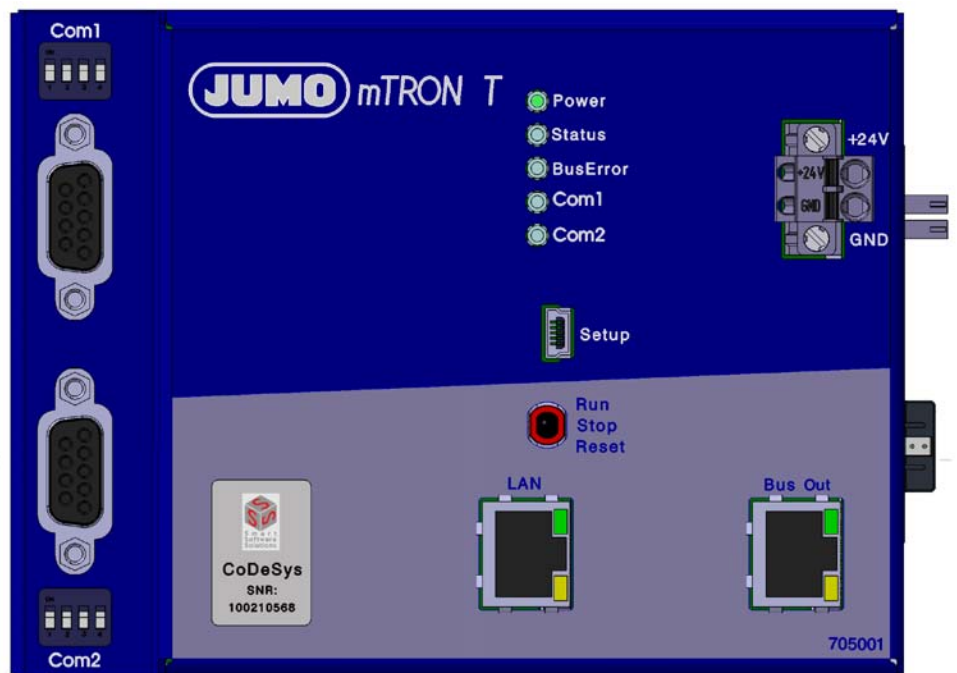


JUMO mTRON T

Measuring, Control, and Automation System Central Processing Unit



Interface Description Modbus



70500100T92Z001K000

V3.00/EN/00575586

1	Introduction	5
1.1	Available technical documentation	5
1.2	Content of the technical documentation	8
1.3	Safety information	11
2	Connecting interfaces	13
2.1	Position of the interfaces	13
2.2	Interface assignment	14
2.3	Serial interface	16
2.4	Ethernet interface	21
2.5	Ethernet settings for Modbus/TCP	22
3	Modbus protocol description	25
3.1	Master/slave principle	25
3.2	TRU transmission mode	25
3.3	Temporal sequence of communication	26
3.4	Structure of a Modbus telegram	27
3.5	Device address	28
3.6	Function codes	28
3.7	Transmission formats (integer, float, double, and text values)	37
3.8	Checksum (CRC16)	41
3.9	Error messages	42
4	Serial transmission modes	49
4.1	Modbus master operation via serial interface	49
4.2	Modbus slave operation via serial interface	51
4.3	PLC access to the serial interface	52
4.4	RS232 and RS422/485	55
5	Ethernet transmission modes	57
5.1	Modbus/TCP	57
5.2	Networking with Modbus/TCP	59
5.3	Modbus master with Modbus/TCP	59
5.4	Modbus slave with Modbus/TCP	60
5.5	HTTP	60
5.6	Browser connection and Web server	61
5.7	E-mail (SMTP and POP3)	62
6	User frames	67
6.1	General information	67
6.2	Structure of the reading and writing processes	67
6.3	Compiling Modbus frames	69

Contents

6.4	Examples for the data transmission options with frames	77
7	Modbus address tables	79
7.1	Data types and access types	79
7.2	Central processing unit addresses	79
7.3	Modbus addresses of other modules	97

1.1 Available technical documentation

The documents specified below are available for the measuring, control, and automation system (previous document number in parentheses).

1.1.1 General information

Product	Type of documentation	No.	Printed	PDF file
Measuring, control, and automation system	Data sheet	70500000T10...	-	X
	System manual ¹	70500000T90... (B 705000.0)	X	-
	Setup program manual	70500000T96... (B 705000.6)	-	X
	System description ²	70500000T98... (B 705000.8)	-	X

¹ Accessory subject to charge

² Includes an overview of the purpose and content of all documents

1.1.2 Base units

Product	Type of documentation	No.	Printed	PDF file
Central processing unit	Data sheet	70500100T10...	-	X
	Operating manual	70500100T90... (B 705001.0)	-	X
	Modbus interface description	70500100T92... (B 705001.2.0)	-	X
	PROFIBUS-DP interface description	70500103T92... (B 705001.2.3)	-	X
	digiLine interface description	70500106T92...	-	X
	Installation instructions	70500100T94... (B 705001.4)	X	X
	CODESYS OPC server operating manual	70500151T90... (B 705001.5.1)	-	X
	Process engineering application operating manual	70500152T90...	-	X
	Operating manual Thyristor power controller (type 70906x; integration in the measuring, control, and automation system)	70500153T90...	-	X

1 Introduction

1.1.3 Input/output modules

Product	Type of documentation	No.	Printed	PDF file
Multichannel controller module	Data sheet	70501000T10...	-	X
	Operating manual	70501000T90... (B 705010.0)	-	X
	Installation instructions	70501000T94... (B 705010.4)	X	X
Relay module 4-channel	Data sheet	70501500T10...	-	X
	Operating manual	70501500T90... (B 705015.0)	-	X
	Installation instructions	70501500T94... (B 705015.4)	X	X
Analog input module 4-channel	Data sheet	70502000T10...	-	X
	Operating manual	70502000T90... (B 705020.0)	-	X
	Installation instructions	70502000T94... (B 705020.4)	X	X
Analog input module 8-channel	Data sheet	70502100T10...	-	X
	Operating manual	70502100T90... (B 705021.0)	-	X
	Installation instructions	70502100T94... (B 705021.4)	X	X
Analog output module 4-channel	Data sheet	70502500T10...	-	X
	Operating manual	70502500T90...	-	X
	Installation instructions	70502500T94...	X	X
Digital input/output module 12-channel	Data sheet	70503000T10...	-	X
	Operating manual	70503000T90... (B 705030.0)	-	X
	Installation instructions	70503000T94... (B 705030.4)	X	X

1.1.4 Special modules

Product	Type of documentation	No.	Printed	PDF file
Router module	Data sheet	70504000T10...	-	X
	Installation instructions	70504000T94... (B 705040.4)	X	X

1.1.5 Operating, visualization, recording

Product	Type of documentation	No.	Printed	PDF file
Multifunction panel 840	Data sheet	70506000T10...	-	X
	Operating manual	70506000T90... (B 705060.0)	-	X
	Modbus interface description	70506000T92... (B 705060.2.0)	-	X
	Installation instructions	70506000T94... (B 705060.4)	X	X
Operating panels	Data sheet	70506500T10...	-	X
	Operating manual	70506500T90...	-	X

1.1.6 Power supply units

Product	Type of documentation	No.	Printed	PDF file
24 V power supply units	Data sheet	70509000T10...	-	X
	Operating instructions QS5.241		X	-
	Operating instructions QS10.241		X	-

1 Introduction

1.2 Content of the technical documentation

The documentation for the measuring, control, and automation system is intended for plant manufacturers and users with specialist training. It has a modular structure and comprises different sections.

In the following subsections, the various types of documents are listed (previous document number in parentheses).

1.2.1 Device documentation in printed form

7050XX00T94... (B 7050XX.4)

Installation instructions

A hard copy of the installation instructions is included in the scope of delivery of every module.

The installation instructions describe the installation of the device and the connection of the supply and signal cables. They also contain the order details and a list of technical data.

The scope of delivery for a power supply unit includes a hard copy of the operating instructions. These include information on installation and electrical connection.

70500000T90... (B 705000.0)

System manual

A hard copy of the system manual can be provided as an accessory subject to charge.

The system manual describes the scope of services of the measuring, control, and automation system and provides all information for project design and startup.

Index divider 1 "System description" summarizes the information applicable to all modules. Module-specific descriptions in the following sections complement the specifications stated here.

Index divider 2 "Setup program" describes the project design of the overall system.

1.2.2 Device documentation in the form of PDF files

The device documentation files specified below are saved as PDF files on the DVD contained in the scope of delivery of a base unit.

70500000T10... (T 705000)

Data sheet

The data sheet provides general information on the measuring, control, and automation system and forms the basis for plant planning and purchase decisions.

7050XX00T10... (T 7050XX)

Data sheet

The data sheets of the individual modules provide specific information, order details, and technical data.

70500000T98... (B 705000.8)

System description

The system description provides an overview of the measuring, control, and automation system. It describes properties that affect the entire system or are equally applicable for all modules.

7050XX00T90... (B 7050XX.0)

Operating manual

The operating manuals of the individual modules contain all information on installation, electrical connection, startup, operation, and – if required – parameterization and configuration.

7050XX0XT92... (B 7050XX.2.X)

Interface description

The interface description provides information about the use of that interface and on communication with other devices, superordinate systems or certain sensors.

7050XX00T94... (B 7050XX.4)

Installation instructions

The installation instructions describe the installation of the device and the connection of the supply and signal cables. The instructions also contain a list of the technical data.

7050XX5XT90... (B 7050XX.5.X)

Operating manual (application)

The operating manual describes the use of a certain application (e. g. PLC application).

1.2.3 Documentation for optional software

The manuals specified below are available on the Internet as PDF files. They also form part of the scope of delivery of the respective software.

70500000T96... (B 705000.6)

Setup program

The manual describes the function of the setup program.

70970100T90... (B 709701.0)

PC evaluation software PCA3000

The operating manual describes the operation and the features of the PC evaluation software. The PC evaluation software helps to visualize and evaluate the recorded process data (measurement data, batch data, messages, etc.).

70970200T90... (B 709702.0)

PCA communication software PCC

The operating manual describes the operation and the features of the PCA communication software. The PCA communication software is responsible for the data transfer from a device or system to a PC or to a network.

70075500T90... (B 700755.0)

Plant visualization software SVS3000

The operating manual describes the operation and features of the plant visualization software. The plant visualization software is responsible for networking interface-ready process devices with a PC.

1 Introduction

1.2.4 Device documentation on the Internet

All documents are available for download on the Internet at www.jumo.net.

Download procedure:

Step	Action
1	On the JUMO website, enter the number of the relevant product group in the search field at the top right (e.g. 705001 for the central processing unit) and start the search. <i>The search results are listed.</i>
2	Select product (click the link).
3	In the "Documentation" dropdown list, select the desired documentation in the required national language (click the link).
4	+++++Open the PDF document or save it as a file.

1.2.5 Training documents on the Internet

Training documents (eLearning courses) on various topics are available at www.jumo.net.

Procedure:

Step	Action
1	On the JUMO website, navigate to the "Support/Services" area.
2	In the "Information & Training" menu on the left-hand side, select "eLearning courses".
3	Click the link "Review of our eLearning courses".
4	Select the desired eLearning course from the overview (click the link). <i>The presentation starts.</i>

1.3 Safety information

1.3.1 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.

1.3.2 Note signs



NOTE!

