	12 EF	00 01	7C 87
Slave	Function code	Word address	Value	CRC

2 Modbus protocol description

2.4.3 Writing n words

Data query

Slave address	Function code 0x10	Address of first word	Number of words	Number of bytes	Word value(s)	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	1 byte	x bytes	2 bytes

Response

Slave address	Function code 0x10	Address of first word	Number of words	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example

Starting from Modbus address 0x13EE, the four values 1,0,1,0 should be written on the first four external digital inputs.

⇒ Chapter 7.3.17 "External analog and digital signals", page 78

Hex code for the data query:					Word 1	Word 2	Word 3	Word 4	
01	10	13 EE	00 04	08	00 01	00 00	00 01	00 00	A7 00
Slave	Function code	Address 1st word	Number of words	Number of bytes					CRC

Hex code for the response:

01	10	13 EE	00 04	A5 7B
Slave	Function code	Address 1st word	Number of words	CRC

2 Modbus protocol description

2.5 Data types

Data type	Description	Access	Possible function codes	Number of Modbus register								
Byte	Low byte of a word as an integer value; the high byte is not used Value ranges: 0 to 255 for unsigned data -127 to 127 for signed data	read only	03, 04	1								
		read/write	03, 04, 06, 16									
Word	Word as integer value Value ranges: 0 to 65535 for unsigned data -32767 to 32767 for signed data	read only	03, 04	1								
		read/write	03, 04, 06, 16									
Float	2 words as 32-bit floating-point number with coding to IEEE 754, whereby it must be kept in mind that bytes 1 and 2 are interchanged with bytes 3 and 4 during transmission S = Sign bit E = Exponent (two's complement) M = 23-bit normalized mantissa IEEE 754 standard coding:	read only	03, 04	2								
		read/write	03, 04, 16									
	<table border="1"> <tr> <td>Byte 1</td> <td>Byte 2</td> <td>Byte 3</td> <td>Byte 4</td> </tr> <tr> <td>SEEEEEEE</td> <td>EMMMMMMM</td> <td>MMMMM- MMM</td> <td>MMMMMMMM</td> </tr> </table>	Byte 1	Byte 2	Byte 3	Byte 4	SEEEEEEE	EMMMMMMM	MMMMM- MMM	MMMMMMMM			
	Byte 1	Byte 2	Byte 3	Byte 4								
	SEEEEEEE	EMMMMMMM	MMMMM- MMM	MMMMMMMM								
	Modbus coding of floating-point variables in device type 703571:											
	Address of the 1st Modbus variable register		Address of the 2nd Modbus variable register									
	Byte 3	Byte 4	Byte 1	Byte 2								
	MMMMMMMM	MMMMM- MMM	SEEEEEEE	EMMMMMMM								
	When creating customer-specific applications, the correct byte sequence in storage format needs to be checked. Many compilers use the following storage format:											
Byte 4	Byte 3	Byte 2	Byte 1									
MMMMMMMM	MMMMM- MMM	EMMMMMMM	SEEEEEEE									
Address x	Address x+1	Address x+2	Address x+3									

2 Modbus protocol description

Data type	Description	Access	Possible function codes	Number of Modbus register
Uint32	2 words (4 bytes) as unsigned integer value Value range: 0 to 4,294,967,295	read only	03, 04	2
		read/write	03, 04, 16	
Char[60]	Character string for up to 20 Unicode characters in UTF-8 coding with up to 3 bytes per character The character string thus consists of up to 60 bytes. Each of the 30 Modbus register words contains 2 successive bytes of the character string. Note that the character string must always have "\0" (ASCII code 0x00) as the terminating code.	read only	03, 04	30
		read/write	03, 04, 06, 16	
Bool	Least significant bit in a word as bit value 0000 0000 0000 0001 = 1 or TRUE 0000 0000 0000 0000 = 0 or FALSE	read only	03, 04	1
		read/write	03, 04, 06, 16	

2 Modbus protocol description

2.6 Examples of data transmission

The function 0x03 (reading) or 0x04 (writing one word) is used to extract integers, floating-point values, and text values.

Data query

Slave address	Function code 0x03 or 0x04	Address of first word	Number of words	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Integer values are transmitted over Modbus in the following format:
First the high, then the low byte.

Response

Slave address	Function code 0x03 or 0x04	Number of bytes read	Word value(s)	Checksum CRC
1 byte	1 byte	1 byte	x bytes	2 bytes

2.6.1 Integer values

Example

In this example, the parameter block number of controller 1 at address 0x1762 is to be read.

Data query:

01	03	17 62	00 01	20 60
Slave	Function code	Address of 1st word	Number of words	CRC

Response (values in Modbus integer format):

01	03	02	00 01	79 84
Slave	Function code	Bytes read	Integer value	CRC

2 Modbus protocol description

After being transmitted from the device, the bytes for the floating-point value must be interchanged accordingly. Many compilers (e.g. Microsoft Visual C++) store the floating-point values in the following sequence:

Floating-point value

Address x	Address x+1	Address x+2	Address x+3
MMMMMMMM	MMMMMMMM	SEEEEEEE	EMMMMMMM
Byte 4	Byte 3	Byte 2	Byte 1



TIP!

The sequence of the bytes depends on how floating-point values are saved in the application concerned. It may be necessary for the bytes to be interchanged in the interface program accordingly.

2.6.3 Character strings (texts)

Character strings are transmitted in ASCII format.



TIP!

A "\0" (ASCII code 0x00) must always be transmitted as the terminating code. Subsequent characters have no meaning.

Since texts are transmitted word by word (16-bit register), 0x00 is also added to the end for an uneven number of bytes (incl. "\0").

The maximum lengths for character strings specified in the address tables contain the final "\0". This means that, in the case of "Char 60", the text must not be longer than 60 bytes.

In the case of 19 Unicode characters having a length of 3 bytes each, only 2 bytes remain for the 20th character. 1 byte is needed for the terminating code.

Example

Querying the text from address 0x1000; this address holds the character string for the device name "Anlage Nord" (ASCII code: 0x41 0x6E 0x6C 0x61 0x67 0x65 0x20 0x4E 0x6F 0x72 0x64).

Hex code for the query:

01	03	10 00	00 07	00 C8
Slave	Function code	Address 1st word	Number of words	CRC

Hex code for the response:

Hex code for the response:			Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7									
01	03	00 0E	41	6E	6C	61	67	65	20	4E	6F	72	64	00	00	AA	AA	C5 DF
Slave	Function code	Bytes read	A	n	l	a	g	e	space	N	o	r	d					CRC

2 Modbus protocol description



TIP!

The value (here: AA) before the CRC sum (here: C5 DF) is not considered, since it follows the terminating code "\0".

2.7 Checksum (CRC16)

Calculation principle

Transmission errors are detected with the aid of the checksum (CRC16). If an error is detected during evaluation, the device concerned does not respond.

CRC = 0xFFFF	
	CRC = CRC XOR BytesOfMessage
	For (1 to 8)
	CRC = SHR(CRC)
	if (flag shifted to the right = 1)
	then
	CRC = CRC XOR 0xA001
	else
while (not all BytesOfMessage processed);	



TIP!

The low byte of the checksum is transmitted first!

Example: The CRC16 checksum CC DD is transmitted and represented in the sequence DD CC.

2 Modbus protocol description

2.8 Error messages

2.8.1 Modbus error codes

Requirements for Modbus communication

The following conditions must be met for a slave to receive, process, and respond to queries:

- Baud rate and/or data format of master and slave must match.
- The correct slave address must be used in the query.
- Slave devices respond only after a successful checksum check of the query by the slave. Otherwise, the query is rejected by the slave.
- The data query from the master must be complete and conform to the Modbus protocol.
- The number of words to be read must be greater than 0.

Error codes

If the data query from the master has been received by the slave without transmission errors but cannot be processed, the slave responds with an error code. The following error codes may occur:

- 01 = Invalid function; for the function codes that device type 703571 supports, see Chapter 2.4 "Function codes", page 11.
- 02 = Invalid address or too many words, or bit should be read or written
- 03 = Value is outside the admissible range
- 08 = Value is write-protected

Response to malfunction

Slave address	Function XX OR 80h	Error code	Checksum CRC
1 byte	1 byte	1 byte	2 bytes

The function code is ORed with 0x80. This sets the most significant byte (msb) to 1.

Example

Data query:

01	06	23 45	00 01	52 5B
Slave	Write word	Word address	Word value	CRC

Response (with error code 2):

01	86	02	C3 A1
Slave	OR function	Errors	CRC

Response with error code 02, because the address 0x2345 does not exist.

2 Modbus protocol description

2.8.2 Error messages for invalid values

For measured values in floating-point format, the error number itself is displayed in the value, i.e. instead of the measured value alone, the error number is returned as part of the value.

Error code for floating-point values	Errors
1.0×10^{37}	Underrange
2.0×10^{37}	Overrange
3.0×10^{37}	Not a valid input value
4.0×10^{37}	Division by zero
5.0×10^{37}	Mathematical error
6.0×10^{37}	Invalid terminal temperature of thermocouple
7.0×10^{37}	Invalid floating-point value
8.0×10^{37}	Integrator or statistics destroyed

Example

Reading in the analog input 1 (IN8) of device type 703571 at Modbus address 0x11E9:

Data query:

08	03	11 E9	00 02	10 5A
Slave	Function code	Word address	Number of words	CRC

Response:

08	03	04	8E 52	7D B4	C8 ED
Slave	Function code	Bytes read	Error code	CRC	

The measured value 0x7DB48E52 ($=3.0 \times 10^{37}$) received from analog input 1 indicates that this is an invalid input value.

The screenshot shows a floating-point value analysis tool. The value is 3.0E37. The sign is +1, the exponent is 2^{124} , and the mantissa is 1.4105932712554932. The encoded value is 0, and the binary representation is 01111101101101001000111001010010. The decimal representation is 3.0E37, the binary representation is 01111101101101001000111001010010, the hexadecimal representation is 0x7db48e52, and the value after casting to double precision is 3.000000106909804E37.

2 Modbus protocol description

2.8.3 Error codes as integer return values

For some longer processes (e.g. sending e-mail or active transmission of frames as Modbus master), an error code is entered in a result field or event list at the end.

Chapter 7.3.18 "Modbus error", page 80

Error codes

Error code	Description
Error list: Program memory management	
1	Program cannot be created
2	Program does not exist
3	Program cannot be deleted
4	Section cannot be deleted
5	Checksum cannot be stored
6	Checksum cannot be read
7	Program cannot be copied
8	Section cannot be copied
9	Program checksum error
10	Program pointer tab. checksum error
11	Program memory end
12	Section does not exist
13	Repeat jump marks
Error list: General input and output	
14	Please use the ENTER key to confirm
15	Invalid number of digits
16	The entry contains invalid characters
17	Value outside of limits
18	Section not programmed correctly
19	Password error
Error list: Keypad and program lock	
26	Keypad is locked
27	Programming is locked
28	Write error in the ser. EEPROM (calib)
29	Hardware error: MANUAL + AUTO locked
30	Editing during active program not permitted
31	Copying during active program not permitted
32	MANUAL not allowed during AUTO lead time
33	Section change! Image update needed
34	No DB number, image update from PLC
35	No DB number for process values from PLC

2 Modbus protocol description

Error code	Description
36	Printer busy or not ready
37	Setpoint value 1 not programmed
38	Set up printer (config. / interface)
39	Possible only when device in MANUAL mode
40	Autotuning already running
41	Time axis elapsed or not programmed
42	Time axis cannot be copied
43	Time axis does not exist
44	Program change is locked
45	MANUAL mode is locked
46	Program start is locked
Error list: Interface processing	
47	Incorrect response length
48	Timeout error (no response)
49	Error reported in telegram protocol
50	Checksum error
51	Parity error
52	Framing error
53	Interface buffer full
54	Addressing error (e.g. address does not exist)
55	Incorrect or unexpected command
Error list: Event processing	
60	Event could not be created
61	Event setting failed
62	Event clear failed
63	Event wait failed
64	Event close failed
65	Event open failed
66	Sync error between group and data manager
Error list: Message processing	
70	No queue memory available
71	Message queue cannot be opened
72	Message pool cannot be created
73	Memory cannot be requested from memory pool
74	Message cannot be sent
Error list: Processing of MQX functions	
80	Task creation failed
81	Hardware timer not created

2 Modbus protocol description

Error code	Description
Error list: Flash memory processing	
90	Flash memory write error
Error list: Other errors	
100	Undefined error
101	Division by zero
102	RAM memory location cannot be detected
103	RTC runtime exceeded
104	ID does not exist
105	Index too large (overflow)
106	Invalid data
107	Invalid parameter
109	String without 0 character
110	Timeout during initialization
111	Value must not be written to
112	Log entry with error bits that trigger debug mode
Error list: Sending e-mail via modem and Ethernet	
120	Step error in state automation
121	Invalid response length
122	No CONNECT from modem
123	FCS checksum incorrect
124	Unexpected value or response
125	.conf request not accepted
126	No .conf request from other end
127	No CHAP request from other end
128	Response timeout
129	Unknown modem response
130	Unexpected OK from modem
131	Unexpected CONNECT from modem
132	Unknown frame received
133	Unexpected PROTOCOL from modem
134	Unexpected COMPRESS from modem
135	Invalid PPP packet received
136	Unexpected BUSY from modem
137	Unknown authentication protocol
138	Ignored LCP option
139	Unexpected DELAYED from modem
140	Unexpected NODIALTONE
141	Unknown PPP protocol

2 Modbus protocol description

Error code	Description
142	Unknown PAP code
143	Ignored IPCP option
144	Ignored IPCP code
145	Unknown CHAP code
146	IP checksum incorrect
147	Unknown IP protocol
148	Unknown ICMP type
149	Unknown LCP type
150	Received as client DNS query
151	Unknown DNS error
152	DNS response is split
153	No IP received via DNS
154	Unknown UDP port
155	TCP checksum incorrect
156	Incorrect TCP port
157	Unknown TCP-SYN option
158	TCP port not used
159	Unknown POP3 response
160	Unknown SMTP response
161	Unknown DNS name
162	No MD5 requested for CHAP
163	Authentication error
164	Canceled from other end
165	Error creating TCP socket
166	Error binding TCP socket
167	Error during TCP connect
168	Error sending TCP telegram
169	Error closing TCP socket
170	Error during TCP listen
171	Reset during TCP accept
172	Error during TCP accept
173	SMTP server reports syntax error
174	TCP socket is already closed
175	Incorrect frame configuration
Error list: File system processing	
200	Error installing partition manager
201	Error installing MFS file system
202	Error uninstalling partition manager

2 Modbus protocol description

Error code	Description
203	Error uninstalling MFS file system

3.1 Location of interfaces

Device type 703571 features an interface RS485 (COM 1) as standard. It is designed for transmitting data using the Modbus protocol (slave or master) and for connecting an ER8 relay module.

An RS422/485, PROFIBUS-DP, or Ethernet interface are also available for retrofitting in the COM 2 expansion slot. The following optional boards are available:

- Serial interface RS422/485 for Modbus RTU (part no. 00581172)
- PROFIBUS-DP (part no. 00581173)
- Ethernet (part no. 00581174)

Both serial interfaces can be operated with the Modbus protocol (Modbus RTU; slave, or master) and for connecting an ER8 relay module.



TIP!

The device's nameplate provides information on which optional interfaces were assembled **when delivered**.

Information on this can be found in the chapter "Identifying the device version" in the operating manual B 703571.7 or the data sheet T 703571.