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 Introduction

2.1 Position of the interfaces

The central processing unit is equipped with a standard LAN interface. It is designed to transmit data using the HTTP protocol (e.g. PC with setup program or Web browser) or the Modbus protocol (Modbus/TCP, master or slave).

Two serial interfaces are also optionally available as additional modules for the central processing unit (Com1 and Com2, 9-pin), which can be used as a RS232 or RS422/485. Both serial interfaces can be operated using the Modbus protocol (Modbus RTU; master or slave).



NOTE!

The type designation on the central processing unit's nameplate provides information on which optional interfaces were assembled **ex-works**.

Information on this can be found in the chapter "Identifying the device version" in the operating manual B 705001.0 or the installation instructions B 705001.4 (the installation instructions are included in the scope of delivery of the central processing unit).

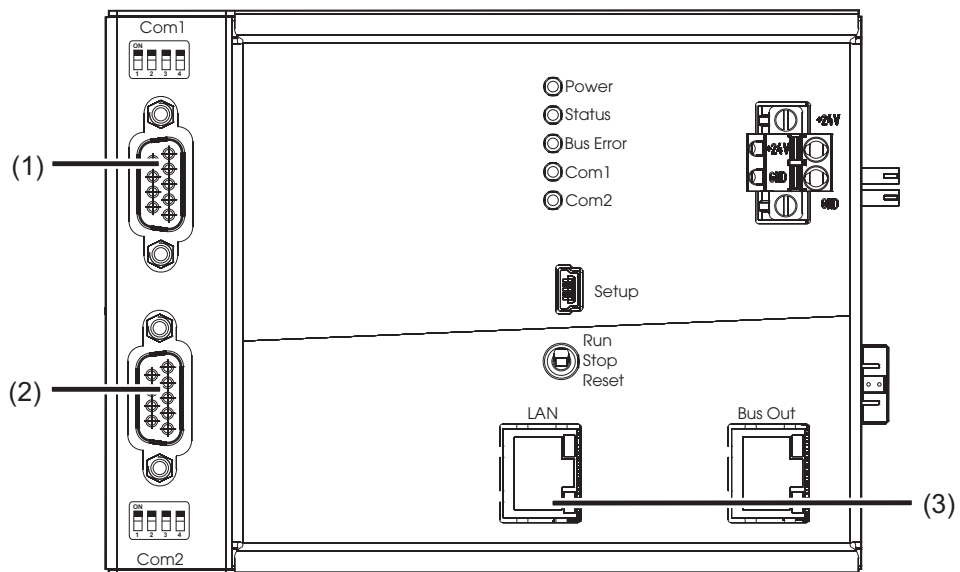


NOTE!

Optional interfaces can also be added **by the user**.

Information on this can be found in the chapter "Retrofitting interfaces" in the operating manual B 705001.0 or the installation instructions B 705001.4 (the installation instructions are included in the scope of delivery of the central processing unit).

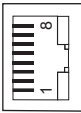
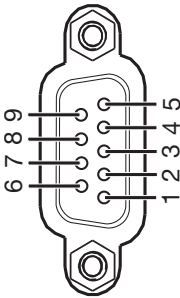
Front view of central processing unit



- (1) Com1 serial interface (9-pin)
- (2) Com2 serial interface (9-pin)
- (3) LAN interface (RJ45)

2 Connecting interfaces

2.2 Interface assignment

Connection	Designation	Connection element		
Ethernet	LAN		1 TX+	Transmission data +
			2 TX-	Transmission data -
			3 RX+	Received data +
			6 RX-	Received data -
Serial interface (RS232)	Com1, Com2		2 RxD	Received data
			3 TxD	Transmission data
			5 GND	Ground
Serial interface (RS422)	Com1, Com2		3 TxD+	Transmission data +
			4 RxD+	Received data +
		5 GND	Ground	
		8 TxD-	Transmission data -	
		9 RxD-	Received data -	
Serial interface (RS485)	Com1, Com2	3 TxD+/RxD+	Transmission/received data +	
		5 GND	Ground	
		8 TxD-/RxD-	Transmission/received data -	
PROFIBUS-DP	Com2	8 RxD/TxD-N (A)	Transmission/received data -	
		3 RxD/TxD-P (B)	Transmission/received data +	
		6 VP (+5 V)	Voltage supply	
		5 DGND	Ground	



NOTE!

A connecting cable with shielding must be used to connect the RS232 interface.
 A twisted connecting cable with shielding must be used to connect the RS422/485 interface.
 To avoid transmission errors, only the signals listed above may be connected.



NOTE!

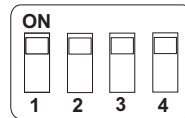
A RJ45 patch/crossover cable (CAT5 or higher) must be used to connect the LAN interface.

2.2.1 Terminating resistors

The internal terminating resistors for the Com1 and Com2 interfaces are only relevant for RS422/485.

The terminating resistors are deactivated by default. To activate them, DIP switches 1 to 4 for the relevant interface must be pushed upward using a suitable tool such as a ballpoint pen (ON position).

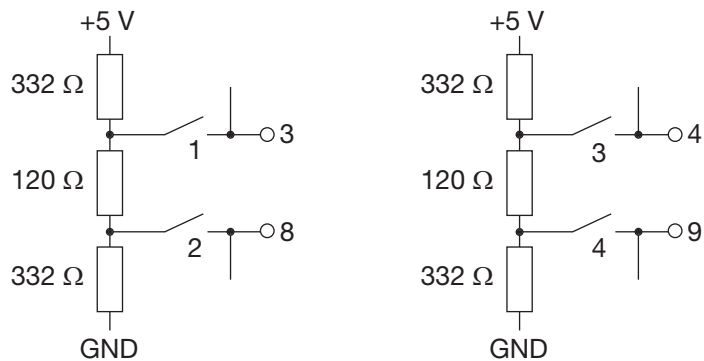
The following figure shows the position of the DIP switches when the terminating resistors are activated.



NOTE!

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

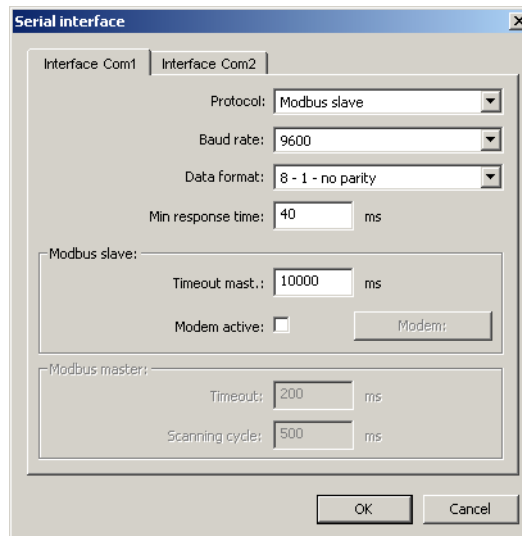
Internal terminating resistors



2 Connecting interfaces

2.3 Serial interface

Setup dialog



Parameters

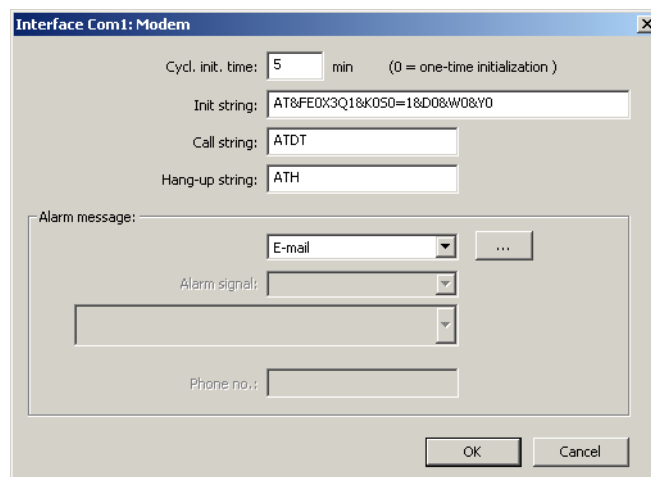
Parameter	Selection/settings	Description
Protocol	Modbus slave Modbus master	Modbus RTU (central processing unit as Modbus slave) The device address is assigned dynamically during hardware configuration. Modbus RTU (central processing unit as Modbus master)
Baud rate	Baud rate with which the interface is operated. 9600 19200 38400	9600 baud 19200 baud 38400 baud
Data format	Data format with which the interface is operated. 8 - 1 - no parity 8 - 1 - odd parity 8 - 1 - even parity	8 data bits, 1 stop bit, no parity 8 data bits, 1 stop bit, odd parity 8 data bits, 1 stop bit, even parity
Min response time	0 to 40 to 500 ms	The minimum response time is adhered to by the Modbus slave before a response is sent following a data request.
Modbus slave		
Timeout mast.	60 to 10000 to 60000 ms	Master monitoring time After this time, a timeout is identified in the Modbus master. An internal digital signal is set in the event of a timeout.

2 Connecting interfaces


Parameter	Selection/settings	Description
Modem active	Not selected (empty) Selected (checkmark)	No modem operation (Modbus slave is connected directly to serial bus). Modem operation (Modbus slave is connected to the Modbus master via the modem). Additional settings are required here ("Modem" button).
Modbus master		
Timeout	60 to 200 to 10000 ms	A request sent by the master is defined as faulty if no answer is received within this time.
Scanning cycle	60 to 500 to 99999 ms	The Modbus master requests the Modbus master data from the Modbus slave at these intervals.

2.3.1 Modem



Setup dialog



Parameters

Parameter	Selection/settings	Description
Cycl. init. time	0 to 5 to 255 min	Time for cyclical initialization of the modem (if the modem is switched on after the system). 0 = one-time initialization (after system has been switched on)
Init string 	AT&FE0X3Q1&K0S0=1&D0&W0&Y0 (ASCII; max. 40 characters)	AT command for modem initialization This default initialization string configures the modem so that it can be called from an external source, answers independently, and receives Modbus commands.

2 Connecting interfaces

Parameter	Selection/settings	Description
Call string 	ATDT (ASCII; max. 24 characters)	AT command for establishing a connection via the modem ATDT = selection with dial tone (DTMF)
Hang-up string 	ATH (ASCII; max. 16 characters)	AT command for disconnection via the modem ATH (or ATH0) = hang up
Alarm message		
Alarm type	Issue of alarm message	
	E-mail	In the event of an alarm, an e-mail is sent (via the e-mail server after connecting to the Internet).
	PC visualization	In the event of an alarm, a modem connection is established to a PC with process visualization software.
Alarm signal	Signal that triggers the alarm message (only for "PC visualization" alarm type)	
	Inactive Module (source of signal)	No alarm message Alarm message is triggered by a signal (high active) that must be selected in the following field (digital selector).
Phone no.	(None) (ASCII; max. 24 characters)	Telephone number for establishing connection to a PC with process visualization software (Only for "PC visualization" alarm type)

Init string

The following Init string is required for operation as a Modbus slave:
AT&FE0X3Q1&K0S0=1&D0&W0&Y0

AT&F = Load current manufacturer profile
E0 = Deactivate character echo
X3 = Deactivate dial tone detection, activate busy tone detection
Q1 = Deactivate command responses
&K0 = Deactivate data flow control
S0=1 = Auto answer after first ring
&D0 = Ignore DTR signal
&W0 = Save current configuration as profile 0
&Y0 = Use profile 0 after switch-on

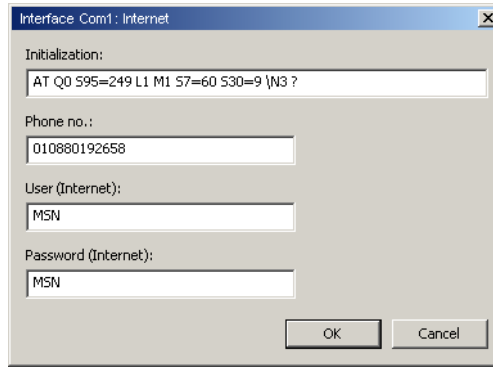
Call string, hang-up string

The dialing and hang-up strings are required if the modem needs to establish a connection to the Internet (to send e-mail via e-mail server) or a PC with process visualization software in the event of an alarm.