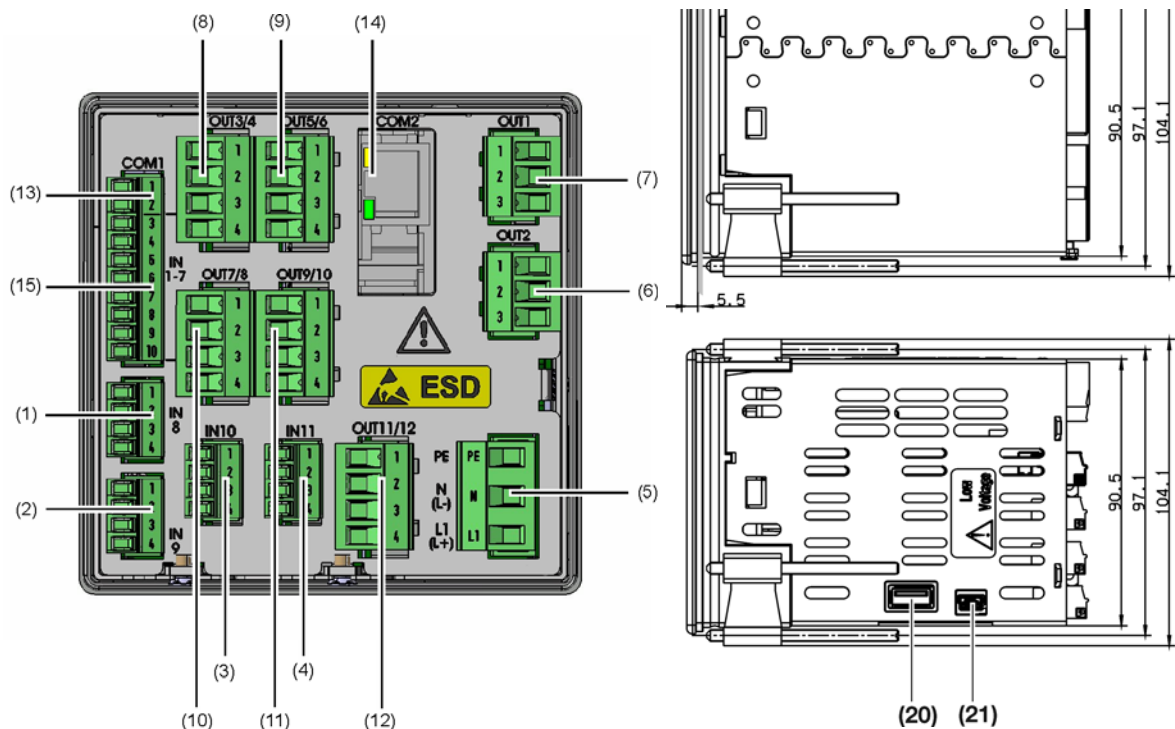


TIP!

Optional interfaces can also be subsequently fitted **by the user**.

Information on this can be found in the chapter "Retrofitting optional boards" in the operating manual B 703571.0.

Overview of connections





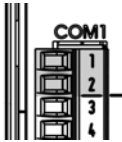
(13) COM 1 for serial interface RS485 (Modbus RTU)

(14) Expansion slot COM 2 for serial interfaces RS422/485 (Modbus RTU), PROFIBUS-DP, or Ethernet

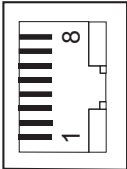


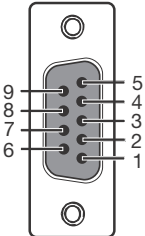
3 Interfaces

3.2 Interfaces

3.2.1 USB device, USB host, and COM 1 interfaces as standard

Connection	(Connection element)	Symbol and terminal designation
USB device interface	(21)	
USB host	(20)	
COM 1 serial interface RS485 (galvanically isolated)	(13)	 <div style="display: inline-block; vertical-align: middle; margin-left: 10px;"> <p>1 TxD+/RxD+ Transmission/received data +</p> <p>2 TxD-/RxD- Transmission/received data -</p> </div>

3.2.2 COM 2 interface can be expanded using optional boards

Connection	(Connection element)	Symbol and terminal designation
Ethernet (LAN)	(14)	 <div style="display: inline-block; vertical-align: middle; margin-left: 10px;"> <p>1 TX+ Transmission data +</p> <p>2 TX- Transmission data -</p> <p>3 RX+ Received data +</p> <p>6 RX- Received data -</p> </div>
Serial interface RS422 (galvanically isolated)		 <div style="display: inline-block; vertical-align: middle; margin-left: 10px;"> <p>1 RxD+ Received data +</p> <p>2 RxD- Received data -</p> <p>3 TxD+ Transmission data +</p> <p>4 TxD- Transmission data -</p> </div>
Serial interface RS485 (galvanically isolated)		 <div style="display: inline-block; vertical-align: middle; margin-left: 10px;"> <p>3 TxD+/RxD+ Transmission/received data +</p> <p>4 TxD-/RxD- Transmission/received data -</p> </div>
PROFIBUS-DP		 <div style="display: inline-block; vertical-align: middle; margin-left: 10px;"> <p>3 RxD/TxD-P (B) Transmission/received data +</p> <p>5 DGND Ground</p> <p>6 VP (+5 V) Voltage supply</p> <p>8 RxD/TxD-N (A) Transmission/received data -</p> </div>



TIP!

A twisted connecting cable with shielding must be used to connect the RS422/485 interface; to the greatest extent possible, it should not be laid close to current-carrying components or cables and its shielding should be grounded on one side. To avoid transmission errors, only the signals listed above may be connected.



TIP!

An RJ45 patch/crossover cable (Cat5 or higher) must be used to connect the LAN interface.

3.2.3 Terminating resistors



TIP!

To ensure fault-free operation, terminating resistors are required at the start and end of an RS422/485 transmission path.

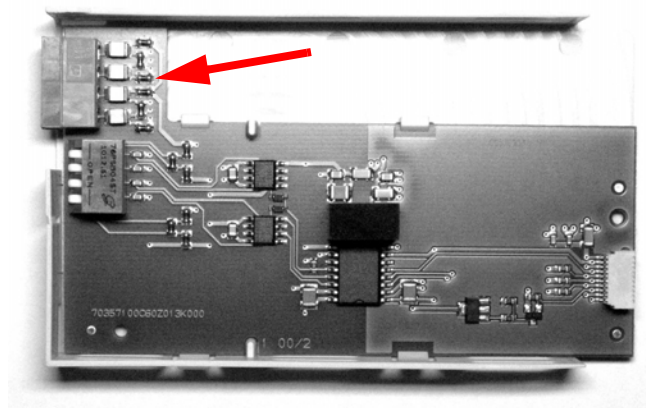
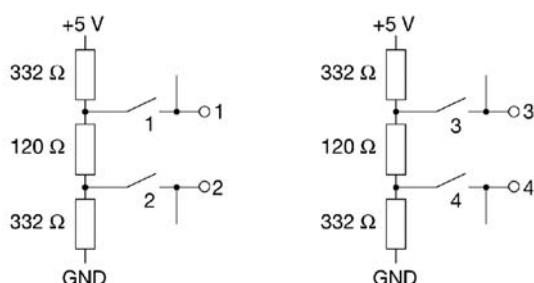
Optional boards for serial interfaces RS422/485 have integrated terminating resistors. The terminating resistors can be activated and deactivated via the DIP switch next to the connection port on the optional board. For the RS422/485 COM 1 interface, fixed terminating resistors are pre-integrated.

Correct installation/configuration of the terminating resistors at the ends of the bus line must be assured before starting up an RS422/485 interface.

The following table shows the positions of the DIP switches on the RS422/485 optional board when the terminating resistors are activated/deactivated.

RS422/485 without terminating resistors (deactivated)	
RS422/485 with terminating resistors (activated)	

Terminating resistors for optional boards



3 Interfaces

3.3 Configuring interfaces

Important information

**WARNING!**

After every configuration change, the device restarts any functions that have been affected by the changes.
Analog and digital inputs may assume undesired states during the startup process. For this reason, configuration changes must never be made while a system is operating.

**CAUTION!**

Incorrect installation or the wrong settings on the device can cause the system to assume an unexpected operating status.
This can disrupt processes or result in damage.
For this reason, it is always necessary to provide safety devices that are independent of the device and to allow settings to be made only by technical personnel.

**TIP!**

When changing configuration data that are relevant for the data monitoring and recording function, data recording is terminated and a new recording session begun.

**TIP!**

Changes to the configuration settings described in this chapter can be made directly on the device or by means of the JUMO PC setup program.

**TIP!**

Changing settings in the "Configuration" menu is possible only if a user is logged in with the appropriate user rights.

**TIP!**

Changes to the configuration only come into effect on exiting the configuration menu ("Exit" menu item or "Close window" button).

Operation, configuration, and parameterization procedures are explained in detail in the operating manual for device type 703571. Information about user login is also provided there.

⇒ B703571.0

Settings for serial interfaces

For all user devices on a bus to be able to communicate with one another, their interface settings must match. The table below shows the setting options for the serial interfaces on device type 703571.

Open: DEVICE MENU > CONFIGURATION > SERIAL INTERFACES > SERIAL INTERFACE 1 TO 2

Configuration item	Selection/settings	Description
Protocol	Modbus master Modbus slave ER8	Communication protocol as master Communication protocol as slave (use only for external relay module)
Baud rate	9600 19200 38400	Transmission speed (symbol rate) of the serial interface
Data format	8 - 1 - no parity 8 - 1 - odd parity 8 - 1 - even parity	Format of the data word Information bit - stop bit - parity
Minimum response time	0 to 500 ms Chapter 4.1 "Modbus timing for serial interface RS422/485", page 37	Minimum time from receipt of a query to transmission of a response This parameter is used to adjust the response speed of the device to slower bus users.
Device address	1 to 254	A bus user's unique ID 0 = Broadcast address ¹ 1 to 247 = Unicast addresses ² 248 to 254 = reserved addresses ³

¹ Device addressing is specified in the Modbus standard. The broadcast address must not be used as a slave address. It is intended for broadcast messages.

² Unicast addresses are intended for use as slave addresses. They are used to give unique IDs to the slave devices so that the master can communicate with them explicitly.

³ In device type 703571, these addresses can be used as slave addresses.

3 Interfaces

Settings for Ethernet interface

A patch/crossover cable with an RJ45 connector is required to use the Ethernet interface. The Ethernet interface can be configured directly on the device or with the aid of the JUMO PC set-up program.

The following protocols can be used for communication over Ethernet:

- Modbus/TCP for communication with Modbus masters over Ethernet
- Communication with the PC setup program via the HTTP protocol
- E-mail sent using the SMTP protocol

DHCP and DNS are also supported. It is possible to obtain the IP configuration automatically via DHCP. If necessary, the IP configuration can also be set up manually.

Open: DEVICE MENU > CONFIGURATION > ETHERNET

Configuration item	Selection/settings	Description
Assign IP address	Manually Automatically	Manually: If no DHCP server is installed on the network and the IP address configuration is known (e.g. by the network administrator), the data are entered manually. Automatically: A DHCP server is installed on the network. After being switched on and booted, device type 703571 receives the IP configuration from the DHCP server; IP configuration takes place automatically.
Manual IP address	Valid IP address ¹	IP address of the device on the network
Subnet mask	Valid subnet mask ¹	The subnet mask specifies which part of the IP address functions as the network address and which part as the host address. It can be used to create subnetworks on a network.
Standard gateway	Valid IP address ¹	IP address of the standard gateway The standard gateway is required for routing into other networks. Without a standard gateway, device type 703571 can only communicate with hosts on the internal network.
DNS server	Valid IP address ¹	IP address of the DNS server DNS = Domain Name System This service permits the DNS name resolution. This is required if device type 703571 is to be launched by entering a URL in the web browser. Without DNS, the IP address must be entered in the web browser.

Configuration item	Selection/settings	Description
Transfer rate	Automatic 10 Mbit/s half duplex 10 Mbit/s full duplex 100 Mbit/s half duplex 100 Mbit/s full duplex	Bit rate and duplex mode of the Ethernet optional board This setting must match the setting of the switch or router port to which device type 703571 is connected.

¹ To enter an IP configuration manually for device type 703571, a valid available IP address on the network must be known. Contact your network administrator to obtain the parameters for manual IP configuration.

**TIP!**

In device type 703571, TCP port 502 is permanently set for Modbus/TCP and cannot be changed.

**TIP!**

With Modbus/TCP, it is not necessary to configure the Modbus device address. Bus users are identified by their IP addresses. The unit ID (Modbus device address in the Modbus/TCP telegram) is permanently set to 255 for device type 703571 (Vergleichen Sie Kapitel 5.1 Modbus/TCP protocol auf Seite 41)

**TIP!**

Each host on a network must have a unique IP address. If an IP address exists several times on the network, an address conflict will result. It must be ensured that fixed IP addresses lie outside the DHCP IP address range and are assigned only once.

**TIP!**

The transmission times in an Ethernet network depend on the network structure and the traffic, among other things. As a result, delays in updating the process values may occur when using Modbus/TCP.

3 Interfaces

4 Modbus via a serial interface

4.1 Modbus timing for serial interface RS422/485

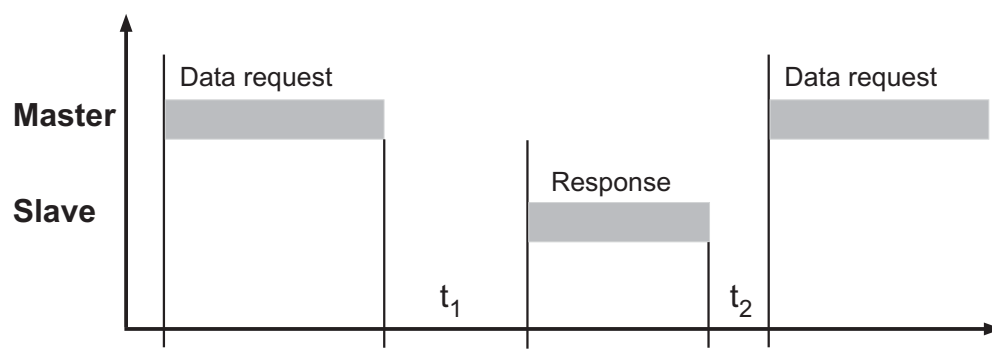
Device type 703571 features a serial interface RS485 on COM 1 as standard.

An RS422/485 can also be retrofitted on expansion slot COM 2.

These interfaces can be used to connect the device to up to 2 bus systems. In each bus, one master can access several slaves that are accessible by means of unique device addresses. This device can assume the role of the master or slave in each bus

Timing of communication

A scanning cycle on a bus proceeds with the following timing:



t_1 Device's internal waiting period prior to checking of the data query and the internal processing time:

min.: 5 ms

typical: 5 to 35 ms

max.: 35 ms or "minimum response time" set

t_2 Waiting period that the master must observe before it starts a new data query with RS485: 35 ms

If the device assumes the role of the master, it observes the time set in the parameter "minimum response time" at the very least at this point.



TIP!

The minimum response time can be set in the configuration.

DEVICE MENU > CONFIGURATION > SERIAL INTERFACE > SERIAL INTERFACE 1 TO 2 This time set is observed at the very least before a response is transmitted (0 to 500 ms). If processing of a query from a master is completed in the slave before the minimum response time has elapsed, the response is not sent until the "minimum response time" has elapsed.

With an RS485 interface, the minimum response time that can be set is needed by the master to switch the interface driver from send to receive.



TIP!

During t_1 and t_2 and during the response time of the slave, no data queries may be generated by the master. Queries made during t_1 and t_2 are ignored by the slave. Queries during the response time invalidate all the data currently on the bus.



TIP!

The terminating code after a data query or data response is 3 characters long. The time needed for these 3 characters depends on the baud rate.

4 Modbus via a serial interface

Character transmission time

The beginning and end of a data block are identified by pauses in transmission. The character transmission time (time to transmit one character) depends on the baud rate and the data format used.