Additional settings

To open this window, use the "..." button:

2 Connecting interfaces



Parameters

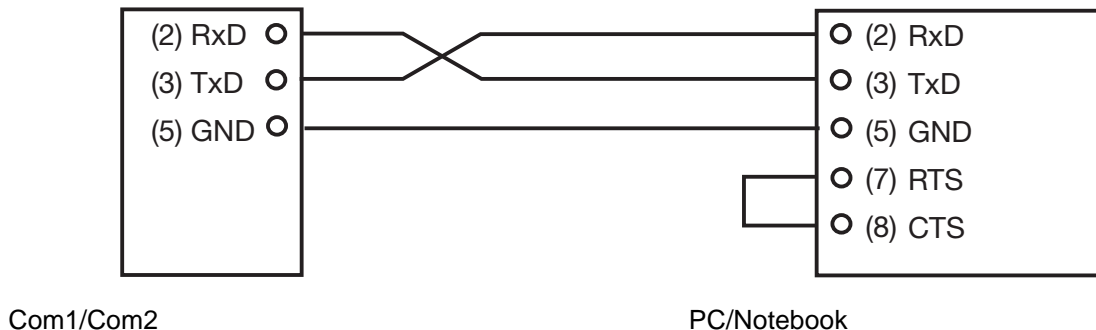
Parameter	Selection/settings	Description
Initialization	AT Q0 S95=249 L1 M1 S7=60 S30=9 \N3 ? (ASCII; max. 50 characters)	AT command for modem changeover This default initialization string switches the modem to the mode for connecting to the Internet (e-mail server).
Phone no.	010880192658 (ASCII; max. 24 characters)	Telephone number for connecting to the Internet (to be requested from the Internet provider)
User (Internet)	MSN (ASCII; max. 64 characters)	User name for logon when connecting to the Internet (to be requested from the Internet provider)
Password (Internet)	MSN (ASCII; max. 64 characters)	Password for logon when connecting to the Internet (to be requested from the Internet provider)

2.3.2 Handshake lines for RS232

When using Com1/Com2 as a RS232 interface, the handshake lines (RTS, CTS) are not used. The RTS line coming from the master is ignored. The response is immediately transmitted from the slave. The CTS line of the master remains open.

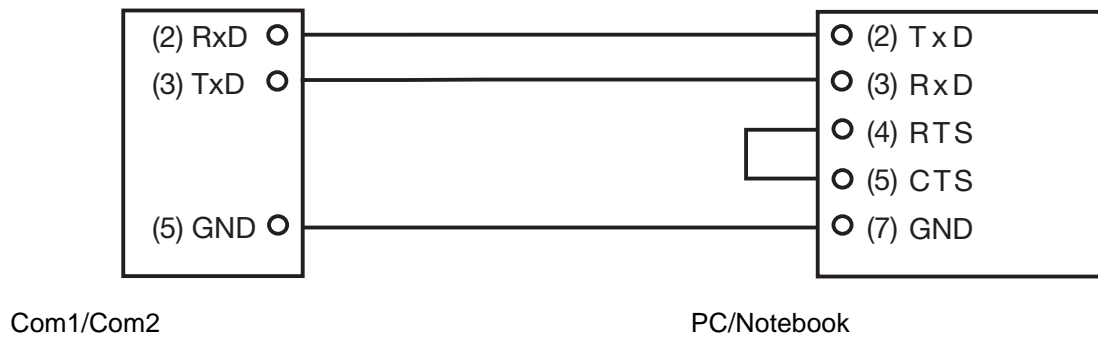
If the program used evaluates the handshake lines, they must be bridged in the cable.

Connecting a PC/Notebook with 9-pin Sub-D socket to Com1/Com2



2 Connecting interfaces

Connecting a PC/Notebook with 25-pin Sub-D socket to Com1/Com2



2.4 Ethernet interface

General information

To use the Ethernet interface, a patch/crossover cable is required that is equipped with a RJ45 plug. The Ethernet interface is configured in the setup software.

Required parameters, such as DHCP functionality, IP address, sub-net mask, gateway address, DNS device name, DNS server, and transfer rate, can be configured in the setup program under **PROJECT NAME > CPU > ONLINE PARAMETERS > ETHERNET.**

The following transmission options can be used via Ethernet:

- Modbus/TCP as the slave provides third-party masters as servers
- Modbus/TCP as the master for reading/writing individual values or entire data frames
- Data transmission using HTTP protocol
- Sending e-mails using SMTP protocol

The DHCP and DNS protocols are also supported. You do not have to obtain a dynamic IP address via DHCP; instead, you can manually assign a fixed address using the setup program. If DHCP is used, the DNS functionality can be used. The device registers itself with the DHCP server using a unique name. It can be uniquely addressed using this name. A DNS device name is then used for addressing.

DNS should also always be used in combination with DHCP, since otherwise the device can no longer be accessed via Ethernet if the IP address is changed.

Supported Ethernet speeds

Speed	Mode
Auto negation	Default setting
10 Mbit/s	Half duplex
10 Mbit/s	Full duplex
100 Mbit/s	Half duplex
100 Mbit/s	Full duplex



NOTE!

Configuration changes are not applied until after the device has been restarted. Additional information on the individual settings for the Ethernet are located in the operating manual of the central processing unit (B 705001.0).

2 Connecting interfaces

2.5 Ethernet settings for Modbus/TCP

This menu is used to implement settings for the Modbus/TCP operating mode. If the central processing unit is operating as a Modbus master, it can communicate with up to four external devices (Modbus slaves; device 1 to 4). If it is operating as a Modbus slave, two external devices (Modbus master) can simultaneously access the central processing unit.

Setup dialog

Parameters

Parameter	Selection/settings	Description
Modbus slave (central processing unit as Modbus slave)		
Port	0 to 502 to 1024	TCP port for Modbus/TCP Changes to the port are not applied until after the system has been restarted.
Modbus master (central processing unit as Modbus master)		
Timeout	60 to 2000 to 10000 ms	A request sent by the master is defined as faulty if no answer is received within this time.
Scanning cycle	60 to 500 to 99999 ms	The Modbus master requests the Modbus master data from the Modbus slave at these intervals.
IP-address	0.0.0.0	IP address of the external device (Modbus slave) The address must be set.
Port	0 to 502 to 1024	TCP port of the external device for Modbus/TCP



NOTE!

To ensure that fixed IP addresses are used, the DHCP must be deactivated in the devices involved, if applicable.

2 Connecting interfaces



NOTE!

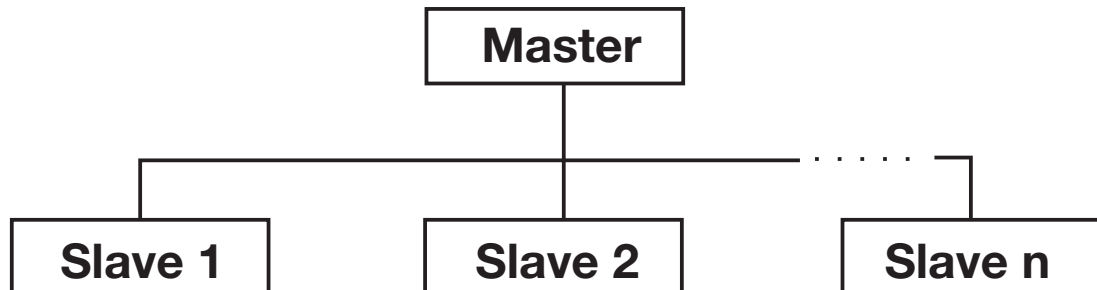
The transfer times in an Ethernet network depend in part on the network architecture and the capacity utilization. This may result in delays during updates of process values.

2 Connecting interfaces

3 Modbus protocol description

3.1 Master/slave principle

Communication between a master (e.g. PC, notebook or central processing unit) and a slave device (e.g. measuring and control system) and a Modbus takes place according to the master-slave principle, in the form of data request/instruction - response.



The master controls the data exchange, the slaves only have a response function. They are identified by their device address.



NOTE!

The central processing unit can be operated as a Modbus master as well as a Modbus slave. The master and slave function can also be used in parallel.

This makes it possible to transmit the external analog, integer, and digital inputs and texts (variables) both from a master device to the central processing unit (slave) as well as actively have the central processing unit (master) import them from one or several slaves.

If the central processing unit is operating as a master, the corresponding Modbus and device addresses must be assigned using the setup settings of the Modbus frames.

3.2 TRU transmission mode

In addition to Modbus/TCP, RTU mode (Remote Terminal Unit) can also be used as the transfer mode. The data is transmitted in the binary format with 8 or 16 bits for integer values and 32 bits for float values. The most significant bit (msb) is transmitted first. The ASCII operating mode is not supported.

Data format

The data format describes the structure of a character transmitted.

Data format (configuration)	Start bit	Data bits	Parity bit	Stop bit	Number of bits
8 - 1 - no parity	1	8	0	1	10
8 - 1 - odd parity	1	8	1	1	11
8 - 1 - even parity	1	8	1	1	11

3 Modbus protocol description

3.3 Temporal sequence of communication

Character transmission time

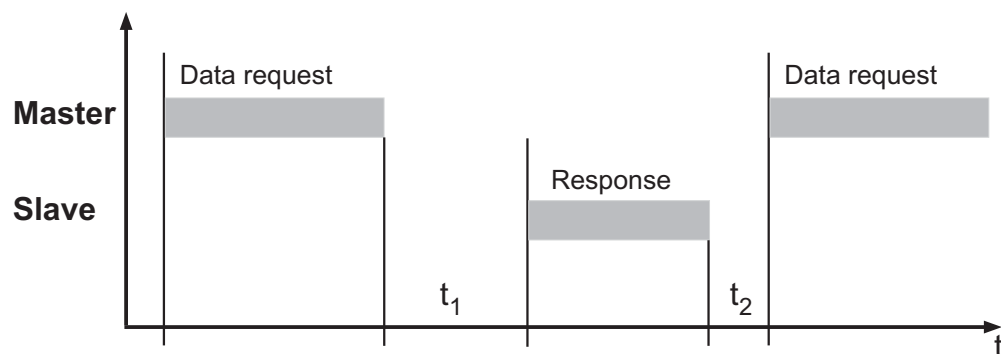
The character transmission time (time taken to transmit one character with 8 data bits) depends on the baud rate and the number of bits of the character (see table for data format):

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

Baud rate [Bd]	Number of bits	Character transmission time [ms]
38400	11	0,286
	10	0,260
19200	11	0,573
	10	0,521
9600	11	1,146
	10	1,042

Time diagram of a data request

A data request proceeds according to the following time diagram:



The data request and response consist of several characters (each with a start bit, 8 data bits, a parity bit if necessary, and a stop bit) which are transmitted coherently.

t_1 The slave has to observe this waiting period before sending the response

min.: 5 ms

typically: 5 to 35 ms

max.:35 ms or the minimum response time set in the configuration

t_2 The master has to observe this waiting period before starting a new data request.

with RS232: at least 3.5 times the character transmission time (end identifier)

with RS485: 35 ms



NOTE!