The following results for a data format of 8 data bits, no parity bit, and one stop bit:

$$\text{Character transmission time [ms]} = 1000 \times 9 \text{ bits} / \text{baud rate}$$

For other data formats, the following results:

$$\text{Character transmission time [ms]} = 1000 \times 10 \text{ bits} / \text{baud rate}$$

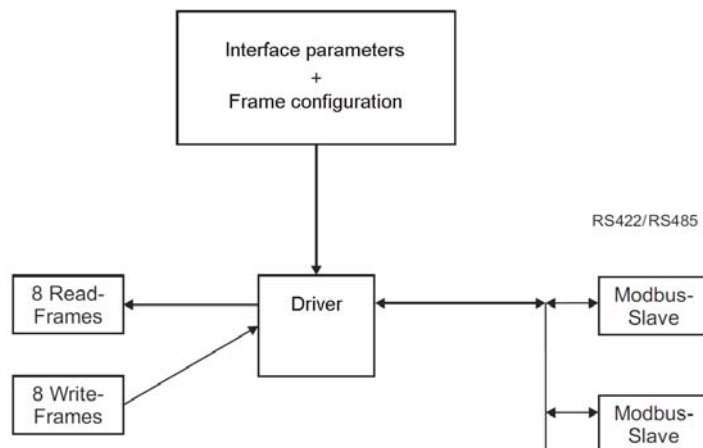
Example

Identifier for end of data query or response for data format 10/9 bits

Waiting period = 3 characters * 1000 * 10 bits/ baud rate

Baud rate [Baud]	Data format [Bit]	Character transmission time [ms]
38400	10	0.260
	9	0.234
19200	10	0.521
	9	0.469
9600	10	1.042
	9	0.938

4.2 Modbus master operation



If device type 703571 has been configured as a master, it can send queries on the bus to slaves. The frames transmitted here are always complete. The corresponding Modbus and device addresses must be specified when configuring the frames in the setup program. For each frame, it is possible to enter the device address and Modbus address used to query the value. Each programmed frame can be disabled by selecting the "Modbus slave" setting for the "interface" parameter.

A maximum of 8 frames can be enabled for reading and a maximum of 8 frames for writing. The enabled frames are cyclically processed in sequence. This also applies to writing frames

4 Modbus via a serial interface

regardless of whether a process value has changed or not. This rules out a transmission that depends on the condition of a change. The complete frame is always transmitted with the frame length configured in the setup program.



TIP!

Double writing of a target variable, e.g. the same variable selector in two read frames, will lead to undefined states and must be avoided.

Timeout

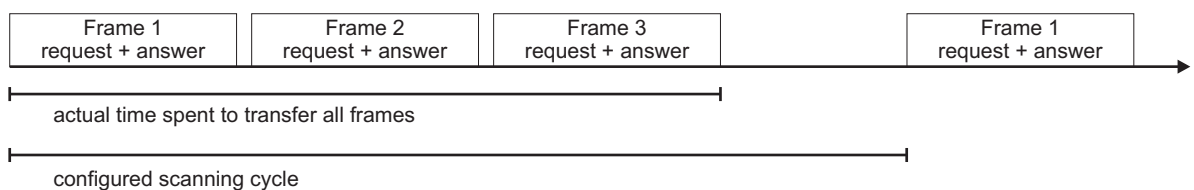
Indicates the maximum timeout used for each sent command to wait for a response before the next command is carried out.

Scanning cycle

The scanning cycle defines the time interval in which the variables are to be read in.

Chronological sequence

The respective interface searches for all frames configured for it (**SETUP NAME > SETUP ONLY > MODBUS FRAMES FOR READING/WRITING > menu item INTERFACE**) and cyclically transmits them in sequence, in the example below to frame 1, frame 2, and frame 3.



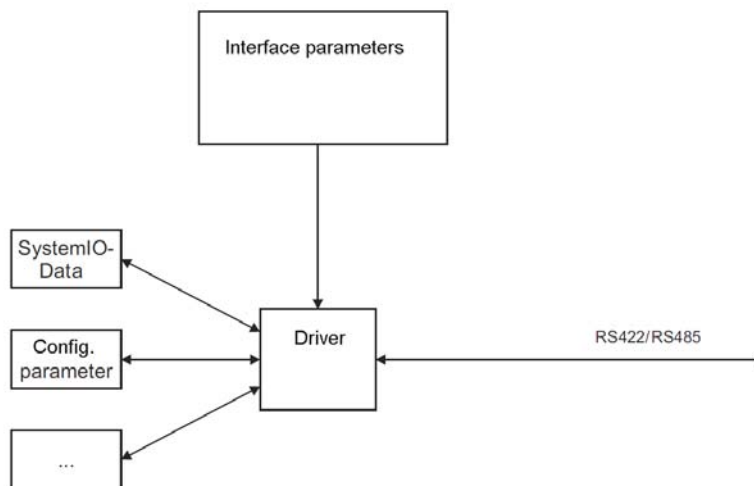
TIP!

The actual time it takes to transmit all frames can be read under the Modbus addresses 0x017E and 0x0187 (serial) and 0x0194, 0x01A7, 0x01BA, and 0x01CD (Modbus/TCP).

If the configured scanning cycle is greater than the elapsed actual time required for scanning, device type 703571 waits to process the next cycle and thus minimizes the bus load. If the configured scanning cycle is too short, all configured frames are completely processed cyclically without a delay.

4 Modbus via a serial interface

4.3 Modbus slave operation



If device type 703571 has been configured as a slave, it responds to Modbus queries from the master on the network. The master controls the data exchange and the slaves only have a response function. They are identified by means of their device addresses.

The master is usually a PC with a setup or visualization program or another Modbus-capable device. The master can query all device variables according to the Modbus address tables of this slave.

⇒ Chapter 7 "Modbus address tables", page 59

5 Modbus and other protocols over Ethernet

5.1 Modbus/TCP protocol

Modbus/TCP uses the Ethernet interface for communicating with the Modbus data. The Modbus telegrams are transmitted via an Ethernet network (IEEE 802.3) using the TCP protocol of the TCP/IP protocol family.

The device type 703571 can assume the role of the slave or master here too.

Structure of a Modbus/TCP telegram

Modbus/TCP is a standardized method for encapsulating a Modbus telegram in a TCP segment and transmitting it via Ethernet.

The Modbus telegram (without CRC) is transmitted with an additional 6-byte or 7-byte "MBAP header" (Modbus Application Header). The seventh byte corresponds to the first serial byte, but has a different designation here.

MBAP header				Modbus telegram
2 bytes Transaction ID	2 bytes Protocol ID	2 bytes Length	1 byte Unit ID	Additional bytes as below, but without CRC
Identical in query and response	Must be 0 for Modbus	Length of query/response in bytes starting with (incl.) "Unit ID"	Corresponds to device address and must be 0xFF or 255 for Modbus/TCP	

For comparison: The "normal" Modbus telegram

Slave address 1 byte	Function code 1 byte	Data field x bytes	CRC16 2 bytes
-------------------------	-------------------------	-----------------------	------------------

Using this protocol, a suitable process data visualization program can read and write system values over a company's own Ethernet network, for example. All device variables in the Modbus address tables can be accessed.

⇒ Chapter 7 "Modbus address tables", page 59

5 Modbus and other protocols over Ethernet

Example: Reading n words

The time of the device is to be read. In this example, the address is 0x11E6 to 0x11E8. See also the Modbus example in Chapter 2.4.1 "Reading n words", page 12.

Query:

MBAP header				Modbus telegram (without slave address and CRC)		
0x0001	0x0000	0x0006	0xFF	0x03	0x11E6	0x0003
2 bytes Transaction ID	2 bytes Protocol ID	2 bytes Length	1 byte Unit ID	1 byte Function code	2 bytes Address of first word	2 bytes Number of words
Assignment of response to query (consecutive numbering)	For Modbus always 0x0000	Length of the query in bytes start- ing with (incl.) "Unit ID"; 6 bytes here (0x06)	For TCP always FF or 255	Function code for "Reading n words"	First word of IP address to be read	3 words are to be read

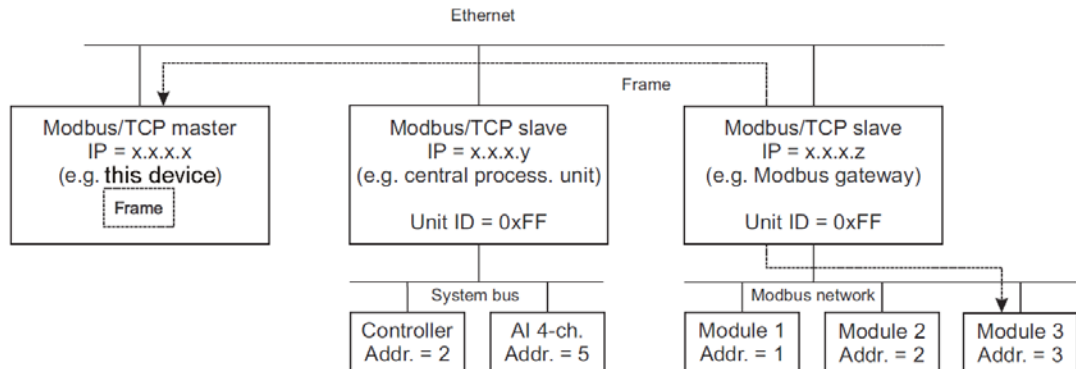
Response:

MBAP header				Modbus telegram (without slave address and CRC)				
0x0001	0x0000	0x0009	0xFF	0x03	0x06	0x000A	0x0000	0x003B
2 bytes Transac- tion ID	2 bytes Protocol ID	2 bytes Length	1 byte Unit ID	1 byte Function code	1 byte Number of bytes read	3 words		
Assign- ment of response to query	For Mod- bus always 0x00	Length of response in bytes (incl.) "Unit ID"; 9 bytes here	For TCP always 0xFF	Function code for "Reading n words"	6 bytes were read	Hour	Minute	Second
						10	00	59

5 Modbus and other protocols over Ethernet

5.2 Networking with Modbus/TCP

The image below provides an overview of the networking options when using the Modbus/TCP protocol:

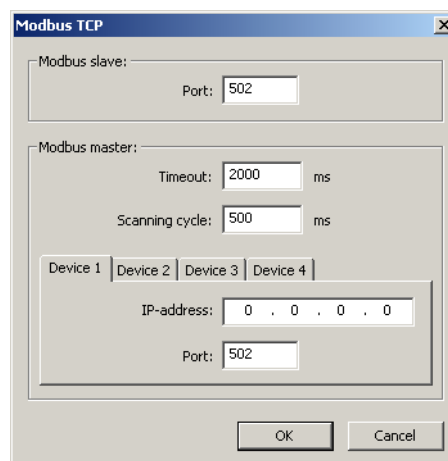


Each Modbus/TCP node is accessible via a unique IP address. The "Unit ID" also contained in the protocol (= device address) must be 0xFF for the node directly attached to the network. However, a node can present itself virtually as several subordinate devices, including a Modbus gateway (on right of image) for all modules behind it. These can then be accessed using Unit ID 1 to 254.

5.3 Modbus master with Modbus/TCP

Up to 4 connections are possible to different Modbus/TCP devices which allow the exchange of a maximum of 8 reading and 8 writing frames. A frame can be used to transmit up to 254 bytes. It operates just like the Modbus master function via the serial interface.

⇒ Chapter 4.2 "Modbus master operation", page 38

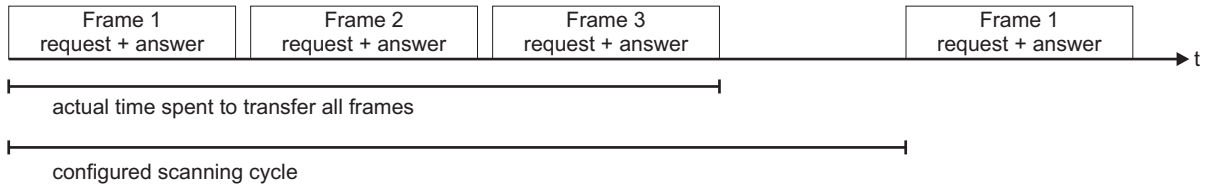


Frames can be exchanged with 4 devices (1 to 4) on the other end using 4 sockets. The Modbus device address that can be edited inside each frame mask and that is also transmitted in the Modbus/TCP protocol can be used to address Modbus gateways and modules behind other base units. The latter then forward the frame to the Modbus node behind them.

5 Modbus and other protocols over Ethernet

Chronological sequence

The respective interface searches for all frames configured for it (setup program: **SETUP ONLY > MODBUS FRAMES FOR READING/WRITING > menu item INTERFACE**) and cyclically transmits them in sequence, in the example below to frame 1, frame 2, and frame 3.



TIP!

The actual time it takes to transmit all frames can be read under the Modbus addresses 0x017E and 0x0187 (serial) and 0x0194, 0x01A7, 0x01BA, and 0x01CD (Modbus/TCP).

If the configured scanning cycle is greater than the elapsed actual time required for scanning, device type 703571 waits to process the next cycle and thus minimizes the bus load. If the configured scanning cycle is too short, all configured frames are completely processed cyclically without a delay.

5.4 Modbus slave with Modbus/TCP

Here, device type 703571 serves as a slave and is available for queries on the bus from the Modbus master. A master can query all device variables according to the Modbus address tables for this slave.

⇒ Chapter 7 "Modbus address tables", page 59



TIP!

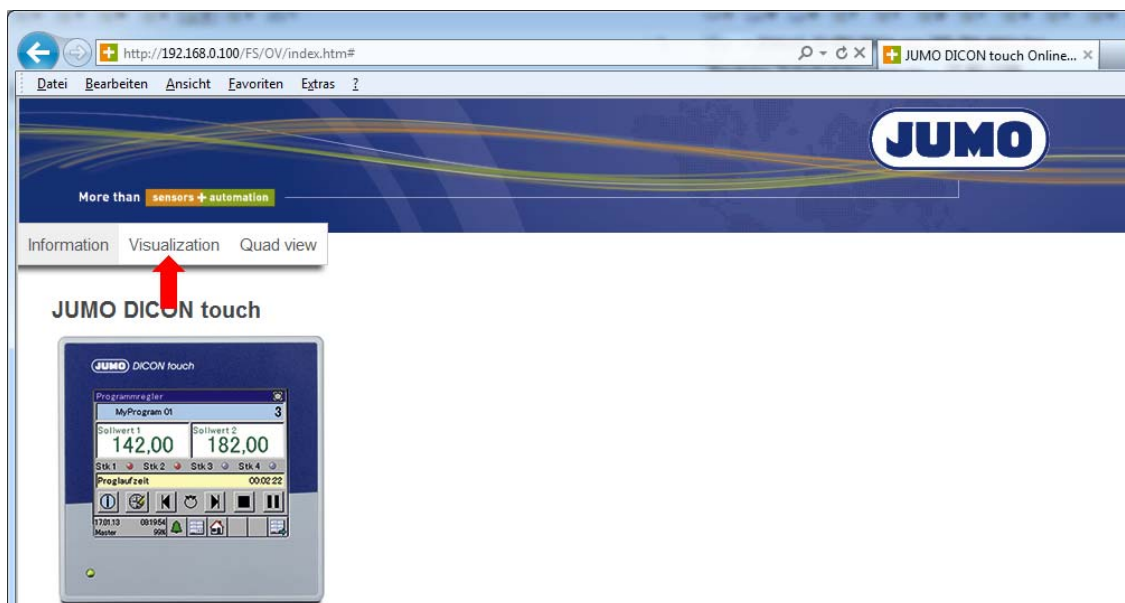
Only two Modbus masters (clients) can access this slave (server) simultaneously via Modbus/TCP. A connection opened by a master is closed by the slave after 30 seconds of inactivity. If a Modbus/TCP port is closed (by a slave or from the other end), it can only be reopened after 10 seconds have elapsed!

The TCP port no. is preset to the value 502_{DEC}. This value can be edited.

5.5 HTTP

Here, device type 703571 is configured as a slave and handles incoming queries as a server via port 80. These queries can come, for example, from a PC with setup program, PC Evaluation Software (PCA), or PCA Communication Software (PCC).

5.6 Browser connection and web server



Device type 703571 can also be accessed by a browser using the HTTP protocol. The required URL for this is the IP address of device type 703571 (in the above example 192.168.0.100). The HTML start page "index.htm" is accessed and can be used to branch to further HTML pages.

The start page "index.htm" and other HTML pages can be loaded on device type 703571 via the setup program **SETUP ONLY > WEB SERVER**.

The Microsoft plug-in "Silverlight®" is required for the browser to use the available default index.htm functions.

The web server supports 8 secure areas which can be assigned 8 users.



TIP!

Information on configuring the web server functionality can be found in operating manual B 703571.0.

5 Modbus and other protocols over Ethernet

5.7 E-mail (SMTP and POP3)

Device type 703571 can send e-mails (e.g. alarms). In this case, it is the master (client) and can access SMTP servers at the standard port (25) as well as POP3 servers at the standard port (110).

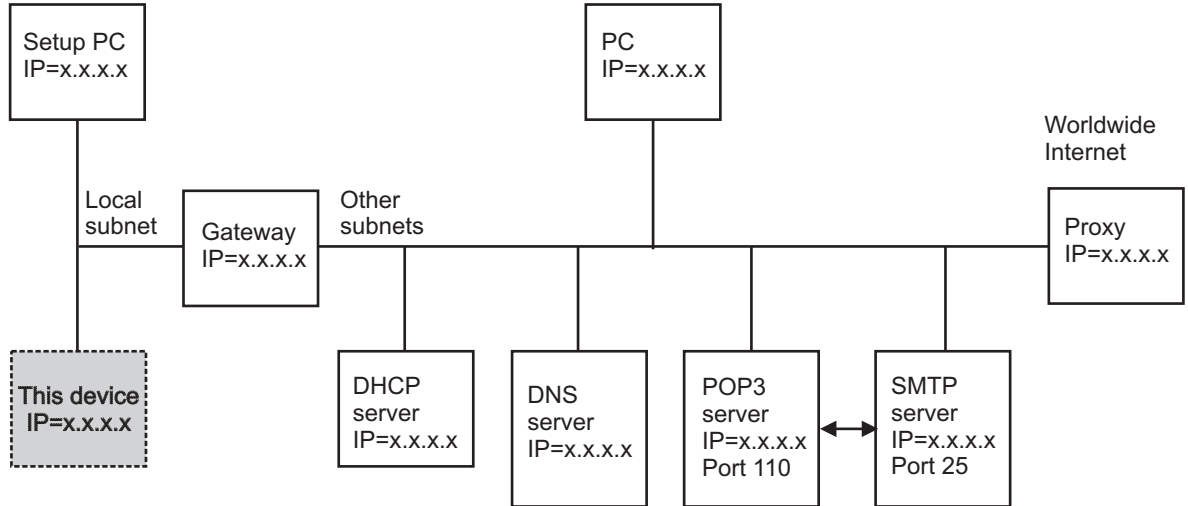


Figure 1: Typical networking in the company network

Function of the individual nodes

Gateway:

Separates local subnets from one another and thus filters the packets. Not all packets are received in every subnet. Packets from outside the local subnet must be addressed to the gateway.

DHCP server:

Can automatically assign an IP address, subnet mask, and gateway address to other nodes when switching on. These parameters can also be entered manually, as a DHCP server is not required.

DNS server:

Changes symbolic names into IP addresses. Example: the query "www.name.de" will generate the response "www.name.de has IP = 0.12.32.45".

POP3 server:

Serves to extract received e-mails from a mail account. The POP3 mail account can be accessed after log-in by entering a user name and password. A successful log-in process often releases the transmission authorization of a connected SMTP server.

SMTP server:

Serves to send e-mails. In many networks, the authorization to send e-mails via a mail account must be released by previously logging in at the corresponding POP3 server.

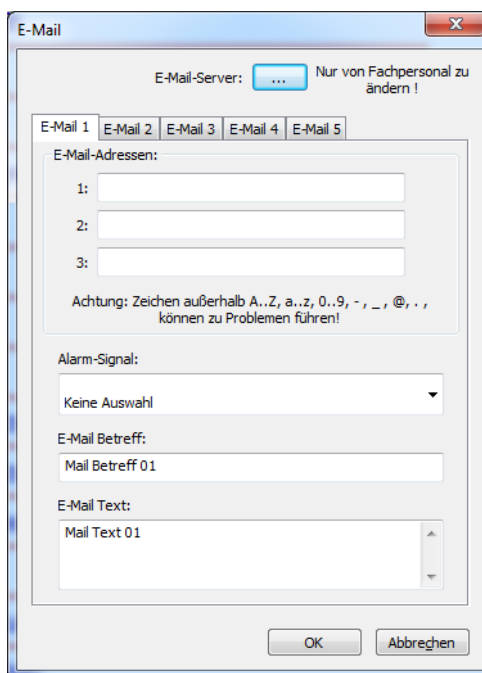
Proxy:

5 Modbus and other protocols over Ethernet

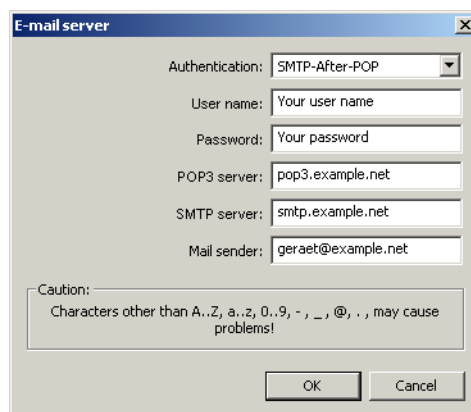
Serves as a gateway between the local company network and the worldwide Internet. It is also used for the conversion of "local" IP addresses (used in the company network) to "one-time" IP addresses (used on the Internet). The device software cannot address a proxy! However, there are also "transparent proxies" that make it possible to address IP addresses worldwide without a special protocol.

Parameters for e-mail and mail server

The displayed parameters can only be edited in the setup program. The relevant screens can be accessed via **SETUP NAME > SETUP ONLY > E-MAIL**.



The screenshot shows the 'E-Mail' configuration window. At the top, it says 'E-Mail-Server: ...' with a note 'Nur von Fachpersonal zu ändern!'. Below this are five tabs labeled 'E-Mail 1' through 'E-Mail 5'. The 'E-Mail 1' tab is selected. Under 'E-Mail-Adressen:', there are three input fields labeled '1:', '2:', and '3:'. A warning message reads: 'Achtung: Zeichen außerhalb A..Z, a..z, 0..9, -, _, @, ., , können zu Problemen führen!'. Below this is an 'Alarm-Signal:' dropdown menu with 'Keine Auswahl' selected. The 'E-Mail Betreff:' field contains 'Mail Betreff 01'. The 'E-Mail Text:' field contains 'Mail Text 01'. At the bottom are 'OK' and 'Abbrechen' buttons.



The screenshot shows the 'E-mail server' configuration window. It has the following fields: 'Authentication:' with a dropdown menu set to 'SMTP-After-POP'; 'User name:' with the text 'Your user name'; 'Password:' with the text 'Your password'; 'POP3 server:' with the text 'pop3.example.net'; 'SMTP server:' with the text 'smtp.example.net'; and 'Mail sender:' with the text 'geraet@example.net'. A 'Caution:' section contains the text: 'Characters other than A..Z, a..z, 0..9, -, _, @, ., , may cause problems!'. At the bottom are 'OK' and 'Cancel' buttons.



TIP!

A mail server that is on the company network (not accessible on the Internet or without proxy addressing) must be entered. This mail server should also be able to transmit e-mails to the Internet.



TIP!

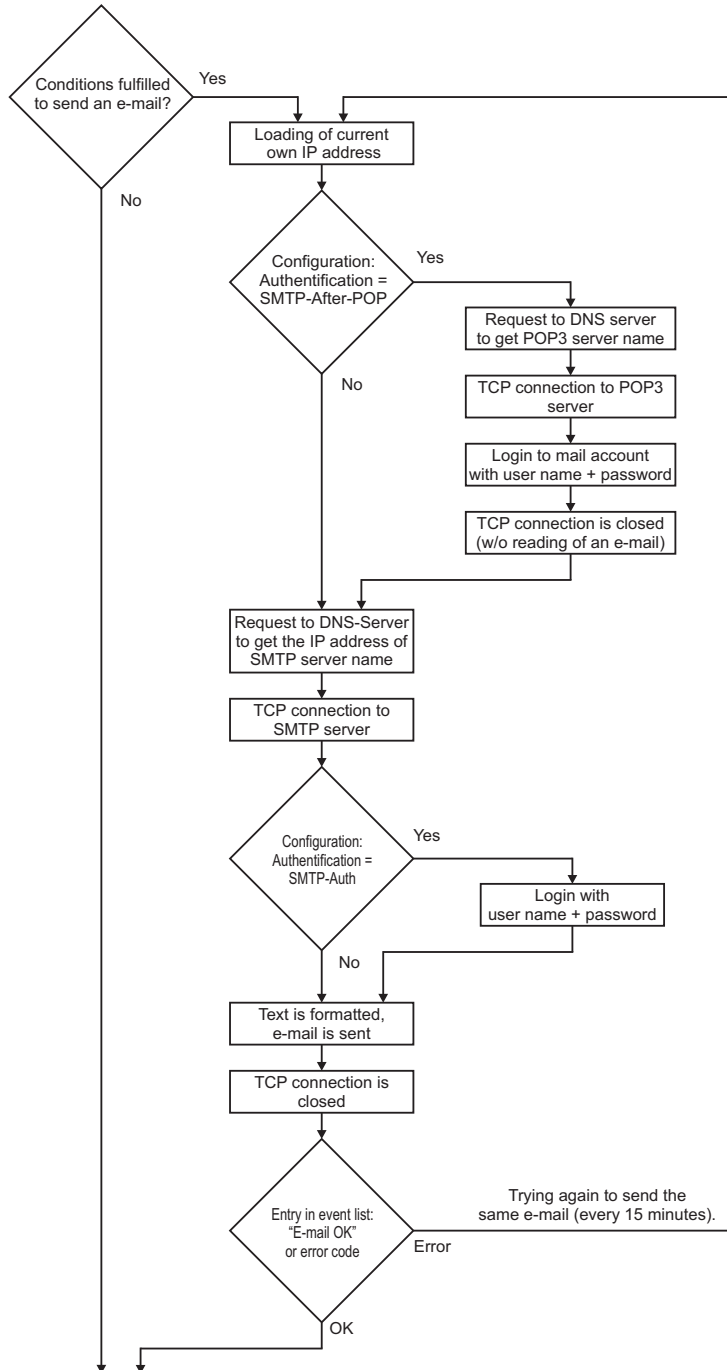
Information on configuring the e-mail functionality can be found in operating manual B 703571.0.

5 Modbus and other protocols over Ethernet

Sending an e-mail via the Internet

Here, several steps depend on configured device parameters. An error code of the event entry (particularly the error codes 120 to 175) can suggest an incorrectly set parameter. An incorrectly entered IP address of the DNS server, for example, generates the error code 153 = "No IP received via DNS".

⇒ Chapter 2.8.3 "Error codes as integer return values", page 24



6.1 General information

Device type 703571 allows the user to individually compile Modbus frames for his application. This achieves a high level of flexibility and the user can thus reduce the data exchanged on the bus to the volume he requires.

This provides a significant advantage with regard to the transmission speed. Numerous variables can be compiled in a large data packet, thus improving the transmission speed, which, in turn, means that the bus is subject to less protocol overhead and fewer changes between the query and response.

The maximum size of a frame is defined as 254 bytes. This corresponds to 127 words for each command. The activated reading or writing frames are cyclically processed in sequence. The frame length specified in the setup program is always used. This length is automatically calculated using the number of entries (64 per frame) in the respective frame. All frames can be used for the Modbus master or Modbus slave.

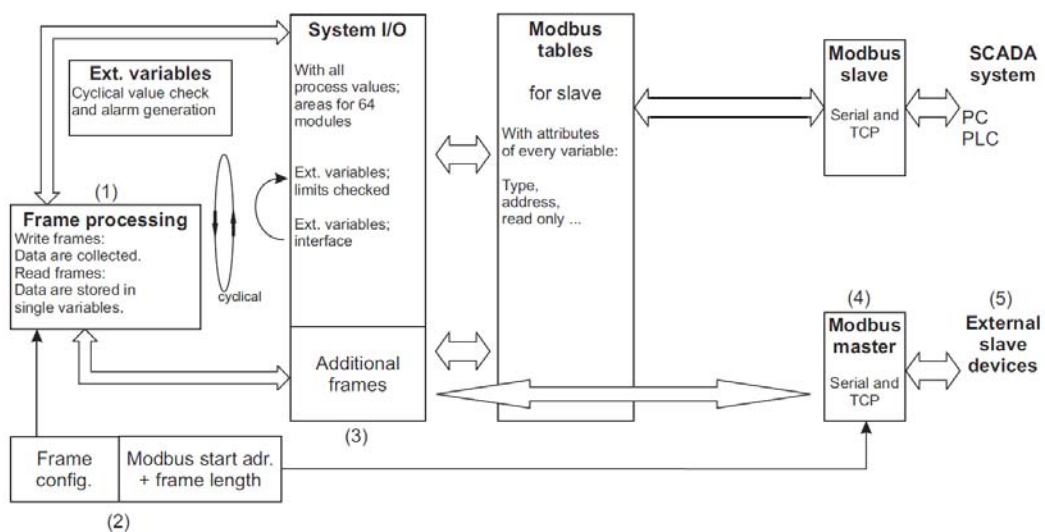


TIP!

There is no locking function against multiple use of the same input variables within several reading frames. The user must prevent this from happening to avoid a data collision.

6.2 Structure of the reading and writing processes

Block diagram of the inputs and outputs for the master function



Reading/writing Modbus (master)

(1)	A cyclical function compiles the individual variables into frames.
(2)	To do this, it accesses the frame arrangement configured in the setup program.
(3)	The complete cyclically updated frames are available.
(4)	Each interface configured as a master transmits (reading or writing) the frames configured for it to the other end.
(5)	The other end receives these frames and responds.

6 User frames



TIP!

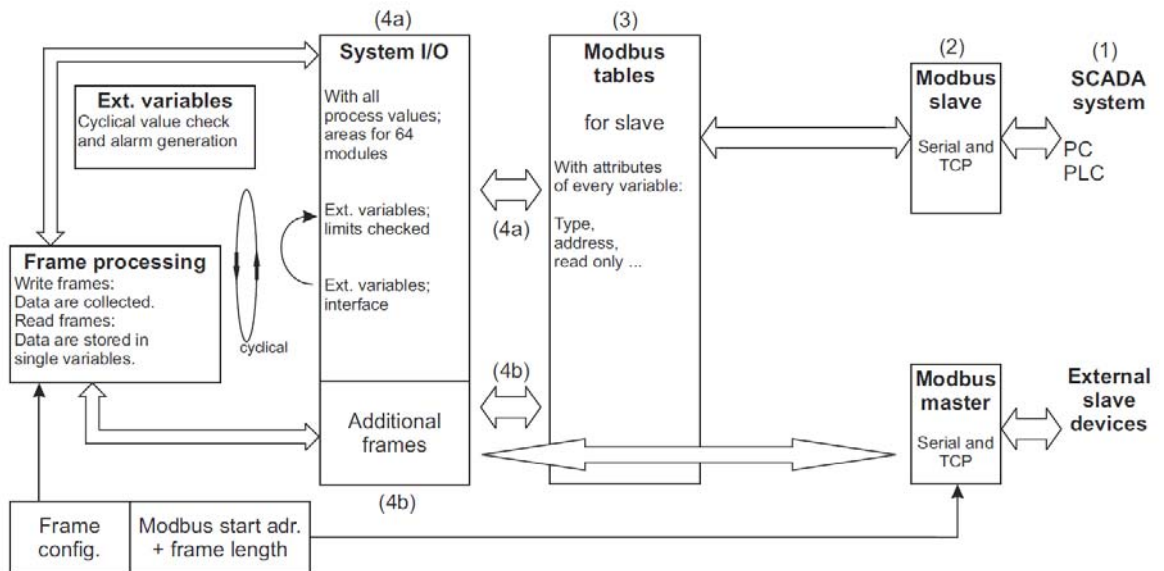
When actively transmitting as a Modbus master, repetitive errors are entered in the event list. Corresponding error codes can be found in Chapter 2.8.3 "Error codes as integer return values", page 24.

Error monitoring

Each frame is monitored for data transmission. If an error occurs, the corresponding error flag is set, an error code is also stored for each frame, and an entry is written in the event list. This entry is not cyclically generated and set; instead, it is generated and set once when the error occurs and then again when the transmission functions again. The following applies to all frames that are read:

if there is no valid response after 3 recurrences, all values of this frame are set to "no input value".