The waiting periods t_1 and t_2 also contain the end identifier (3.5 × character transmission time), which follows after each data request or response.

3 Modbus protocol description



NOTE!

A minimum response time can be configured in the central processing unit using the setup program under **CPU > CONFIGURATION LEVEL > SERIAL INTERFACE**. This preset time is the minimum waiting time before an answer is transmitted (0 to 500 ms). If a smaller value is set, then the response time may be longer than the preset value (internal processing takes longer), the central processing unit responds as soon as internal processing is completed. A preset time of 0 ms means that the device responds with the maximum possible speed. The minimum response time, which can be set is required by the RS485 interface in the master, to switch over the interface drivers from transmit to receive. This parameter is not required for the RS232 interface.



NOTE!

No data requests from the master are permitted during t_1 and t_2 and during the slave response time. Data requests made during t_1 and t_2 are ignored by the slave. Data requests during the response time will result in the invalidation of all data currently on the bus.

3.4 Structure of a Modbus telegram

Data structure

All telegrams have the same structure:

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

Each telegram contains four fields:

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

3 Modbus protocol description

3.5 Device address

The device address can be set between 1_{DEC} and 254_{DEC}. It is assigned for the central processing unit and each module in the setup program under **PROJECT NAME > PROJECT > HARDWARE ARRANGEMENT**. Each Modbus station must have a unique device address. The following data exchange options are available for accessing the connected stations.

Query

This is a data request/instruction from the master to a slave via the corresponding device address (1 to 254). The slave addressed responds.

Broadcast

The broadcast is an instruction from the master to all slaves via the device address 0 (e.g. for transmitting a specific value to all slaves).

The connected slaves do not respond. In such a case, the correct acceptance of the value by the slaves should be checked by a subsequent readout at each individual slave. A data request with the device address 0 is meaningless.



NOTE!

A maximum of 31 slaves can be accessed via the RS485 interface.

The device address 0 is reserved as a Modbus broadcast address:

an instruction from the master to address 0 is executed by all slaves, however, none of the slaves respond (since it would otherwise lead to a data collision).

The address is specified in binary format in the transmission protocol.

3.6 Function codes

Function overview

The functions described in the following are available for the readout of measured values, device and process data as well as for writing specific data.

Function number	Function	Limiting
0x01 or 0x02	Read n bit	Max. 256 bits (16 bytes)
0x03 or 0x04	Read n words	Max. 127 words (254 bytes)
0x05	Write one bit	Max. 1 bit
0x06	Write one word	Max. 1 word (2 bytes)
0x10	Write n words	Max. 127 words (254 bytes)



NOTE!

A hexadecimal number is marked by a preceding "0x".

Example: 0x0010 (= 16_{DEC})



NOTE!

If the central processing unit does not respond to these functions or outputs an error code, it can be evaluated.

⇒ Chapter 3.9 "Error messages", page 42

3 Modbus protocol description

Important information on the bit commands

To use bit commands correctly and evaluate the results correctly, it is important to understand the order in which the data words and the bits they contain are arranged.

The Modbus standard specifies that when reading out word by word the most significant byte (msb) is specified first. The bit arrangement within the data words, however, starts with the least significant bit (lsb). This must be observed when reading out bit values.

Example for calculating the bit address from the word address

⇒ Chapter 7 "Modbus address tables", page 79

The word address tables are specified in hexadecimal form in the Modbus address tables in the left column. The bit addresses must be calculated from this word address to read or write individual bits. To determine the bit address of bit 8 for the word address 0x0009 in a bit field, the following calculation must be performed: bit address = (word address_{Hex} × 10_{Hex}) + bit number_{Hex}

Hexadecimal calculation	Decimal calculation
Bit address = (0x0009 × 0x10) + 0x8	Bit address = (9 × 16) + 8
Result: 98 _{HEX}	Result: 152 _{DEC}

3 Modbus protocol description

3.6.1 Read n bit

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

Data request

Slave address	Function 0x01 or 0x02	Address of first bit	Number of bit	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave address	Function 0x01 or 0x02	Number of bytes read	Bit value(s)	Checksum CRC
1 byte	1 byte	1 byte	x bytes	2 bytes



NOTE!

The response always comes in full bytes of 8 bits each. Bit values that are not requested are filled with the value 0.

Example with the data type bit field 32

The objective is to read out the state of the digital inputs 2 and 3 of a 12-channel digital input/output module. Its Modbus address here is 5.

According to Chapter 7.3.5 "Modbus addresses for each 12-channel digital input/output module", page 117, the word address for this is 0x0002. It contains the values in bits 1 and 2

Hexadecimal calculation of bit address for bit 1	Decimal calculation of bit address for bit 1
Bit address = $(0x0002 \times 0x10) + 0x1$	Bit address = $(2 \times 16) + 1$
Result: 21	Result: 33

Hexadecimal calculation of bit address for bit 2	Decimal calculation of bit address for bit 2
Bit address = $(0x0002 \times 0x10) + 0x2$	Bit address = $(2 \times 16) + 2$
Result: 22	Result: 34

Since this example involves a bit field 32, you must make sure that the data is in a double word (32 bit). The start address 0x0002 is the low word. The high word, which is read out first, has the address 0x0003. As a result, when calculating the correct bit address, the value 10_{Hex} or alternatively the value 16_{Dec} must be **added** again to the calculated bit address to request the desired bit value at the right spot in the low word.

You receive 31_{Hex} or 49_{Dec} for the bit address of bit 1 and 32_{Hex} or 50_{Dec} for the bit address of bit 2. A corresponding data request in Modbus format is provided below:

3 Modbus protocol description

Data request:

05	01	00 31	00 02	ED 80
Slave	Function			CRC

Response:

05	01	01	02	D1 79
Slave	Function	Bytes	Bit value	CRC

The value of the read out bits is 02_{Hex} (= 2_{Dec} = 10_{Bin}): bit 0 = 0 and bit 1 = 1



NOTE!

If the bits are not to be read out of a 32-bit field, but rather from a 16-bit field, it is not necessary to add 10_{Hex} or 16_{Dec}, since there is no preceding high word.

Example with the data type bit field 64

The objective is to read out the state of the digital variables 1 to 54 of the multifunction panel. Its Modbus address here in the example is 7.

According to Chapter 7.3.1 "Modbus addresses for each HMI module", page 97, the word address for this is 0x0002. It contains the values in bits 0 to 53:

Hexadecimal calculation of the bit address for the first bit of the bit field	Decimal calculation of the bit address for the first bit of the bit field
Bit address = (0x0002 × 0x10) + 0x0	Bit address = (2 × 16) + 0
Result: 0x20	Result: 32

Hexadecimal calculation of the bit address for the last bit of the bit field	Decimal calculation of the bit address for the last bit of the bit field
Bit address = 0x20 + 0x40	Bit address = 32 + 64
Result: 60	Result: 96

Since it is a 64-bit field, 64 values in the example are to be read starting with the address 20_{Hex}.

Data request:

07	01	00 20	00 40	56 3C
Slave	Function	Address of 1st bit	Number of bit	CRC

Response:

07	01	08	10 00	00 00	00 01	01 00	C9 7B
Slave	Function	Bytes read	Bit values			CRC	

As is also the case in the example with the data type bit field 32, the bit values are not directly read out of the data response without detecting the exact assignment.

A bit field 64 consists of 4 words each with 16 bits. The word with the highest value is always transmitted first, followed by the word with the next-highest value and so on until the lowest val-

3 Modbus protocol description

ue is reached. The least significant bit (lsb) in each word is also transmitted first here. This example yields the following arrangement:

Bit 48 to bit 53	Bit 32 to bit 47	Bit 16 to bit 31	Bit 0 to bit 15
Highest value	Lowest value
DV 49 to DV 54	DV 33 to DV 48	DV 17 to DV 32	DV 1 to DV 16

The bit value from the data response must now be analyzed byte for byte:

Hexadecimal	Decimal	Binary	Byte type	Word type
10	16	0001 0000	Low byte	Highest value
00	0	0000 0000	High byte	
00	0	0000 0000	Low byte	...
00	0	0000 0000	High byte	
00	0	0000 0000	Low byte	...
01	1	0000 0001	High byte	
01	1	0000 0001	Low byte	Lowest value
00	0	0000 0000	High byte	

This analysis and arrangement from the table yields:

- first low byte (highest word): 0001 0000_{Bin}
That means: the 5th spur starting with spur 49 has the value 1.
- The next to last high byte: 0000 0001_{Bin}
That means: the 9th spur starting with spur 17 has the value 1.
- Last low byte (lowest word): value 0000 0001_{Bin}
That means: the 1st spur has the value 1.

The spurs 1, 25, and 53 therefore have the value 1. The rest have the value 0.

3 Modbus protocol description

3.6.2 Reading n words

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

Data request

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

Example

Read the IP address of the central processing unit. Here in the example, it is the address 10.10.1.69. Since each IP address range is stored in a word, 4 words (8 bytes) must be read out. They and other Modbus addresses can be found in the Chapter 7 "Modbus address tables", page 79.

Data request:

01	03	10 6D	00 04	14 D1
Slave	Function	Address of the 1st word	Number of words	CRC

Response (values in the Modbus float format):

01	03	08	00 0A	00 0A	00 01	00 45	E5 37
Slave	Function	Bytes read	10	10	1	69	CRC
IP-address							

3 Modbus protocol description

3.6.3 Write one bit

In the write bit function, the data blocks for the instruction and response are identical.

Instruction

Slave address	Function 0x05	Bit address	Bit value	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes
				Bit value xx = 00: bit is set to 0
				Bit value xx = FF: bit is set to 1

Response

Slave address	Function 0x05	Bit address	Bit value	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example

In this example, manual mode for controller channel 1 of a controller module is to be activated at Modbus address 2. To do this, it must logically be set to "1":

Instruction:

02	05	1C 80	FF 00	71 8A
Slave	Function	Bit address	Set bit to 1	CRC

Response:

02	05	1C 80	FF 00	71 8A
Slave	Function	Bit address	Set bit to 1	CRC

3 Modbus protocol description

3.6.4 Write one word

In the write word function, the data blocks for the instruction and response are identical.

Instruction

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

Response

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

Example

In this example, a command is to be written to program generator 1 of the central processing unit. The slave address of the central processing unit here is 1, the word address is 0x1691 (Kapitel 7.2 "Central processing unit addresses", ab Seite 79), and the value to be written is to be 5 for manual mode.

Instruction:

01	06	16 91	00 05	6C 1C
Slave	Function	Word address	Value	CRC

Response:

01	06	16 91	00 05	6C 1C
Slave	Function	Word address	Value	CRC

3 Modbus protocol description

3.6.5 Write n words

Instruction

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

Example

Writing the word "test" (ASCII code: 0x54 0x65 0x73 0x74 0x00) starting with the word address 0x13F1, so that this text is written as an external batch text 1 for the multifunction panel interface 1 (coming from the central processing unit) to the multifunction panel.

⇒ Chapter 7.2 "Central processing unit addresses", page 79

Instruction:

01	10	13 F1	00 03	06	54 65 73 74 00 00	40 F2
Slave	Function	Address of the 1st word	Number of words	Number of bytes	Text in ASCII	CRC

Response:

01	10	13 F1	00 03	F5 80
Slave	Function	Address of the 1st word	Number of words	CRC

3 Modbus protocol description

3.7 Transmission formats (integer, float, double, and text values)

Function 0x03 or 0x04 (reading n words) is used to read out integer, float, double, and text values.

Data request

Slave address	Function 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 in the following format via Modbus:
first the high byte and then the low byte.

Response

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

3.7.1 Integer values

Example

In this example, the value of the integer variable 1 is to be read out at address 0x11F1 of the central processing unit. The value here is to be "4" (word value 0x0004).

Data request:

01	03	11 F1	00 01	C5 D0
Slave	Function	Address of the 1st word	Number of words	CRC

Response (values in the Modbus float format):

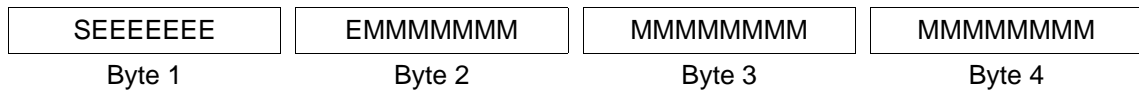
01	03	02	00 04	8D 19
Slave	Function	Bytes read	Integer value	CRC

3 Modbus protocol description

3.7.2 Float values

For float values, Modbus operates with the IEEE 754 standard format (32 bit), but with the difference that byte 1 and 2 are changed over with byte 3 and 4.

Single-float format (32 bit) according to standard IEEE 754

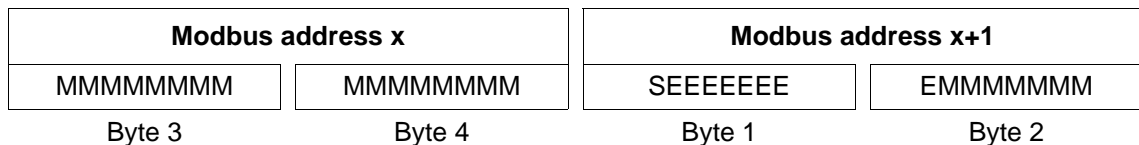


S - sign bit

E - exponent (two's complement)

M - 23 bits normalized mantissa

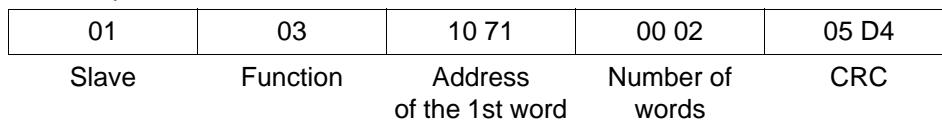
Modbus float format



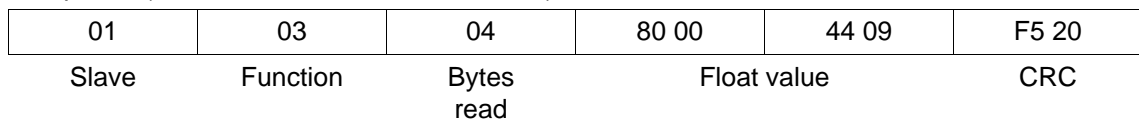
Example

In this example, the value of analog variable 1 is to be read out at address 0x1071 of the central processing unit. The value here is to be 550.0 (0x44098000 in the IEEE 754 format).

Data request:

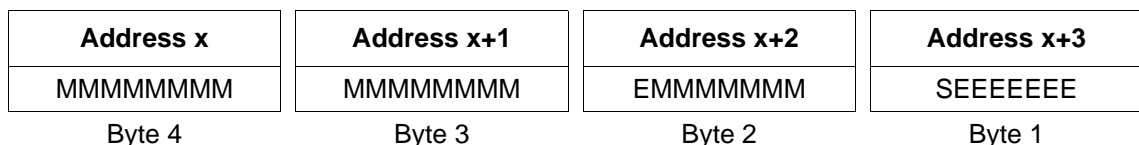


Response (values in the Modbus float format):



Once transmission from the central processing unit is completed, the bytes of the float value need to be changed over accordingly. A large number of compilers (e.g. Microsoft Visual C++) file the float values in the following order:

Float value



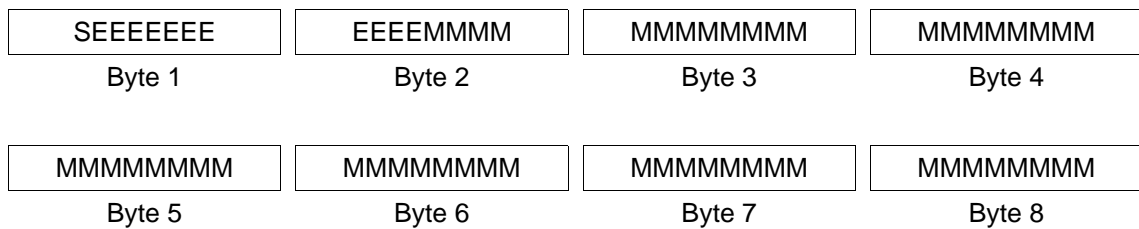
3 Modbus protocol description

**NOTE!**

The order of the bytes depends on how the float values are saved in the respective application. The bytes may have to be changed over in the interface program accordingly.

3.7.3 Double values

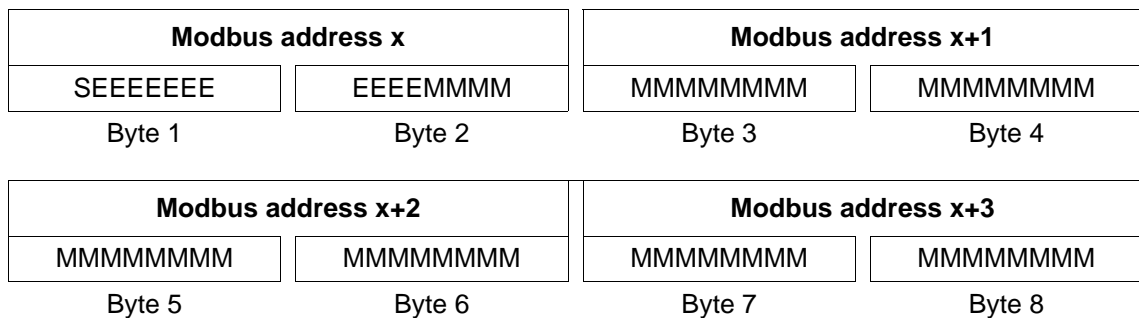
For double values, Modbus also operated with the IEE 754 standard format (32 bit). Unlike float values, bytes are not changed over for double values.

Double-float format (32 bit) acc. to IEEE 754 standard

S - sign bit

E - exponent (two's complement)

M - 52 bits normalized mantissa

Modbus double format

3 Modbus protocol description

Example

In this example, the value of analog variable 1 is to be read out at address 0x1071 of the central processing unit. The value here is to be 1234567.89 (0x4132D687E3D70A3D in the IEEE 754 format).

Data request:

01	03	10 71	00 04	D2 10
Slave	Function	Address of the 1st word	Number of words	CRC

Response (values in the Modbus double format):

01	03	08	41 32	D6 87	E3 D7	0A 3D	CD A4
Slave	Function	Bytes read	Double value			CRC	



NOTE!

The order of the bytes depends on how the double values are saved in the respective application. The bytes may have to be changed over in the interface program accordingly.

3.7.4 Strings (texts)

Strings are transmitted in the ASCII format.



NOTE!

To mark the end, the last character to be transmitted must always be a "\0" (ASCII code 0x00). The subsequent characters have no meaning.

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

The maximum lengths for strings specified in the address tables contain the final "\0". This means, in the case of "char 11", the text can consist of max. 10 readable characters.

⇒ Chapter 7 "Modbus address tables", page 79

Example

Request the text of address 0x13F1; the string "test" (ASCII code: 0x54, 0x65, 0x73, 0x74, 0x00) is under this address.

Request: 010313F100027E11 (CRC16 = 117E)

Response: 010304**54657374**0000AAB5B6 (CRC16 = B6B5)



NOTE!

The value (here: AA) before the CRC sum (here: B5B6) is not considered, since it follows the end marker "\0".

3 Modbus protocol description

3.8 Checksum (CRC16)

Calculation scheme

The checksum (CRC16) is used to detect transfer errors. If an error is identified during evaluation, the device concerned does not respond.

CRC = 0xFFFF	
CRC = CRC XOR BytesOfMessage	
For (1 to 8)	
CRC = SHR(CRC)	
if (shifted to the right flag = 1	
then	else
CRC = CRC XOR 0xA0001	
while (not all BytesOfMessage processed);	



NOTE!

The low byte of the checksum is transmitted first.

Example: the CRC16 checksum CC DD is transmitted and displayed in the order DD CC.

Example

Request status of digital variable 1 at address 0x1371:

Instruction: read a word from address 0x1371

01	03	13 71	00 01	95 D0
Slave	Function	Address	Read a word	CRC

Response (CRC16 = 0x8479)

01	03	02	00 01	79 84
Slave	Function	Number of bytes	Word 1	CRC

Word 1 = 1 means that digital variable 1 = 1.

3 Modbus protocol description

3.9 Error messages

3.9.1 Modbus error codes

The slave device does not respond

The slave will not respond in the following cases:

- The baud rate and/or data format of the master and slave do not match.
- The used device address does not match with the slave address contained in the protocol.
- The checksum (CRC) is not correct.
- The instruction from the master is incomplete or over-defined.
- The number of words to be read is zero.

In these cases, the data request should be sent again after a time-out time of 2 s has elapsed.

Error codes

If the data request of the master has been received by the slave without transmission errors but could not be processed, the slave will respond with an error code. The following error codes may occur:

- 01 = Invalid function
- 02 = Invalid address or excessive number of words or bits are to be read or written
- 03 = Value is outside of the permissible range
- 08 = Value is write-protected

Response in case of error

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 highest-value byte (msb) to 1.

Example

Data request:

01	06	1636	0001	FCA2
Slave	Write word	Word address	Word value	CRC

Response (with error code 2):

01	86	08	43A6
Slave	Function OR	Errors	CRC

Response with error code 08, because address 0x1636 is write-protected.

3 Modbus protocol description

3.9.2 Error messages for invalid values

For measured values in the float format, the error number appears directly in the value, i.e. it contains the error number instead of the measured value.

Error code with float values	Errors
1.0×10^{37}	Underrange
2.0×10^{37}	Overrange
3.0×10^{37}	No valid input value
4.0×10^{37}	Division by zero
5.0×10^{37}	Math error
6.0×10^{37}	Invalid terminal temperature of thermocouple
7.0×10^{37}	Invalid float value
8.0×10^{37}	Integrator or statistics destroyed

Example

Reading in the analog input 1 of a 4-channel analog input module at Modbus address 8:

Data request:

08	03	00 52	00 02	43 65
Slave	Function	Word address	Number of words	CRC

Response:

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

The measured value 0x7DB48E52 ($=3.0 \times 10^{37}$) supplied by analog input 1 indicates that the input value is invalid.