Block diagram of the inputs and outputs for the slave function



Reading/writing Modbus (slave)

(1)	An external master issues a query (reading or writing).
(2)	The interface receives the query.
(3)	The Modbus table assigns the Modbus addresses to the device variables.
(4a)	The variable value is accessed and the Modbus response is generated.
(4b)	The frames configured for the Modbus master can also be accessed for checking or variable scaling/type conversions.

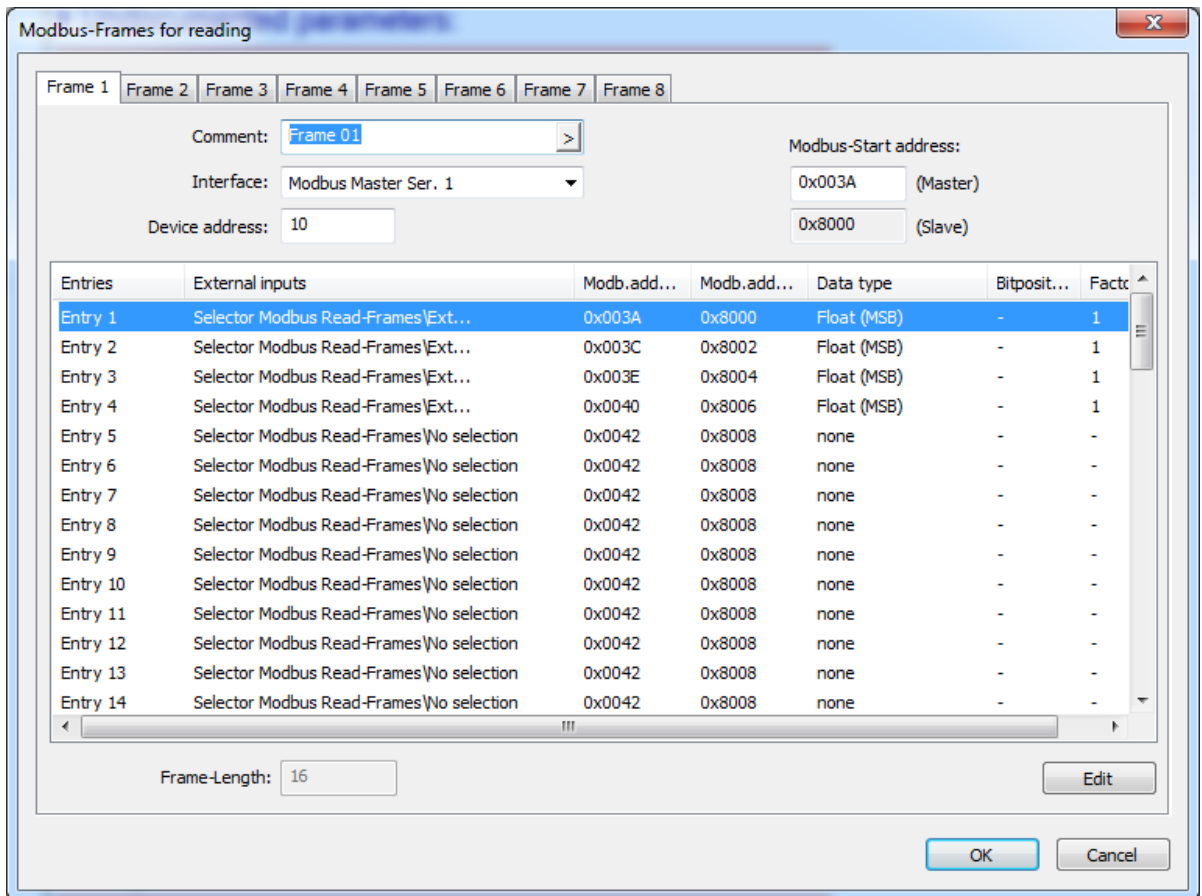
6.3 Compiling Modbus frames

The "Modbus frames for reading" and "Modbus frames for writing" options are found under **SETUP NAME > SETUP ONLY** in the setup program.

6.3.1 Modbus frames for reading

This function is used to compile up to eight Modbus frames for reading process values of external devices (via interface) individually for each opposite end. The process values (analog, integer, and digital values, and text) are written to the selected variables from the received Modbus telegram and are available for use in the system. Each frame can be used to configure up to 64 entries (variables); the process values are then grouped and transmitted in a Modbus telegram.

Setup dialog



Parameters

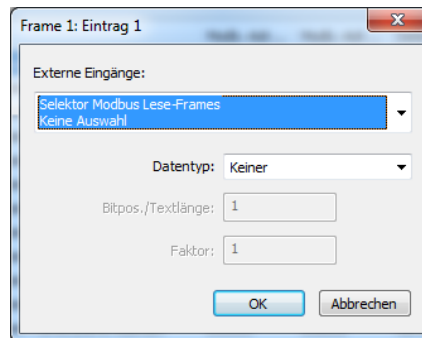
Parameter	Selection/settings	Description
Comment	Frame 1 Use default text or edit text.	Comment on the more detailed description of the frame

6 User frames

Parameter	Selection/settings	Description
Interface		The selection determines whether the frame is actively transmitted as a master or only available for queries as a slave. In the case of a Modbus master, the interface on which the relevant frame is used is also specified. If it is a LAN interface, the external device to be addressed must also be selected.
	Modbus slave	Compiled frame only available for queries as a slave
	Modbus master TCP 1	Modbus master; Modbus TCP via LAN (Ethernet), device 1 addressed
	Modbus master TCP 2	Modbus master; Modbus TCP via LAN (Ethernet), device 2 addressed
	Modbus master TCP 3	Modbus master; Modbus TCP via LAN (Ethernet), device 3 addressed
	Modbus master TCP 4	Modbus master; Modbus TCP via LAN (Ethernet), device 4 addressed
	Modbus master serial 1	Modbus master; Modbus RTU via COM 1
Modbus master serial 2	Modbus master; Modbus RTU via COM 2	
Device address	1 to 255	Device address of the external device (Modbus slave)
Modbus start address (master)	0x0000 to 0xFFFF	Modbus start address (offset) of the external slave device (device type 703571 as a Modbus master) The setup program uses the start address as an initial value for the addresses of the entries. The Modbus description of the external device must be observed for this.
Modbus start address (slave)	0x8800	Displays the Modbus start address (offset) in device type 703571 if this frame is to be queried as a Modbus slave. The start address is used for the setting in the external Modbus master.
Entry 1 to entry 64	Select the desired entry (double-click the line with the entry or mark and then click the "Edit" button).	
Frame length	0 to 254	Displays the frame length (byte(s))

Editing

When the "Edit" button is clicked, this window appears:



Parameters

Parameter	Selection/settings	Description
Variables	Analog, digital, integer, and text variables	
	Inactive Select variable	No variable selected Selector for selecting a variable
Data type	The data type preset depends on the type of the external input in the setup program. It can subsequently be changed.	
	None	No data type selected
	Float (LSB)	Floating-point number; least significant bit (LSB) is transmitted first
	Float (MSB)	Floating-point number; most significant bit (MSB) is transmitted first (This is the standard format for floating-point values with Modbus.)
	Integer (1 byte)	Integer with length of 1 byte
	Integer (2 bytes)	Integer with length of 2 bytes
	Integer (4 bytes)	Integer with length of 4 bytes
	Unsign. int. (1 byte)	Integer without sign, with length of 1 byte
	Unsign. int. (2 bytes)	Integer without sign, with length of 2 bytes
	Unsign. int. (4 bytes)	Integer without sign, with length of 4 bytes
Text (1 char./word) Text (2 char./word)		Text, 1 character per word Text, 2 characters per word
	Integer (1 byte): 0 to 7 Integer (2 bytes): 0 to 15 Integer (4 bytes): 0 to 31 Text (1 char./word): 1 to 127 Text (2 char./word): 1 to 254	The setting ranges of the bit position or text length depend on the data type selected. (This parameter is not active for the "floating-point value" data type.)

6 User frames

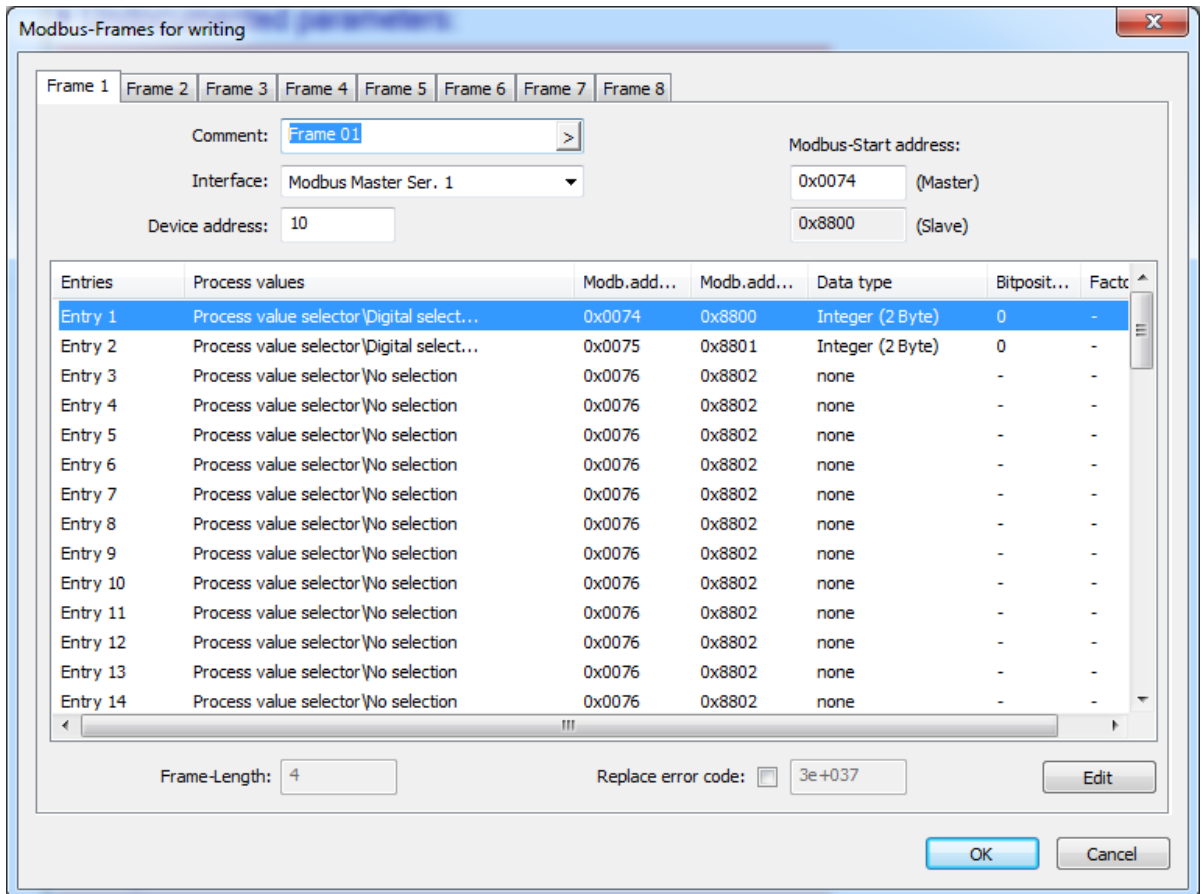
Parameter	Selection/settings	Description
Factor	Using the factor makes it possible to transmit floating-point values in the integer format. The transmitter multiplies the data with the corresponding factor before transmission. The data must be divided by the same value in the receiver.	
	Complete floating-point value range allowed; default value = 1.0	This factor is used to rescale values during the transmission, in particular for simultaneous type conversions. (This parameter is not active for the "text" data type.)

6.3.2 Modbus frames for writing

This function is used to compile up to eight Modbus frames for writing process values to external devices (via interface) individually for each opposite end. The process values (analog, integer, and digital signals, and text) are written to the frames by the system and are available to external devices.

Each frame can be used to configure up to 64 entries (process values), which are then grouped and transmitted in a Modbus telegram.

Setup dialog



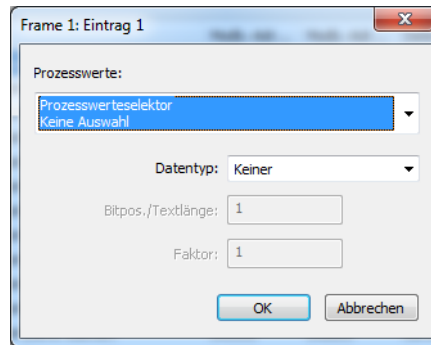
Parameters

Parameter	Selection/settings	Description
Comment	Frame 1 Use default text or edit text.	Comment on the more detailed description of the frame
Interface	The selection determines whether the frame is actively transmitted as a master or only available for queries as a slave. In the case of a Modbus master, the interface on which the relevant frame is used is also specified. If it is a LAN interface, the external device to be addressed must also be selected.	
	Modbus slave	Compiled frame only available for queries as a slave
	Modbus master TCP 1	Modbus master; Modbus TCP via LAN (Ethernet), device 1 addressed
	Modbus master TCP 2	Modbus master; Modbus TCP via LAN (Ethernet), device 2 addressed
	Modbus master TCP 3	Modbus master; Modbus TCP via LAN (Ethernet), device 3 addressed
	Modbus master TCP 4	Modbus master; Modbus TCP via LAN (Ethernet), device 4 addressed
	Modbus master serial 1	Modbus master; Modbus RTU via COM 1
	Modbus master serial 2	Modbus master; Modbus RTU via COM 2
Device address	0 to 1 to 255	Device address of the external device (Modbus slave).
Modbus start address (master)	0x0000 to 0xFFFF	Modbus start address (offset) of the external slave device (device type 703571 as a Modbus master) The setup program uses the start address as an initial value for the addresses of the entries. The Modbus description of the external device must be observed for this.
Modbus start address (slave)	0x8800	Displays the Modbus start address (offset) in device type 703571 if this frame is to be queried as a Modbus slave. The start address is used for the setting in the external Modbus master.
Entry 1 to entry 64	Select the desired entry (double-click the line with the entry or mark and then click the "Edit" button).	
Frame length	0 to 254	Displays the frame length (byte(s))
Error code	-3.37E+38 to +3.0E+37 to +3.37E+38	Value (code) that is transmitted instead of the measured value if a malfunction occurs (for measured values in the floating-point format).

6 User frames

Editing

When the "Edit" button is clicked, this window appears:



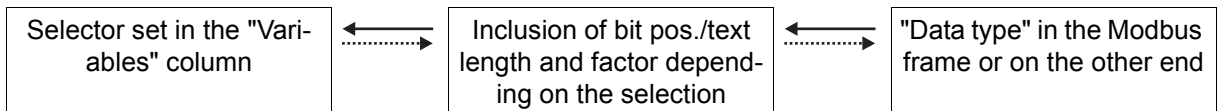
Parameters

Parameter	Selection/settings	Description
Process value selector	Analog, digital, integer, and text selector	
	Inactive Select variable	No variable selected Selector for selecting a variable
Data type	The data type preset depends on the type of the process value in the setup program. It can subsequently be changed.	
	None	No data type selected
	Float (LSB)	Floating-point number; least significant bit (LSB) is transmitted first
	Float (MSB)	Floating point number; most significant bit (MSB) is transmitted first. (This is the standard format for floating-point values with Modbus.)
	Integer (1 byte)	Integer with length of 1 byte
	Integer (2 bytes)	Integer with length of 2 bytes
	Integer (4 bytes)	Integer with length of 4 bytes
	Unsign. int. (1 byte)	Integer without sign, with length of 1 byte
	Unsign. int. (2 bytes)	Integer without sign, with length of 2 bytes
	Unsign. int. (4 bytes)	Integer without sign, with length of 4 bytes
Text (1 char./word)	Text (1 char./word)	Text, 1 character per word
	Text (2 char./word)	Text, 2 characters per word
Bit pos./text length	Integer (1 byte): 0 to 7 Integer (2 bytes): 0 to 15 Integer (4 bytes): 0 to 31 Text (1 char./word): 1 to 127 Text (2 char./word): 1 to 254	The setting ranges of the bit position or text length depend on the data type selected. (This parameter is not active for the "floating-point value" data type.)