3 Modbus protocol description

3.9.3 Error codes as integer return values

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

Error codes

Error code	Description
Error list: program memory management	
1	Program cannot be created
2	Program not available
3	Program cannot be deleted
4	Segment cannot be deleted
5	Checksum cannot be saved
6	Checksum cannot be read
7	Program cannot be copied
8	Segment cannot be copied
9	Program checksum error
10	Program pointer tab checksum error
11	Program memory end
12	Section not available
13	Repeat jump labels
Event list: general input and output	
14	Please acknowledge with ENTER
15	Invalid number of places
16	The entry contains invalid characters
17	Value not within the limits
18	Segment incorrectly programmed
19	Password error
Error list: PROFIBUS order processing	
20	Busy flag not reset by the master
21	Inadmissible job
22	Error on data acceptance
23	No cyclical data existing
24	Inadmissible structure length
25	Inadmissible header ID
Error list: keyboard and program lock	
26	Keyboard locked
27	Programming locked
28	Write error in the ser. EEPROM (calib)
29	Hardware error: MANUAL + AUTO locked

3 Modbus protocol description

Error code	Description
30	Edit is inadmissible when the program is active
31	Copy is inadmissible when the program is active
32	MANUAL is inadmissible during AUTO lead time
33	Segment change Image update required
34	No DB number, image update by PLC
35	No DB number for process values of PLC
36	Printer loaded or not operational
37	Setpoint value 1 was not programmed
38	Configure printer (config./interface)
39	Only possible, when the device is in MANUAL mode
40	Self-optimization already running
41	Time axis elapsed or not programmed
42	Time axis cannot be copied
43	Time axis not available
44	Program change is locked
45	MANUAL mode locked
46	Program start locked
Error list: interface processing	
47	Incorrect response length
48	Time-out error (no response)
49	Error reported in telegram protocol
50	Checksum error
51	Parity error
52	Framing error
53	Interface buffer full
54	Address 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	Queue memory does not exist
71	Message queue cannot be opened

3 Modbus protocol description

Error code	Description
72	Message pool cannot be generated
73	Memory from message pool cannot be inquired
74	Message cannot be transmitted
Error list: processing MQX functions	
80	Task creation failed
81	Hardware-Timer not created
Error list: flash processing	
90	Data flash write error
Error list: other errors	
100	Undefined error
101	Division by zero
102	RAM cannot be detected
103	RTC run-time overrun
104	ID does not exist
105	Index too large (overflow)
106	Invalid data
107	Invalid parameter
109	String without 0 characters
110	Time-out during initialization
111	Value must not be written to
112	Log entry with error bits initiating debug mode
Error list: e-mail dispatch via modem and Ethernet	
120	Step error in the status automatic
121	Invalid response length
122	No CONNECT from the modem
123	FCS checksum incorrect
124	Unexpected value or response
125	Conf-Request not accepted
126	No Conf-Request from the other end
127	No Chap-Request from the other end
128	Response time-out
129	Unknown modem response
130	Unexpected OK by the modem
131	Unexpected CONNECT from the modem
132	Unknown frame received
133	Unexpected PROTOCOL by the modem
134	Unexpected COMPRESS from the modem
135	Invalid PPP package received

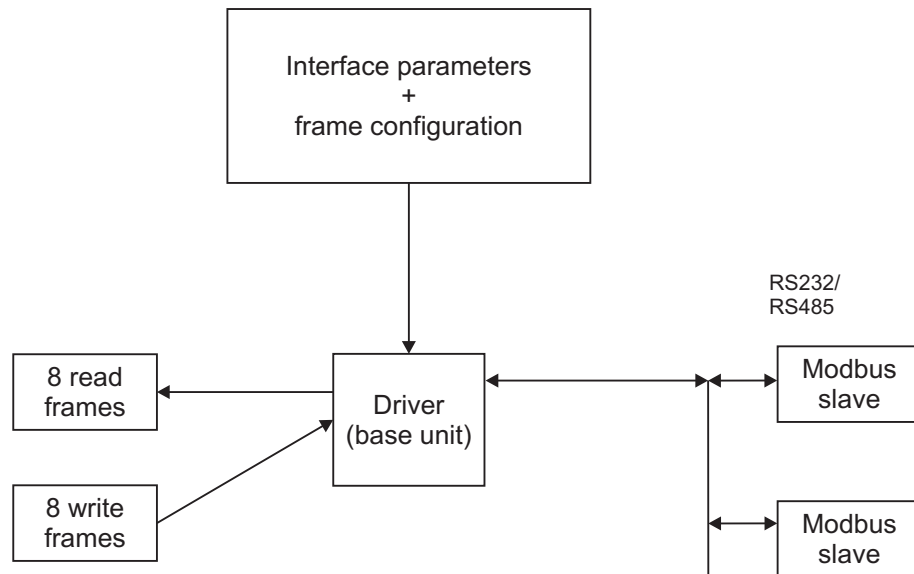
3 Modbus protocol description

Error code	Description
136	Unexpected BUSY from the modem
137	Unknown authentication protocol
138	Ignored LCP option
139	Unexpected DELAYED from the modem
140	Unexpected NODIALTONE
141	Unknown PPP protocol
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	As a client DNS request received
151	Unknown DNS error
152	DNS response is divided
153	No IP received via DNS
154	Unknown UDP port
155	TCP checksum incorrect
156	TCP port incorrect
157	Unknown TCP-SYN option
158	Unused TCP port
159	Unknown POP3 response
160	Unknown SMTP response
161	Unknown DNS name
162	No MD5 requested from CHAP
163	Authentication error
164	Cancel from other end
165	Error when creating TCP socket
166	Error when binding TCP socket
167	Error on TCP connect
168	Error when transmitting TCP telegram
169	Error when closing TCP socket
170	Error on TCP listing
171	Reset on TCP accept
172	Error on TCP accept
173	SMTP server indicates syntax error

3 Modbus protocol description

Error code	Description
174	TCP socket is already closed
175	Incorrect frame configuration
Error list: file system processing	
200	Error when installing the partition manager
201	Error when installing the MFS file system
202	Error when deinstalling the partition manager
203	Error when deinstalling the MFS file system

4.1 Modbus master operation via serial interface



If the central processing unit has been configured as a master, it can send requests on the bus to slaves. Complete frames are always transmitted here. 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 request 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 maximum 8 frames for writing. The enabled frames are cyclically processed consecutively. This also applies to writing frames regardless of whether a process value is 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.



NOTE!

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 time-out 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 used to read in variables.

Number of external inputs during Modbus master operation

If Modbus master operation is selected, the number of input and outputs is limited to

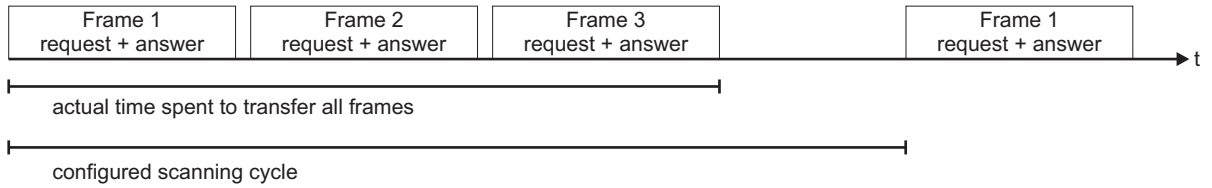
- 64 float, integer, and binary values and
- 9 texts (27 texts as of system version 02; 90 texts as of system version 04).

The external texts can be read into the central processing unit and provided to the optional multifunction panel as a batch text.

4 Serial transmission modes

Chronological sequence

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

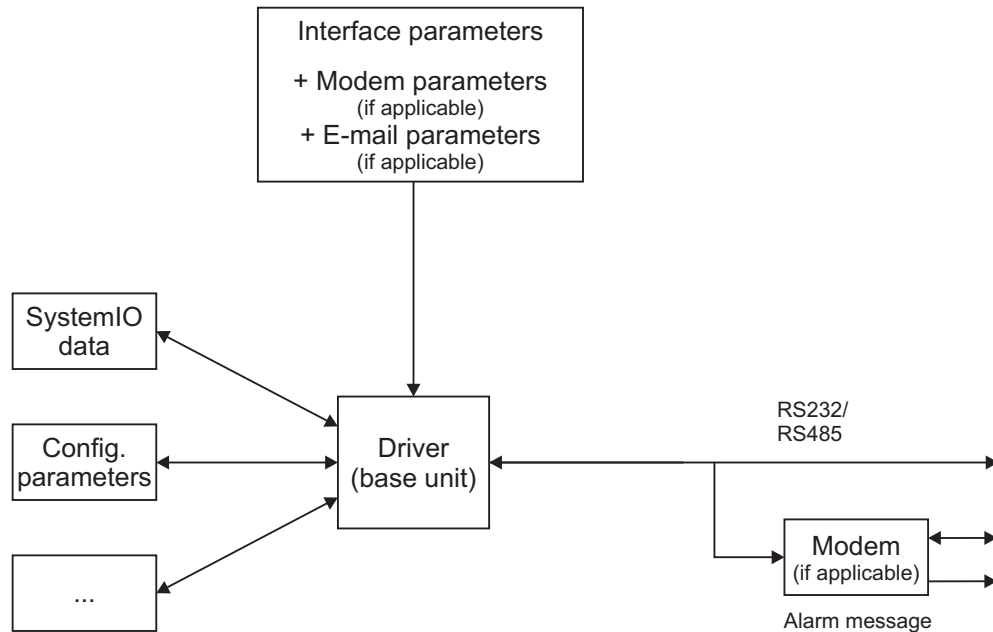


NOTE!

The actual time it takes to transmit all frames can be read under the Modbus addresses 0x00EC and 0x010C (serial) and under the Modbus addresses 0x009D, 0x00B0, 0x00C3, and 0x00D6 (Modbus/TCP).

If the configured scanning cycle is greater than the elapsed actual time required for scanning, the central processing unit 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.

4.2 Modbus slave operation via serial interface



If the central processing unit has been configured as a slave, it responds to Modbus requests from the master in the network. The master controls the data exchange and the slaves only have a response function. They are identified by means of their device address.

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

⇒ Chapter 7 "Modbus address tables", page 79

During slave operation, the central processing unit can operate as several virtual devices with several device addresses.

⇒ Chapter 5.2 "Networking with Modbus/TCP", page 59

All modules connected to the central processing unit can be addressed with their respective device addresses that were assigned in the setup program under **HARDWARE ARRANGEMENT**. This makes it possible to always use the same Modbus addresses for the same type of modules.

4 Serial transmission modes

4.3 PLC access to the serial interface

4.3.1 Activation

The PLC can access the serial interface(s). To do this, the ComSetSettings() initialization function must be used.

Once the PLC has programmed its parameters using ComSetSettings(), a new protocol ("PLC-controlled") is activated. The previous parameters are locally overwritten here. The new protocol and where applicable, the changed parameters cannot be viewed in the device-side configuration; the setup program still displays the previous parameters.

To reactivate the previous parameters, the PLC must end its access using ComClose(). Alternatively, changing the device-side configuration can restore access to the interface on the device; the corresponding protocol is activated and the configured parameters are used here.

If the PLC no longer has interface access after a change on the device-side, it receives the error code ERR_NET_NOTCONNECTED with the ComRead() or ComWrite() functions. If the PLC still attempts to establish access, it must call the ComSetSettings() function again in this case, since the parameters that were most recently set are valid.

4.3.2 API functions

The following API functions are provided to the user:

- RTS_HANDLE CDECL SysComOpen(short sPort, RTS_RESULT *pResult)
= request interface handler
- RTS_HANDLE CDECL SysComOpen2(COMSettings *pSettings, COMSettingsEx *pSettingsEx, RTS_RESULT *pResult)
= request interface handler and simultaneously call up ComSetSettings
- RTS_RESULT CDECL SysComSetSettings(RTS_HANDLE hCom, COMSettings *pSettings, COMSettingsEx *pSettingsEx)
= obtain interface access and configure interface parameter
- RTS_RESULT CDECL SysComGetSettings (RTS_HANDLE hCom, COMSettings *pSettings, COMSettingsEx *pSettingsEx)
= read out interface parameter
- unsigned int CDECL SysComRead(RTS_HANDLE hCom, unsigned char *pbyBuffer, unsigned int uiSize, unsigned long ulTimeout, RTS_RESULT *pResult)
= request receiver
- unsigned int CDECL SysComWrite(RTS_HANDLE hCom, unsigned char *pbyBuffer, unsigned int uiSize, unsigned long ulTimeout, RTS_RESULT *pResult)
= start transmission
- RTS_RESULT CDECL SysComSetTimeout(RTS_HANDLE hCom, unsigned long ulTimeout)
= set character time-out
- RTS_RESULT CDECL SysComClose(RTS_HANDLE hCom)
= return interface to the original protocol

All API functions return a 3S error code in RTS_RESULT, which must be evaluated by the PLC.



NOTE!

These API functions are based on the standard CODESYS functions (SysCom library). The individual parameters and error codes are described in the PLC online help.

4.3.3 End identifier conditions

The 3S definitions contain all the necessary end identifier conditions, which are required for building a package when receiving individual characters:

- Length in bytes
is transmitted as parameter *uiSize* for *ComRead()*
= max. buffer size to disable the length as the end condition
- Response time-out in milliseconds
is transmitted as parameter *ulTimeout* for *ComRead()*
= "SYS_NOWAIT" = 0 (endless) to disable the time-out
- Character time-out in milliseconds
is transmitted in parameter *pSettings* for *ComSetSettings()* or as parameter *ulTimeout* for *ComSetTimeout()*
= "SYS_NOWAIT" = 0 (endless) to disable the time-out

Complete packages are already transmitted character by character in the background during transmission. The *ComWrite()* function immediately ends without blocking (parameter *ulTimeout* is ignored).

4.3.4 Protocol scenarios

All usual applications can be covered with the defined end identifier conditions.

PLC as a master

A master starts with the transmission and then waits for a limited time (response time-out) for the response of the addressed slave.

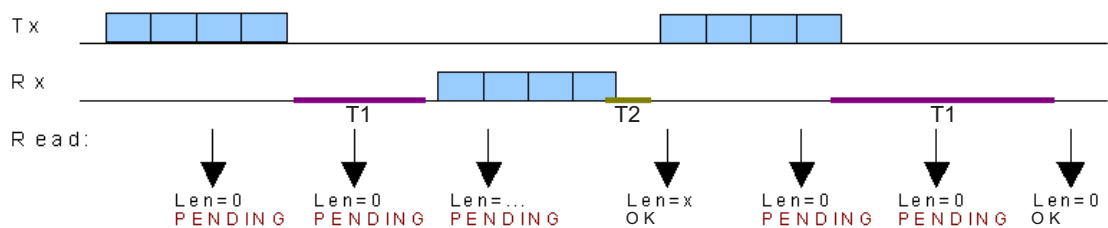
The parameters for the end identifier conditions must be set as follows:

possibly: length = x bytes

response time-out T1 = 2000 ms (example)

possibly: character time-out T2 = y ms

One of the two conditions – length or character time-out – must be set so that a started response package can be detected as completed.



4 Serial transmission modes

PLC as a slave

A slave continuously waits for reception and only transmits if it is requested to do so by the master.

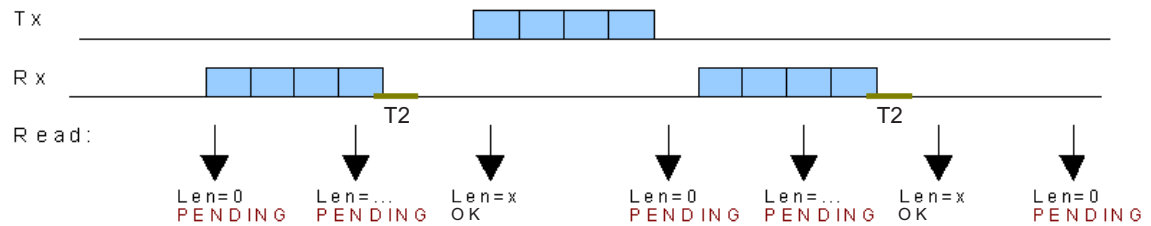
The parameters for the end identifier conditions must be set as follows:

possibly: length = x bytes

response time-out = 0 ms (endless)

possibly: character time-out T2 = y ms

One of the two conditions – length or character time-out – must be set so that a started receiving package can be detected as completed.



PLC as a unidirectional slave (barcode)

When connecting a barcode scanner, the slave also continuously waits for reception while the transmission direction is always disabled.

In this case, the correct package formation is particularly important: if a label is read, its content must be transmitted in a single string. To do this, the start and end of the text must be reliably detected. Character time-outs of 75 ms are usual.

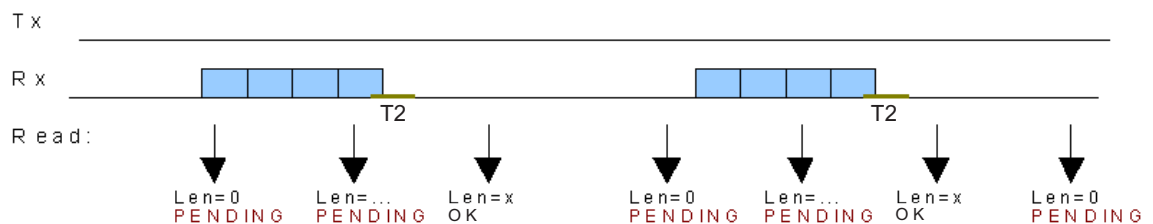
The parameters for the end identifier conditions must be set as follows:

possibly: length = x bytes

response time-out = 0 ms (endless)

possibly: character timeout T2 = y ms

One of the two conditions – length or character time-out – must be set so that a started receiving package can be detected as completed.



4.4 RS232 and RS422/485

The serial interface is implemented using two different additional cards for the central processing unit. The type of the used interface board, RS232 or RS422/485, is automatically detected by the central processing unit via the hardware ID.

In Modbus slave operation, these interfaces can be used to connect visualization software and also for Modem connections.

External process values can be read in during Modbus master operation (no modem operation). This way, you can increase the number of input values.

4 Serial transmission modes

5.1 Modbus/TCP

Modbus/TCP uses Ethernet as a transmission standard. Two transmission options can be used here:

- Modbus/TCP slave for transmitting individual values
- Modbus/TCP master for transmitting entire data frames

The advantage of using Modbus/TCP and the Ethernet interface is in the high speed and the company-wide availability of the connected devices.

Modbus/TCP is a standardized process in which a Modbus telegram is packaged (tunneled) into a TCP frame via Ethernet.

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

Structure of a Modbus/TCP telegram

MBAP header				Modbus telegram
2-byte transaction ID	2-byte protocol ID	2-byte length	1-byte unit ID	Other bytes as displayed below but without CRC
Identical in request and response	Must be 0 for Modbus	Length of request or response in bytes starting with (incl.) "unit ID"	Corresponds to device address and must be TCP 0xFF or 0 (0 = broadcast)	

For comparison: the "normal" Modbus telegram

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

This protocol can be used, e.g. by a suitable process data visualization program, to read and write system values via a company-wide Ethernet network. All device variables from the Modbus address tables can be accessed.

⇒ Chapter 7 "Modbus address tables", page 79

5 Ethernet transmission modes

Example: reading n words

Read the IP address of the central processing unit. Here in the example, it is the address 10.10.1.69. Since each IP address range is stored in a word, 4 words (8 bytes) must be read out.

See also Modbus example in Chapter 3.6.2 "Reading n words", page 33.

Request:

MBAP header				Modbus telegram (without slave address and CRC)		
00 01	00 00	00 06	FF	03	10 6D	00 04
2-byte transaction ID	2-byte protocol ID	2-byte length	1-byte unit ID	1-byte function code	2-byte address first word	2-byte number of words
Response assignment to the request (consecutive numbering)	Always 0x00 with Modbus	Length of the request in bytes starting with (incl.) "unit ID"; 6 bytes here (0x06)	Always 0xFF with TCP (except for broadcast)	Function code for "reading n words"	First word of the IP address to be read	4 words are to be read

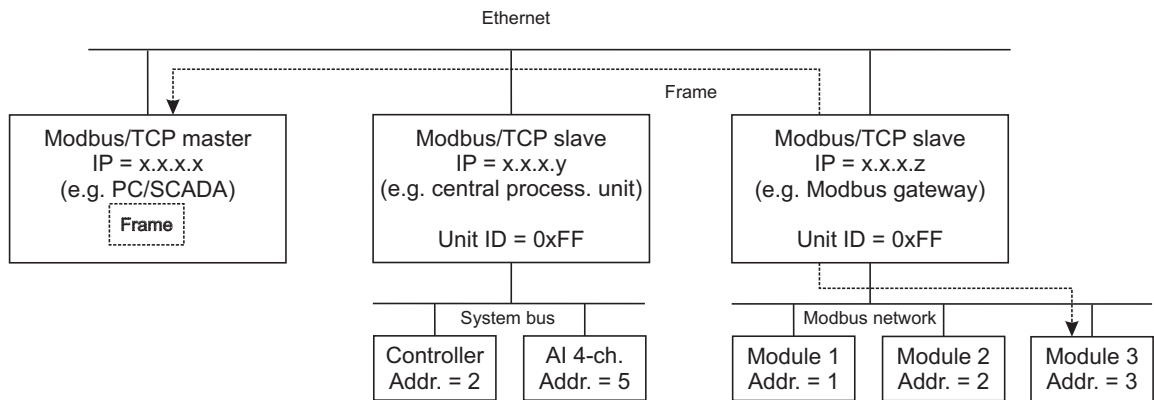
Response:

MBAP header				Modbus telegram (without slave address and CRC)					
00 01	00 00	00 0B	FF	03	08	00 0A	00 0A	00 01	00 45
2-byte transaction ID	2-byte protocol ID	2-byte length	1-byte unit ID	1-byte function code	1-byte number of bytes read	8-byte data read			
Response assignment to the request (consecutive numbering)	Always 0x00 with Modbus	Length of the response in bytes starting with (incl.) "unit ID"; 11 bytes here (0x0B)	Always 0xFF with TCP (except for broadcast)	Function code for "reading n words"	8 bytes were read	IP address consisting of 4 words (8 bytes): 10. 10. 1. 69			

5 Ethernet transmission modes

5.2 Networking with Modbus/TCP

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

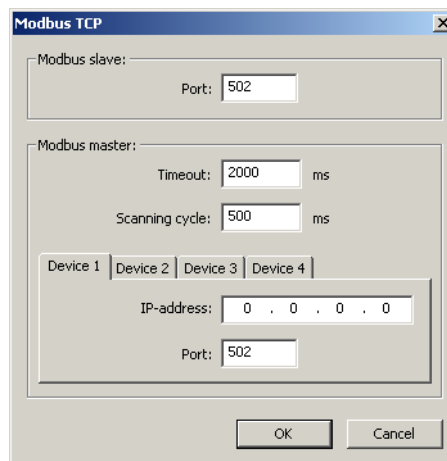


Each Modbus/TCP node can be uniquely addressed by an IP address. The "unit ID" (= device address) also contained in the protocol must be the node 0xFF directly connected to the network. A node, however, can virtually operate as several lower-level devices and therefore also the central processing unit (center of image) for all connected modules. They can be accessed with device addresses 1 to 254 (as assigned in the setup program under "**HARDWARE ARRANGEMENT**").