Parameter	Selection/settings	Description
Factor	Using the factor makes it possible to transmit floating-point values in the integer format. The transmitter multiplies the data with the corresponding factor before transmission. The data must be divided by the same value in the receiver.	
	Complete floating-point value range allowed; default value = 1.0	This factor is used to rescale values during the transmission, in particular for simultaneous type conversions. (This parameter is not active for the "text" data type.)

6.4 Examples for the data transmission options with frames

There are flexible customization options between the content of the variables in device type 703571 and the data format on the other end. The "Variables" column (see above screenshot) determines the data format within device type 703571; the "Data type" column determines the data format in the Modbus frame or on the other end. They do not have to match. As a result, type conversion can be carried out. The logical direction is displayed in the following diagram:



TIP!

For reading frames, the logical direction is the arrow to the left, and for writing frames the arrow to the right.

The permissible setting options are:

Floating-point value	←→	x factor	←→	Floating-point value
Floating-point value	←→	x factor	←→	Integer value
Integer value	←→	x factor	←→	Floating-point value
Integer value	←→	x factor	←→	Integer value
Binary/Boolean value	←→	Bit position	←→	Integer value ¹
Binary/Boolean value	←→	Bit position	←→	"None" ²
Text	←→	Text length in bytes	←→	Text (1 character/word) Text (2 characters/word)

¹ Only the configured bit is evaluated

² Evaluates additional bits of higher integers

6 User frames

7 Modbus address tables

The tables in this chapter list all process and device data for device type 703571 with their Modbus address, the data type, and the possible access options (Modbus function codes).



CAUTION!

Write operations result in saving to the EEPROM or flash memory. These memory modules only have a limited number of write cycles (approx. 10,000 or 100,000).

Writing on them can therefore result in a memory error in the event of a power failure.

"External analog inputs" are not saved in the EEPROM or flash memory and are suitable for write cycles.

7.1 Data types and access types

Data types

Bit x	Bit no. x (bit 0 is the least significant bit)
Bool	Boolean value (TRUE or FALSE); can be read or written as a word. The value range is 0 to 1.
Byte	1 byte = 8 bits; can be read or written as word. The value range is 0 to 255.
Word	1 word = 2 bytes = 16 bits
Int32	Integer (32 bits) = 2 words
Uint32	Unsigned integer (4 bytes) = 32 bits = 2 words
Long	Long integer (4 bytes) = 32 bits = 2 words
Float	Floating-point value (4 bytes) acc. to IEEE 754 This device saves the floating-point values in the format Float MSB.
Char[60]	Text with 60 characters, with 2 characters in each word
Bit field 32	Bit field 32 bits long

Access types

R/O	Read only
W/O	Write only
R/W	Read write

7 Modbus address tables

7.2 Configuration data and parameters

7.2.1 Device software version

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x000A	0010	Device software version	LONG	R/O

7.2.2 Device name

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1000	4096	Device name The 1st and 2nd bytes of the character string consisting of Unicode characters with UTF-8 coding	Char[60]	R/O
...		
0x101D	4125	The 59th and 60th bytes of the character string consisting of Unicode characters with UTF-8 coding		

7.2.3 Limit values for limit value monitoring 1 to 16

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1071	4209	Limit value for limit value monitoring 1	FLOAT	R/W
0x1073	4211	Limit value for limit value monitoring 2	FLOAT	R/W
0x1075	4213	Limit value for limit value monitoring 3	FLOAT	R/W
0x1077	4215	Limit value for limit value monitoring 4	FLOAT	R/W
0x1079	4217	Limit value for limit value monitoring 5	FLOAT	R/W
0x107B	4219	Limit value for limit value monitoring 6	FLOAT	R/W
0x107D	4221	Limit value for limit value monitoring 7	FLOAT	R/W
0x107F	4223	Limit value for limit value monitoring 8	FLOAT	R/W
0x1081	4225	Limit value for limit value monitoring 9	FLOAT	R/W
0x1083	4227	Limit value for limit value monitoring 10	FLOAT	R/W
0x1085	4229	Limit value for limit value monitoring 11	FLOAT	R/W
0x1087	4231	Limit value for limit value monitoring 12	FLOAT	R/W
0x1089	4233	Limit value for limit value monitoring 13	FLOAT	R/W
0x108B	4235	Limit value for limit value monitoring 14	FLOAT	R/W
0x108D	4237	Limit value for limit value monitoring 15	FLOAT	R/W
0x108F	4239	Limit value for limit value monitoring 16	FLOAT	R/W

7 Modbus address tables

7.2.4 Ramp slope, controllers 1, 2

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1091	4241	Ramp slope for positive gradient, controller 1	FLOAT	R/W
0x1093	4243	Ramp slope for negative gradient, controller 1	FLOAT	R/W
0x1095	4245	Ramp slope for positive gradient, controller 2	FLOAT	R/W
0x1097	4247	Ramp slope for negative gradient, controller 2	FLOAT	R/W

7.2.5 Controller 1, parameter block 1

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1099	4249	XP1 proportional band (P component)	FLOAT	R/W
0x109B	4251	XP2 proportional band (P component)	FLOAT	R/W
0x109D	4253	TV1 derivative time (D component)	FLOAT	R/W
0x109F	4255	TV2 derivative time (D component)	FLOAT	R/W
0x10A1	4257	TN1 reset time (I component)	FLOAT	R/W
0x10A3	4259	TN2 reset time (I component)	FLOAT	R/W
0x10A5	4261	CY1 cycle time	FLOAT	R/W
0x10A7	4263	CY2 cycle time	FLOAT	R/W
0x10A9	4265	XSH switching hysteresis between 1st and 2nd digital output	FLOAT	R/W
0x10AB	4267	XD1 switching differential of the 1st digital output	FLOAT	R/W
0x10AD	4269	XD2 switching differential of the 2nd digital output	FLOAT	R/W
0x10AF	4271	TT actuator time	INT16	R/W
0x10B0	4272	Y0 working point	INT16	R/W
0x10B1	4273	Y1 output value limit	INT16	R/W
0x10B2	4274	Y2 output value limit	INT16	R/W
0x10B3	4275	TK1 minimum start time of the 1st digital output	FLOAT	R/W
0x10B5	4277	TK2 minimum start time of the 2nd digital output	FLOAT	R/W

7.2.6 Controller 1, parameter block 2

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x10B7	4279	XP1 proportional band (P component)	FLOAT	R/W
0x10B9	4281	XP2 proportional band (P component)	FLOAT	R/W

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
0x10BB	4283	TV1 derivative time (D component)	FLOAT	R/W
0x10BD	4385	TV2 derivative time (D component)	FLOAT	R/W
0x10BF	4387	TN1 reset time (I component)	FLOAT	R/W
0x10C1	4289	TN2 reset time (I component)	FLOAT	R/W
0x10C3	4291	CY1 cycle time	FLOAT	R/W
0x10C5	4293	CY2 cycle time	FLOAT	R/W
0x10C7	4295	XSH switching hysteresis between 1st and 2nd digital output	FLOAT	R/W
0x10C9	4297	XD1 switching differential of the 1st digital output	FLOAT	R/W
0x10CB	4299	XD2 switching differential of the 2nd digital output	FLOAT	R/W
0x10CD	4301	TT actuator time	INT16	R/W
0x10CE	4302	Y0 working point	INT16	R/W
0x10CF	4303	Y1 output value limit	INT16	R/W
0x10D0	4304	Y2 output value limit	INT16	R/W
0x10D1	4305	TK1 minimum start time of the 1st digital output	FLOAT	R/W
0x10D3	4307	TK2 minimum start time of the 2nd digital output	FLOAT	R/W

7.2.7 Controller 1, parameter block 3

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x10D5	4309	XP1 proportional band (P component)	FLOAT	R/W
0x10D7	4311	XP2 proportional band (P component)	FLOAT	R/W
0x10D9	4313	TV1 derivative time (D component)	FLOAT	R/W
0x10DB	4315	TV2 derivative time (D component)	FLOAT	R/W
0x10DD	4317	TN1 reset time (I component)	FLOAT	R/W
0x10DF	4319	TN2 reset time (I component)	FLOAT	R/W
0x10E1	4321	CY1 cycle time	FLOAT	R/W
0x10E3	4323	CY2 cycle time	FLOAT	R/W
0x10E5	4325	XSH switching hysteresis between 1st and 2nd digital output	FLOAT	R/W
0x10E7	4327	XD1 switching differential of the 1st digital output	FLOAT	R/W
0x10E9	4329	XD2 switching differential of the 2nd digital output	FLOAT	R/W
0x10EB	4331	TT actuator time	INT16	R/W
0x10EC	4332	Y0 working point	INT16	R/W
0x10ED	4333	Y1 output value limit	INT16	R/W
0x10EE	4334	Y2 output value limit	INT16	R/W
0x10EF	4335	TK1 minimum start time of the 1st digital output	FLOAT	R/W

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x10F1	4337	TK2 minimum start time of the 2nd digital output	FLOAT	R/W

7.2.8 Controller 1, parameter block 4

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x10F3	4339	XP1 proportional band (P component)	FLOAT	R/W
0x10F5	4341	XP2 proportional band (P component)	FLOAT	R/W
0x10F7	4343	TV1 derivative time (D component)	FLOAT	R/W
0x10F9	4345	TV2 derivative time (D component)	FLOAT	R/W
0x10FB	4347	TN1 reset time (I component)	FLOAT	R/W
0x10FD	4349	TN2 reset time (I component)	FLOAT	R/W
0x10FF	4351	CY1 cycle time	FLOAT	R/W
0x1101	4353	CY2 cycle time	FLOAT	R/W
0x1103	4355	XSH switching hysteresis between 1st and 2nd digital output	FLOAT	R/W
0x1105	4357	XD1 switching differential of the 1st digital output	FLOAT	R/W
0x1107	4359	XD2 switching differential of the 2nd digital output	FLOAT	R/W
0x1109	4361	TT actuator time	INT16	R/W
0x110A	4362	Y0 working point	INT16	R/W
0x110B	4363	Y1 output value limit	INT16	R/W
0x110C	4364	Y2 output value limit	INT16	R/W
0x110D	4365	TK1 minimum start time of the 1st digital output	FLOAT	R/W
0x110F	4367	TK2 minimum start time of the 2nd digital output	FLOAT	R/W

7.2.9 Controller 2, parameter block 1

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1111	4369	XP1 proportional band (P component)	FLOAT	R/W
0x1113	4371	XP2 proportional band (P component)	FLOAT	R/W
0x1115	4373	TV1 derivative time (D component)	FLOAT	R/W
0x1117	4375	TV2 derivative time (D component)	FLOAT	R/W
0x1119	4377	TN1 reset time (I component)	FLOAT	R/W
0x111B	4379	TN2 reset time (I component)	FLOAT	R/W
0x111D	4381	CY1 cycle time	FLOAT	R/W

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x111F	4383	CY2 cycle time	FLOAT	R/W
0x1121	4385	XSH switching hysteresis between 1st and 2nd digital output	FLOAT	R/W
0x1123	4387	XD1 switching differential of the 1st digital output	FLOAT	R/W
0x1125	4389	XD2 switching differential of the 2nd digital output	FLOAT	R/W
0x1127	4391	TT actuator time	INT16	R/W
0x1128	4392	Y0 working point	INT16	R/W
0x1129	4393	Y1 output value limit	INT16	R/W
0x112A	4394	Y2 output value limit	INT16	R/W
0x112B	4395	TK1 minimum start time of the 1st digital output	FLOAT	R/W
0x112D	4397	TK2 minimum start time of the 2nd digital output	FLOAT	R/W

7.2.10 Controller 2, parameter block 2

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x112F	4399	XP1 proportional band (P component)	FLOAT	R/W
0x1131	4401	XP2 proportional band (P component)	FLOAT	R/W
0x1133	4403	TV1 derivative time (D component)	FLOAT	R/W
0x1135	4405	TV2 derivative time (D component)	FLOAT	R/W
0x1137	4407	TN1 reset time (I component)	FLOAT	R/W
0x1139	4409	TN2 reset time (I component)	FLOAT	R/W
0x113B	4411	CY1 cycle time	FLOAT	R/W
0x113D	4413	CY2 cycle time	FLOAT	R/W
0x113F	4415	XSH switching hysteresis between 1st and 2nd digital output	FLOAT	R/W
0x1141	4417	XD1 switching differential of the 1st digital output	FLOAT	R/W
0x1143	4419	XD2 switching differential of the 2nd digital output	FLOAT	R/W
0x1145	4421	TT actuator time	INT16	R/W
0x1146	4422	Y0 working point	INT16	R/W
0x1147	4423	Y1 output value limit	INT16	R/W
0x1148	4424	Y2 output value limit	INT16	R/W
0x1149	4425	TK1 minimum start time of the 1st digital output	FLOAT	R/W
0x114B	4427	TK2 minimum start time of the 2nd digital output	FLOAT	R/W

7 Modbus address tables

7.2.11 Controller 2, parameter block 3

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x114D	4429	XP1 proportional band (P component)	FLOAT	R/W
0x114F	4431	XP2 proportional band (P component)	FLOAT	R/W
0x1151	4433	TV1 derivative time (D component)	FLOAT	R/W
0x1153	4435	TV2 derivative time (D component)	FLOAT	R/W
0x1155	4437	TN1 reset time (I component)	FLOAT	R/W
0x1157	4439	TN2 reset time (I component)	FLOAT	R/W
0x1159	4441	CY1 cycle time	FLOAT	R/W
0x115B	4443	CY2 cycle time	FLOAT	R/W
0x115D	4445	XSH switching hysteresis between 1st and 2nd digital output	FLOAT	R/W
0x115F	4447	XD1 switching differential of the 1st digital output	FLOAT	R/W
0x1161	4449	XD2 switching differential of the 2nd digital output	FLOAT	R/W
0x1163	4451	TT actuator time	INT16	R/W
0x1164	4452	Y0 working point	INT16	R/W
0x1165	4453	Y1 output value limit	INT16	R/W
0x1166	4454	Y2 output value limit	INT16	R/W
0x1167	4455	TK1 minimum start time of the 1st digital output	FLOAT	R/W
0x1169	4457	TK2 minimum start time of the 2nd digital output	FLOAT	R/W

7.2.12 Controller 2, parameter block 4

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x116B	4459	XP1 proportional band (P component)	FLOAT	R/W
0x116D	4461	XP2 proportional band (P component)	FLOAT	R/W
0x116F	4463	TV1 derivative time (D component)	FLOAT	R/W
0x1171	4465	TV2 derivative time (D component)	FLOAT	R/W
0x1173	4467	TN1 reset time (I component)	FLOAT	R/W
0x1175	4469	TN2 reset time (I component)	FLOAT	R/W
0x1177	4471	CY1 cycle time	FLOAT	R/W
0x1179	4473	CY2 cycle time	FLOAT	R/W
0x117B	4475	XSH switching hysteresis between 1st and 2nd digital output	FLOAT	R/W
0x117D	4477	XD1 switching differential of the 1st digital output	FLOAT	R/W

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x117F	4479	XD2 switching differential of the 2nd digital output	FLOAT	R/W
0x1181	4481	TT actuator time	INT16	R/W
0x1182	4482	Y0 working point	INT16	R/W
0x1183	4483	Y1 output value limit	INT16	R/W
0x1184	4484	Y2 output value limit	INT16	R/W
0x1185	4485	TK1 minimum start time of the 1st digital output	FLOAT	R/W
0x1187	4487	TK2 minimum start time of the 2nd digital output	FLOAT	R/W

7.2.13 Setpoint values, controller 1

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1189	4489	Setpoint W1 (controller 1)	FLOAT	R/W
0x118B	4491	Setpoint W2 (controller 1)	FLOAT	R/W
0x118D	4493	Setpoint W3 (controller 1)	FLOAT	R/W
0x118F	4495	Setpoint W4 (controller 1)	FLOAT	R/W

7.2.14 Setpoint values, controller 2

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1191	4497	Setpoint W1 (controller 2)	FLOAT	R/W
0x1193	4499	Setpoint W2 (controller 2)	FLOAT	R/W
0x1195	4501	Setpoint W3 (controller 2)	FLOAT	R/W
0x1197	4503	Setpoint W4 (controller 2)	FLOAT	R/W

7.2.15 Filter time constant, offset

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1199	4505	Filter time constant, analog input IN8	FLOAT	R/W
0x119B	4507	Filter time constant, analog input IN9	FLOAT	R/W
0x119D	4509	Filter time constant, analog input IN10	FLOAT	R/W
0x119F	4511	Filter time constant, analog input IN11	FLOAT	R/W
0x11A1	4513	Offset, analog input IN8	FLOAT	R/W
0x11A3	4515	Offset, analog input IN9	FLOAT	R/W

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x11A5	4517	Offset, analog input IN10	FLOAT	R/W
0x11A7	4519	Offset, analog input IN11	FLOAT	R/W

7.2.16 Analog and digital flags 1 to 8

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x11A9	4521	Analog flag 1	FLOAT	R/W
0x11AB	4523	Analog flag 2	FLOAT	R/W
0x11AD	4525	Analog flag 3	FLOAT	R/W
0x11AF	4527	Analog flag 4	FLOAT	R/W
0x11B1	4529	Analog flag 5	FLOAT	R/W
0x11B3	4531	Analog flag 6	FLOAT	R/W
0x11B5	4533	Analog flag 7	FLOAT	R/W
0x11B7	4535	Analog flag 8	FLOAT	R/W
0x11B9	4537	Digital flag 1	BOOL	R/W
0x11BA	4538	Digital flag 2	BOOL	R/W
0x11BB	4539	Digital flag 3	BOOL	R/W
0x11BC	4540	Digital flag 4	BOOL	R/W
0x11BD	4541	Digital flag 5	BOOL	R/W
0x11BE	4542	Digital flag 6	BOOL	R/W
0x11BF	4543	Digital flag 7	BOOL	R/W
0x11C0	4544	Digital flag 8	BOOL	R/W

7.2.17 Controller type, control direction

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x11CB	4555	Controller type, controller 1	INT16	R/W
0x11CC	4556	Control direction, controller 1	INT16	R/W
0x11CD	4557	Controller type, controller 2	INT16	R/W
0x11CE	4558	Control direction, controller 2	INT16	R/W

7 Modbus address tables

7.3 Process values

7.3.1 Transmission times of the frames

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x017E	382	Modbus master, COM 1: actual cycle time in 5 ms ticks	LONG	R/O
0x0187	391	Modbus master, COM 2: actual cycle time in 5 ms ticks	LONG	R/O
0x0194	404	Modbus master, TCP device 1: actual cycle time in 5 ms ticks	LONG	R/O
0x01A7	423	Modbus master, TCP device 2: actual cycle time in 5 ms ticks	LONG	R/O
0x01BA	442	Modbus master, TCP device 3: actual cycle time in 5 ms ticks	LONG	R/O
0x01CD	461	Modbus master, TCP device 4: actual cycle time in 5 ms ticks	LONG	R/O

7.3.2 Date, time

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x11E3	4579	Time : Year (2013)	INT16	R/O
0x11E4	4580	Time : Month	INT16	R/O
0x11E5	4581	Time : Day	INT16	R/O
0x11E6	4582	Time : Hour	INT16	R/O
0x11E7	4583	Time : Minute	INT16	R/O
0x11E8	4584	Time : Second	INT16	R/O

7.3.3 Measuring inputs 1 to 4

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x11E9	4585	Universal input IN8, measured value	FLOAT	R/O
0x11F3	4595	Universal input IN9, measured value	FLOAT	R/O
0x11FD	4605	Universal input IN10, measured value	FLOAT	R/O
0x1207	4615	Universal input IN11, measured value	FLOAT	R/O

7 Modbus address tables

7.3.4 Controller 1

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x122D	4653	Actual value, controller 1	FLOAT	R/O
0x122F	4655	Setpoint value, controller 1	FLOAT	R/W
0x1231	4657	Output level, controller 1	FLOAT	R/O
0x1233	4659	Output level output 1, controller 1	FLOAT	R/O
0x1235	4661	Output level output 2, controller 1	FLOAT	R/O
0x1237	4663	1st output, controller 1	INT16	R/O
0x1238	4664	2nd output, controller 1	INT16	R/O
0x1239	4665	Controller cycle alarm, controller 1	INT16	R/O
0x123A	4666	Output level alarm, controller 1	INT16	R/O
0x123B	4667	Manual mode, controller 1	INT16	R/O
0x123C	4668	Tune, controller 1	INT16	R/O

7.3.5 Controller 2

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x123D	4669	Actual value, controller 2	FLOAT	R/O
0x123F	4671	Setpoint value, controller 2	FLOAT	R/W
0x1241	4673	Output level, controller 2	FLOAT	R/O
0x1243	4675	Output level output 1, controller 2	FLOAT	R/O
0x1245	4677	Output level output 2, controller 2	FLOAT	R/O
0x1247	4679	1st output, controller 2	INT16	R/O
0x1248	4680	2nd output, controller 2	INT16	R/O
0x1249	4681	Controller cycle alarm, controller 2	INT16	R/O
0x124A	4682	Output level alarm, controller 2	INT16	R/O
0x124B	4683	Manual mode, controller 2	INT16	R/O
0x124C	4684	Tune, controller 2	INT16	R/O

7.3.6 Setpoint values, controller 1

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x124D	4685	Setpoint WR (ramp value)	FLOAT	R/W
0x124F	4687	Setpoint WE (ramp end value)	FLOAT	R/O
0x1251	4689	Setpoint W1	FLOAT	R/O
0x1253	4691	Setpoint W2	FLOAT	R/O

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1255	4693	Setpoint W3	FLOAT	R/O
0x1257	4695	Setpoint W4	FLOAT	R/O
0x1259	4697	Setpoint value, interface	FLOAT	R/W
0x125B	4699	Setpoint limit WA (start)	FLOAT	R/O
0x125D	4701	Setpoint limit WE (end)	FLOAT	R/O

7.3.7 Setpoint values, controller 2

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x125F	4703	Setpoint WR (ramp value)	FLOAT	R/W
0x1261	4705	Setpoint WE (ramp end value)	FLOAT	R/O
0x1263	4707	Setpoint W1	FLOAT	R/O
0x1265	4709	Setpoint W2	FLOAT	R/O
0x1267	4711	Setpoint W3	FLOAT	R/O
0x1269	4713	Setpoint W4	FLOAT	R/O
0x126B	4715	Setpoint value, interface	FLOAT	R/W
0x126D	4717	Setpoint limit WA (start)	FLOAT	R/O
0x126F	4719	Setpoint limit WE (end)	FLOAT	R/O

7.3.8 Program controller

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1271	4721	Present program number	INT16	R/O
0x1272	4722	Present section number	INT16	R/O
0x1273	4723	Setpoint W1	FLOAT	R/O
0x1275	4725	Setpoint W2	FLOAT	R/O
0x1277	4727	Operating contacts	INT16	R/O
0x1278	4728	Parameter block number	INT16	R/O
0x1279	4729	Remaining section time	LONG	R/O
0x127B	4731	Section run time	LONG	R/O
0x127D	4733	Remaining program time	LONG	R/O
0x127F	4735	Program runtime	LONG	R/O
0x1281	4737	Section end value W1	FLOAT	R/O
0x1283	4739	Section end value W2	FLOAT	R/O
0x1285	4741	Tolerance band MIN	FLOAT	R/O

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1287	4743	Tolerance band MAX	FLOAT	R/O
0x1289	4745	Program generator mode	INT16	R/O
		0=Basic status 1=Manual operating mode 2=Prerun 3=Program start 4=Automatic operation 6=Program end		R/O
0x128A	4746	Program generator automatic signal	INT16	R/O
0x128B	4747	Program generator hold signal	INT16	R/O
0x128C	4748	Program generator tolerance band signal	INT16	R/O
0x128D	4749	Program generator program end signal	INT16	R/O
0x128E	4750	Program generator restart error	INT16	R/O

7 Modbus address tables

7.3.9 Math, logic

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x128F	4751	Math 1	FLOAT	R/O
0x1291	4753	Math 2	FLOAT	R/O
0x1293	4755	Math 3	FLOAT	R/O
0x1295	4757	Math 4	FLOAT	R/O
0x1297	4759	Math 5	FLOAT	R/O
0x1299	4761	Math 6	FLOAT	R/O
0x129B	4763	Math 7	FLOAT	R/O
0x129D	4765	Math 8	FLOAT	R/O
0x129F	4767	Math 1, min. alarm	BOOL	R/O
0x12A0	4768	Math 2, min. alarm	BOOL	R/O
0x12A1	4769	Math 3, min. alarm	BOOL	R/O
0x12A2	4770	Math 4, min. alarm	BOOL	R/O
0x12A3	4771	Math 5, min. alarm	BOOL	R/O
0x12A4	4772	Math 6, min. alarm	BOOL	R/O
0x12A5	4773	Math 7, min. alarm	BOOL	R/O
0x12A6	4774	Math 8, min. alarm	BOOL	R/O
0x12A7	4775	Math 1, max. alarm	BOOL	R/O
0x12A8	4776	Math 2, max. alarm	BOOL	R/O
0x12A9	4777	Math 3, max. alarm	BOOL	R/O
0x12AA	4778	Math 4, max. alarm	BOOL	R/O
0x12AB	4779	Math 5, max. alarm	BOOL	R/O
0x12AC	4780	Math 6, max. alarm	BOOL	R/O
0x12AD	4781	Math 7, max. alarm	BOOL	R/O
0x12AE	4782	Math 8, max. alarm	BOOL	R/O
0x12AF	4783	Logic signal 1	BOOL	R/O
0x12B0	4784	Logic signal 2	BOOL	R/O
0x12B1	4785	Logic signal 3	BOOL	R/O
0x12B2	4786	Logic signal 4	BOOL	R/O
0x12B3	4787	Logic signal 5	BOOL	R/O
0x12B4	4788	Logic signal 6	BOOL	R/O
0x12B5	4789	Logic signal 7	BOOL	R/O
0x12B6	4790	Logic signal 8	BOOL	R/O
0x12B7	4791	Logic alarm 1	BOOL	R/O
0x12B8	4792	Logic alarm 2	BOOL	R/O
0x12B9	4793	Logic alarm 3	BOOL	R/O

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x12BA	4794	Logic alarm 4	BOOL	R/O
0x12BB	4795	Logic alarm 5	BOOL	R/O
0x12BC	4796	Logic alarm 6	BOOL	R/O
0x12BD	4797	Logic alarm 7	BOOL	R/O
0x12BE	4798	Logic alarm 8	BOOL	R/O

7.3.10 Digital inputs 1 to 7 and their alarms

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x12BF	4799	Digital input 1	BOOL	R/O
0x12C0	4800	Digital input 2	BOOL	R/O
0x12C1	4801	Digital input 3	BOOL	R/O
0x12C2	4802	Digital input 4	BOOL	R/O
0x12C3	4803	Digital input 5	BOOL	R/O
0x12C4	4804	Digital input 6	BOOL	R/O
0x12C5	4805	Digital input 7	BOOL	R/O
0x12C6	4806	Reserved	BOOL	R/O
0x12C7	4807	Digital input 1, alarm	INT16	R/O
0x12C8	4808	Digital input 2, alarm	INT16	R/O
0x12C9	4809	Digital input 3, alarm	INT16	R/O
0x12CA	4810	Digital input 4, alarm	INT16	R/O
0x12CB	4811	Digital input 5, alarm	INT16	R/O
0x12CC	4812	Digital input 6, alarm	INT16	R/O
0x12CD	4813	Digital input 7, alarm	INT16	R/O
0x12CE	4814	Reserved	INT16	R/O

7.3.11 Limit value monitoring, actual values and alarms

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x12CF	4815	Limit value output 1	BOOL	R/O
0x12D0	4816	Limit value output 2	BOOL	R/O
0x12D1	4817	Limit value output 3	BOOL	R/O
0x12D2	4818	Limit value output 4	BOOL	R/O
0x12D3	4819	Limit value output 5	BOOL	R/O
0x12D4	4820	Limit value output 6	BOOL	R/O

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x12D5	4821	Limit value output 7	BOOL	R/O
0x12D6	4822	Limit value output 8	BOOL	R/O
0x12D7	4823	Limit value output 9	BOOL	R/O
0x12D8	4824	Limit value output 10	BOOL	R/O
0x12D9	4825	Limit value output 11	BOOL	R/O
0x12DA	4826	Limit value output 12	BOOL	R/O
0x12DB	4827	Limit value output 13	BOOL	R/O
0x12DC	4828	Limit value output 14	BOOL	R/O
0x12DD	4829	Limit value output 15	BOOL	R/O
0x12DE	4830	Limit value output 16	BOOL	R/O
0x12DF	4831	Limit value monitoring 1, alarm signal	BOOL	R/O
0x12E0	4832	Limit value monitoring 2, alarm signal	BOOL	R/O
0x12E1	4833	Limit value monitoring 3, alarm signal	BOOL	R/O
0x12E2	4834	Limit value monitoring 4, alarm signal	BOOL	R/O
0x12E3	4835	Limit value monitoring 5, alarm signal	BOOL	R/O
0x12E4	4836	Limit value monitoring 6, alarm signal	BOOL	R/O
0x12E5	4837	Limit value monitoring 7, alarm signal	BOOL	R/O
0x12E6	4838	Limit value monitoring 8, alarm signal	BOOL	R/O
0x12E7	4839	Limit value monitoring 9, alarm signal	BOOL	R/O
0x12E8	4840	Limit value monitoring 10, alarm signal	BOOL	R/O
0x12E9	4841	Limit value monitoring 11, alarm signal	BOOL	R/O
0x12EA	4842	Limit value monitoring 12, alarm signal	BOOL	R/O
0x12EB	4843	Limit value monitoring 13, alarm signal	BOOL	R/O
0x12EC	4844	Limit value monitoring 14, alarm signal	BOOL	R/O
0x12ED	4845	Limit value monitoring 15, alarm signal	BOOL	R/O
0x12EE	4846	Limit value monitoring 16, alarm signal	BOOL	R/O

7.3.12 Digital control signals

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x12EF	4847	Digital control signal 1	BOOL	R/W
0x12F0	4848	Digital control signal 2	BOOL	R/W
0x12F1	4849	Digital control signal 3	BOOL	R/W
0x12F2	4850	Digital control signal 4	BOOL	R/W
0x12F3	4851	Digital control signal 5	BOOL	R/W
0x12F4	4852	Digital control signal 6	BOOL	R/W

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x12F5	4853	Digital control signal 7	BOOL	R/W
0x12F6	4854	Digital control signal 8	BOOL	R/W
0x12F7	4855	Digital control signal 1, alarm	BOOL	R/O
0x12F8	4856	Digital control signal 2, alarm	BOOL	R/O
0x12F9	4857	Digital control signal 3, alarm	BOOL	R/O
0x12FA	4858	Digital control signal 4, alarm	BOOL	R/O
0x12FB	4859	Digital control signal 5, alarm	BOOL	R/O
0x12FC	4860	Digital control signal 6, alarm	BOOL	R/O
0x12FD	4861	Digital control signal 7, alarm	BOOL	R/O
0x12FE	4862	Digital control signal 8, alarm	BOOL	R/O

7.3.13 Timers

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1307	4871	Timer output 1	BOOL	R/O
0x1308	4872	Timer output 2	BOOL	R/O
0x1309	4873	Timer status 1	BOOL	R/O
0x130A	4874	Timer status 2	BOOL	R/O
0x130B	4875	Timer end signal 1	BOOL	R/O
0x130C	4876	Timer end signal 2	BOOL	R/O
0x130D	4877	Timer, tolerance band signal 1	BOOL	R/O
0x130E	4878	Timer, tolerance band signal 2	BOOL	R/O
0x130F	4879	Timer hold 1	BOOL	R/O
0x1310	4880	Timer hold 2	BOOL	R/O
0x1311	4881	Timer prerun 1	BOOL	R/O
0x1312	4882	Timer prerun 2	BOOL	R/O
0x1313	4883	Timer runtime 1	LONG	R/O
0x1315	4885	Timer runtime 2	LONG	R/O
0x1317	4887	Timer remaining running time 1	LONG	R/O
0x1319	4889	Time remaining running time 2	LONG	R/O
0x131B	4891	Timer lead time 1	LONG	R/O
0x131D	4893	Timer lead time 2	LONG	R/O
0x131F	4895	Timer after-run time 1	LONG	R/O
0x1321	4897	Timer after-run time 2	LONG	R/O
0x1323	4899	Timer end time 1	LONG	R/O
0x1325	4901	Timer end time 2	LONG	R/O

7 Modbus address tables

7.3.14 Analog and digital flag

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1327	4903	Analog flag 1	FLOAT	R/O
0x1329	4905	Analog flag 2	FLOAT	R/O
0x132B	4907	Analog flag 3	FLOAT	R/O
0x132D	4909	Analog flag 4	FLOAT	R/O
0x132F	4911	Analog flag 5	FLOAT	R/O
0x1331	4913	Analog flag 6	FLOAT	R/O
0x1333	4915	Analog flag 7	FLOAT	R/O
0x1335	4917	Analog flag 8	FLOAT	R/O
0x1337	4919	Digital flag 1	BOOL	R/O
0x1338	4920	Digital flag 2	BOOL	R/O
0x1339	4921	Digital flag 3	BOOL	R/O
0x133A	4922	Digital flag 4	BOOL	R/O
0x133B	4923	Digital flag 5	BOOL	R/O
0x133C	4924	Digital flag 6	BOOL	R/O
0x133D	4925	Digital flag 7	BOOL	R/O
0x133E	4926	Digital flag 8	BOOL	R/O

7.3.15 Device operation

Modbus address		Signal designation	Data type	
Hex.	Dec.			
0x134B	4939	Operation, tune start, controller 1	BOOL	R/W
0x134C	4940	Operation, tune stop, controller 1	BOOL	R/W
0x134D	4941	Operation, tune start, controller 2	BOOL	R/W
0x134E	4942	Operation, tune stop, controller 2	BOOL	R/W
0x134F	4943	Operation, manual mode, controller 1	BOOL	R/W
0x1350	4944	Operation, automatic, controller 1	BOOL	R/W
0x1351	4945	Operation, manual mode, controller 2	BOOL	R/W
0x1352	4946	Operation, automatic, controller 2	BOOL	R/W
0x1353	4947	Operation, manual output level, controller 1	FLOAT	R/W
0x1355	4949	Operation, manual output level, controller 2	FLOAT	R/W
0x1357	4951	Operation, actuator drive on controller 1	BOOL	R/W
0x1358	4952	Operation, actuator drive for controller 1	BOOL	R/W
0x1359	4953	Operation, actuator drive on controller 2	BOOL	R/W
0x135A	4954	Operation, actuator drive for controller 2	BOOL	R/W

7 Modbus address tables

Modbus address		Signal designation	Data type	
Hex.	Dec.			
0x135B	4955	Operation, program start	BOOL	R/W
0x135C	4956	Operation, program stop	BOOL	R/W
0x135D	4957	Operation, program break	BOOL	R/W
0x135E	4958	Operation, manual program	BOOL	R/W
0x135F	4959	Operation, fast forward	BOOL	R/W
0x1360	4960	Operation, next section	BOOL	R/W
0x1361	4961	Operation, previous section	BOOL	R/W
0x1362	4962	Operation, command buffer	INT16	R/W
0x1363	4963	Operation, program start, program number	INT16	R/W
0x1364	4964	Operation, program start, start section	INT16	R/W
0x1365	4965	Operation, program start, lead time	LONG	R/W
0x1367	4967	Operation, program start, section remaining time	LONG	R/W
0x1369	4969	Operation, program start, start time	LONG	R/W
0x136B	4971	Operation, program start, week day	INT16	R/W
0x136E	4974	Manual operating mode, setpoint W1	FLOAT	R/W
0x1370	4976	Manual operating mode, setpoint W2	FLOAT	R/W
0x1372	4978	Manual operating mode, operating contacts	INT16	R/W
0x1373	4979	Manual operating mode, parameter block number	INT16	R/W
0x1374	4980	Operation, limit value monitoring acknowledgement	BOOL	R/W
0x1375	4981	Operation, service acknowledgement	BOOL	R/W
0x1376	4982	Operation, timer start T1	BOOL	R/W
0x1377	4983	Operation, timer stop T1	BOOL	R/W
0x1378	4984	Operation, timer hold T1	BOOL	R/W
0x1379	4985	Operation, timer acknowledgement T1	BOOL	R/W
0x137A	4986	Operation, timer restart T1	BOOL	R/W
0x137B	4987	Operation, timer start T2	BOOL	R/W
0x137C	4988	Operation, timer stop T2	BOOL	R/W
0x137D	4989	Operation, timer hold T2	BOOL	R/W
0x137E	4990	Operation, timer acknowledgement T2	BOOL	R/W
0x137F	4991	Operation, timer restart T2	BOOL	R/W
0x1380	4992	Operation, hold, ramp 1	BOOL	R/W
0x1381	4993	Operation, stop, ramp 1	BOOL	R/W
0x1382	4994	Operation, restart, ramp 1	BOOL	R/W
0x1383	4995	Operation, hold, ramp 2	BOOL	R/W
0x1384	4996	Operation, stop, ramp 2	BOOL	R/W
0x1385	4997	Operation, restart, ramp 2	BOOL	R/W
0x1386	4998	Operation, function key 1	BOOL	R/W

7 Modbus address tables

Modbus address		Signal designation	Data type	
Hex.	Dec.			
0x1387	4999	Operation, function key 2	BOOL	R/W

7.3.16 Outputs

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x13B0	5040	Switching output 1	INT16	R/O
0x13B1	5041	Switching output 2	INT16	R/O
0x13B2	5042	Switching output 3	INT16	R/O
0x13B3	5043	Switching output 4	INT16	R/O
0x13B4	5044	Switching output 5	INT16	R/O
0x13B5	5045	Switching output 6	INT16	R/O
0x13B6	5046	Switching output 7	INT16	R/O
0x13B7	5047	Switching output 8	INT16	R/O
0x13B8	5048	Switching output 9	INT16	R/O
0x13B9	5049	Switching output 10	INT16	R/O
0x13BA	5050	Switching output 11	INT16	R/O
0x13BB	5051	Switching output 12	INT16	R/O
0x13BC	5052	ER8(1) switching output 1 to 8	INT16	R/O
0x13BD	5053	ER8(2) switching output 1 to 8	INT16	R/O

7.3.17 External analog and digital signals

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x13BE	5054	External analog input 1, (via interface)	FLOAT	R/W
0x13C0	5056	External analog input 2, (via interface)	FLOAT	R/W
0x13C2	5058	External analog input 3, (via interface)	FLOAT	R/W
0x13C4	5060	External analog input 4, (via interface)	FLOAT	R/W
0x13C6	5062	External analog input 5, (via interface)	FLOAT	R/W
0x13C8	5064	External analog input 6, (via interface)	FLOAT	R/W
0x13CA	5066	External analog input 7, (via interface)	FLOAT	R/W
0x13CC	5068	External analog input 8, (via interface)	FLOAT	R/W
0x13CE	5070	External analog input 1	FLOAT	R/O
0x13D0	5072	External analog input 2	FLOAT	R/O
0x13D2	5074	External analog input 3	FLOAT	R/O
0x13D4	5076	External analog input 4	FLOAT	R/O

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x13D6	5078	External analog input 5	FLOAT	R/O
0x13D8	5080	External analog input 6	FLOAT	R/O
0x13DA	5082	External analog input 7	FLOAT	R/O
0x13DC	5084	External analog input 8	FLOAT	R/O
0x13DE	5086	External analog alarm, min. 1	BOOL	R/O
0x13DF	5087	External analog alarm, min. 2	BOOL	R/O
0x13E0	5088	External analog alarm, min. 3	BOOL	R/O
0x13E1	5089	External analog alarm, min. 4	BOOL	R/O
0x13E2	5090	External analog alarm, min. 5	BOOL	R/O
0x13E3	5091	External analog alarm, min. 6	BOOL	R/O
0x13E4	5092	External analog alarm, min. 7	BOOL	R/O
0x13E5	5093	External analog alarm, min. 8	BOOL	R/O
0x13E6	5094	External analog alarm, max. 1	BOOL	R/O
0x13E7	5095	External analog alarm, max. 2	BOOL	R/O
0x13E8	5096	External analog alarm, max. 3	BOOL	R/O
0x13E9	5097	External analog alarm, max. 4	BOOL	R/O
0x13EA	5098	External analog alarm, max. 5	BOOL	R/O
0x13EB	5099	External analog alarm, max. 6	BOOL	R/O
0x13EC	5100	External analog alarm, max. 7	BOOL	R/O
0x13ED	5101	External analog alarm, max. 8	BOOL	R/O
0x13EE	5102	External digital input 1	BOOL	R/W
0x13EF	5103	External digital input 2	BOOL	R/W
0x13F0	5104	External digital input 3	BOOL	R/W
0x13F1	5105	External digital input 4	BOOL	R/W
0x13F2	5106	External digital input 5	BOOL	R/W
0x13F3	5107	External digital input 6	BOOL	R/W
0x13F4	5108	External digital input 7	BOOL	R/W
0x13F5	5109	External digital input 8	BOOL	R/W
0x13F6	5110	External digital input, alarm 1	BOOL	R/O
0x13F7	5111	External digital input, alarm 2	BOOL	R/O
0x13F8	5112	External digital input, alarm 3	BOOL	R/O
0x13F9	5113	External digital input, alarm 4	BOOL	R/O
0x13FA	5114	External digital input, alarm 5	BOOL	R/O
0x13FB	5115	External digital input, alarm 6	BOOL	R/O
0x13FC	5116	External digital input, alarm 7	BOOL	R/O
0x13FD	5117	External digital input, alarm 8	BOOL	R/O

7 Modbus address tables

7.3.18 Modbus error

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1532	5426	COM 1 error state	Bool	R/O
0x1533	5427	COM 1 last error code	Word	R/O
0x1534	5428	COM 2 error state	Bool	R/O
0x1535	5429	COM 2 last error code	Word	R/O
0x1536	5430	Modbus/TCP slave 1 error state	Bool	R/O
0x1537	5431	Modbus/TCP slave 1 last error code	Word	R/O
0x1538	5432	Modbus/TCP slave 2 error state	Bool	R/O
0x1539	5433	Modbus/TCP slave 2 last error code	Word	R/O

7.3.19 Data, controller 1

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x175C	5980	Standardized output level, controller 1	FLOAT	R/O
0x175E	5982	Average output level, controller 1	FLOAT	R/O
0x1760	5984	Actual value slope, controller 1	FLOAT	R/O
0x1762	5986	Parameter block number (1 to 4)	INT16	R/O
0x1763	5987	Setpoint value number (1 to 4)	INT16	R/O

7.3.20 Data, controller 2

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x1764	5988	Standardized output level, controller 2	FLOAT	R/O
0x1766	5990	Average output level, controller 2	FLOAT	R/O
0x1768	5992	Actual value slope, controller 2	FLOAT	R/O
0x176A	5994	Parameter block number (1 to 4)	INT16	R/O
0x176B	5995	Setpoint value number (1 to 4)	INT16	R/O

7 Modbus address tables

7.3.21 Program editor memory

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x4000	16384	Function 6=Delete program (0x86) 11=Read program (0x8B) 12=Write program (0x8C)	INT16	R/W
0x4002	16386	Error	INT16	R/O
0x4003	16387	Number of sections 1 to 50	INT16	R/W
0x4004	16388	Program number 1 to 10	INT16	R/W
0x4005	16389	Reserved	INT16	R/W

7.3.22 Program editor memory, section 1, 2 to 50

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
Section no. 1:				
0x4006	16390	Setpoint W1	FLOAT	R/W
0x4008	16392	Setpoint W2	FLOAT	R/W
0x400A	16394	Tolerance band min.	FLOAT	R/W
0x400C	16396	Tolerance band max.	FLOAT	R/W
0x400E	16398	Time	INT32	R/W
0x4010	16400	Gradient	FLOAT	R/W
0x4012	16402	Operating contacts	INT16	R/W
0x4013	16403	Repeat section (high byte INT8) Repeat number (low byte INT8)	INT16	R/W
0x4014	16404	Parameter block (high byte INT8) Gradient programming (low byte INT8)	INT16	R/W
Section no. 2: Has an address offset of 10 hex.				
0x4016	16406	Setpoint W1	FLOAT	R/W
0x4018	16408	Setpoint W2	FLOAT	R/W
0x401A	16410	Tolerance band min.	FLOAT	R/W
0x401C	16412	Tolerance band max.	FLOAT	R/W
0x401E	16414	Time	INT32	R/W
0x4020	16416	Gradient	FLOAT	R/W
0x4022	16418	Operating contacts	INT16	R/W
0x4023	16419	Repeat section (high byte INT8) Repeat number (low byte INT8)	INT16	R/W
0x4024	16420	Parameter block (high byte INT8) Gradient programming (low byte INT8)	INT16	R/W

7 Modbus address tables

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
Calculation formula: Address section no.(n) = 4006Hex + (n-1 Dec.)*10Hex Address section no.(50) = 4006Hex + (31 Hex)*10Hex.= 4006Hex+310Hex= 4316 Hex . .				
Section 50: Has an address offset of (31 Hex)*(10 Hex) = 310 Hex				
0x4316	17174	Setpoint W1	FLOAT	R/W
0x4318	17176	Setpoint W2	FLOAT	R/W
0x431A	17178	Tolerance band min.	FLOAT	R/W
0x431C	17180	Tolerance band max.	FLOAT	R/W
0x431E	17182	Time	INT32	R/W
0x4320	17184	Gradient	FLOAT	R/W
0x4322	17186	Operating contacts	INT16	R/W
0x4323	17187	Repeat section (high byte INT8) Repeat number (low byte INT8)	INT16	R/W
0x4324	17188	Parameter block (high byte INT8) Gradient programming (low byte INT8)	INT16	R/W

7.3.23 Program name

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x4800	18432	Program name, editing memory	Char73	R/W
0xA800	43008	Current program name of the program running	Char73	R/O

7.3.24 Compiled reading frames

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x8000	32768	Read Frame 1	Byte[254]	R/W
0x8080	32896	Read Frame 2	Byte[254]	R/W
0x8100	33024	Read Frame 3	Byte[254]	R/W
0x8180	33152	Read Frame 4	Byte[254]	R/W
0x8200	33280	Read Frame 5	Byte[254]	R/W
0x8280	33408	Read Frame 6	Byte[254]	R/W
0x8300	33536	Read Frame 7	Byte[254]	R/W
0x8380	33664	Read Frame 8	Byte[254]	R/W

7 Modbus address tables

7.3.25 Compiled writing frames

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x8800	34816	Write Frame 1	Byte[254]	R/O
0x8880	34944	Write Frame 2	Byte[254]	R/O
0x8900	35072	Write Frame 3	Byte[254]	R/O
0x8980	35200	Write Frame 4	Byte[254]	R/O
0x8A00	35328	Write Frame 5	Byte[254]	R/O
0x8A80	35456	Write Frame 6	Byte[254]	R/O
0x8B00	35584	Write Frame 7	Byte[254]	R/O
0x8B80	35712	Write Frame 8	Byte[254]	R/O

7 Modbus address tables



JUMO GmbH & Co. KG

Street address:
Moritz-Juchheim-Straße 1
36039 Fulda, Germany
Delivery address:
Mackenrodtstraße 14
36039 Fulda, Germany
Postal address:
36035 Fulda, Germany
Phone: +49 661 6003-0
Fax: +49 661 6003-607
E-mail: mail@jumo.net
Internet: www.jumo.net

JUMO Instrument Co. Ltd.

JUMO House
Temple Bank, Riverway
Harlow - Essex CM20 2DY, UK
Phone: +44 1279 63 55 33
Fax: +44 1279 63 52 62
E-mail: sales@jumo.co.uk
Internet: www.jumo.co.uk

JUMO Process Control, Inc.

6733 Myers Road
East Syracuse, NY 13057, USA
Phone: 315-437-5866
1-800-554-5866
Fax: 315-437-5860
E-mail: info.us@jumo.net
Internet: www.jumousa.com