5.3 Modbus master with Modbus/TCP

Up to 4 connection are possible to different Modbus/TCP devices which allow the exchange of maximum 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 "Serial transmission modes", page 49

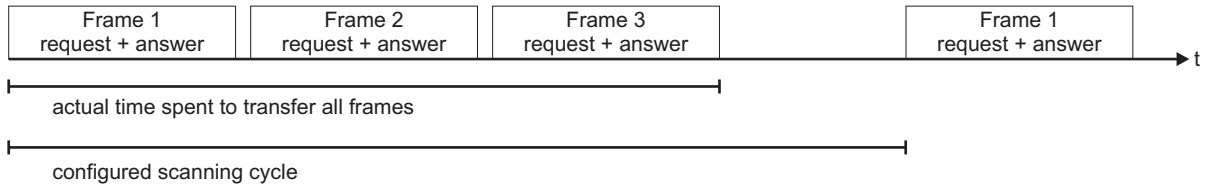


Frames can be exchanged with 4 devices (1 to 4) on the other end using 4 sockets. The Modbus device address that can be edited inside each frame mask and that is also transmitted in the Modbus/TCP protocol can be used to address Modbus gateways and modules behind other basic modules. The latter then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

5 Ethernet transmission modes

Chronological sequence

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



NOTE!

The actual time it takes to transmit all frames can be read under the Modbus addresses 0x00EC and 0x010C (serial) and under the Modbus addresses 0x009D, 0x00B0, 0x00C3, and 0x00D6 (Modbus/TCP).

If the configured scanning cycle is greater than the elapsed actual time required for scanning, the central processing unit 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, the central processing unit serves as a slave and is available for requests on the bus from the Modbus master. A master can request all device variables according to the Modbus address tables of this slave.

⇒ Chapter 7 "Modbus address tables", page 79



NOTE!

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

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

5.5 HTTP

In this case, the central processing unit is configured as a slave and handles incoming requests as a server via port 80. These requests, for example, can be carried out by a PC with setup program, PC evaluation software (PCA), or PCA communication software (PCC).

5.6 Browser connection and Web server



The central processing unit can also be accessed by a browser using the HTTP protocol. The required URL for this is the IP address of the central processing unit (in the above example 10.13.6.177). 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 the central processing unit via the setup program **SETUP NAME > CPU > CONFIGURATION LEVEL > 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.



NOTE!

Information on configuring the Web server functionality can be found in the central processing unit's operating manual B 705001.0.

5 Ethernet transmission modes

5.7 E-mail (SMTP and POP3)

The central processing unit 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)¹ as well as POP3 servers at the standard port (110)

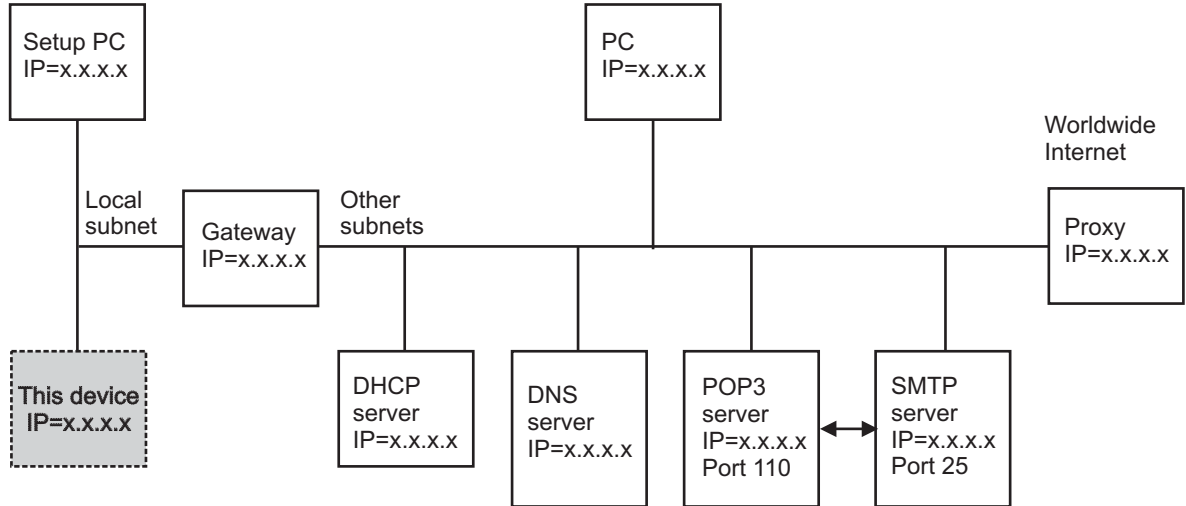


Figure 1: Typical networking in the company network

Function of the individual nodes

Gateway:

separates local sub-nets from each other and thus filters the packets. Not all packets are received in every sub-net. Packets from outside the local sub-net must be addressed to the gateway.

DHCP server:

can automatically assign an IP address, sub-net mask, and gateway address to other stations 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 request "www.name.de" will generate the "www.name.de has IP=10.12.32.45" response.

POP3 server:

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

SMTP server:

serves to transmit e-mails. The authorization to transmit e-mails via a mail account must be released in several networks by previously logging in at the corresponding POP3 server.

Proxy:

¹ As of system version 05: SMTP port number changeable

5 Ethernet transmission modes

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 "once-only" IP addresses (used in 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 masks can be accessed via **CPU > SETUP ONLY > E-MAIL**.

E-mail

Send via: Ethernet

E-mail server: ... Can only be altered by trained personnel!

E-mail 1 | E-mail 2 | E-mail 3 | E-mail 4 | E-mail 5

E-mail addresses:

1:

2:

3:

Caution: Characters other than A..Z, a..z, 0..9, -, _, @, ., may cause problems!

Alarm signal:

Inactive

Subject:

Mail Betreff 01

Contents:

Mail Text 01

OK Cancel

E-mail server

Authentication: SMTP-After-POP

User name: Your user name

Password: Your password

POP3 server: pop3.example.net

SMTP server: smtp.example.net

Mail sender: geraet@example.net

Caution: Characters other than A..Z, a..z, 0..9, -, _, @, ., may cause problems!

OK Cancel



NOTE!

When "transmitting via: Ethernet", a mail server that is in the company network (not accessible in the Internet or without proxy addressing) must be entered. This mail server should also be able to transmit E-mails to the Internet.



NOTE!

When "transmitting via: modem", the modem establishes a direct connection to the Internet and a mail server located on the Internet must be entered.

5 Ethernet transmission modes



NOTE!

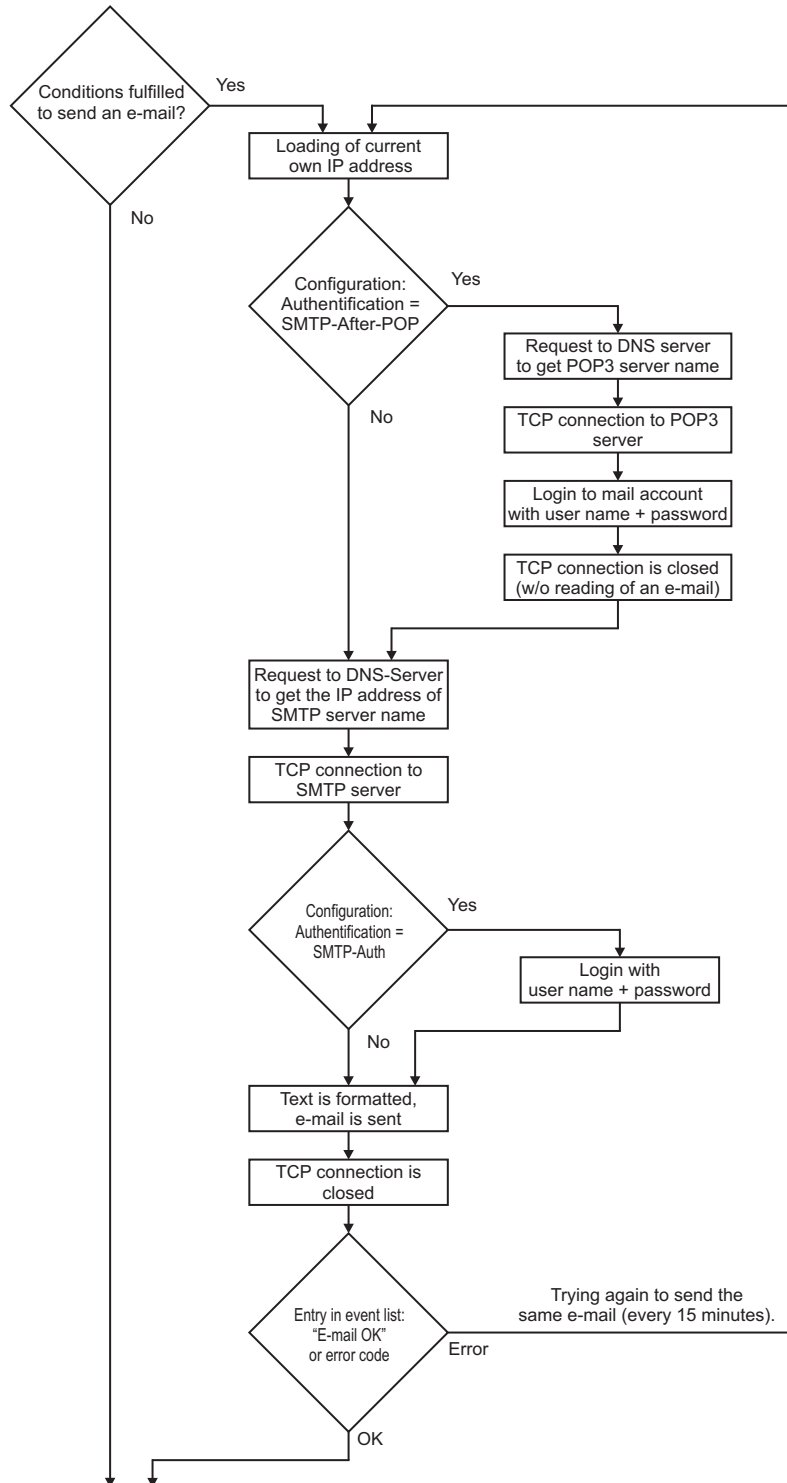
Information on configuring the e-mail functionality can be found in the central processing unit's operating manual B 705001.0.

Transmitting 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 174) 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 3.9.3 "Error codes as integer return values", page 44

5 Ethernet transmission modes



5 Ethernet transmission modes

6.1 General information

The central processing unit 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 request and response.

The maximum size of a frame is defined to 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.

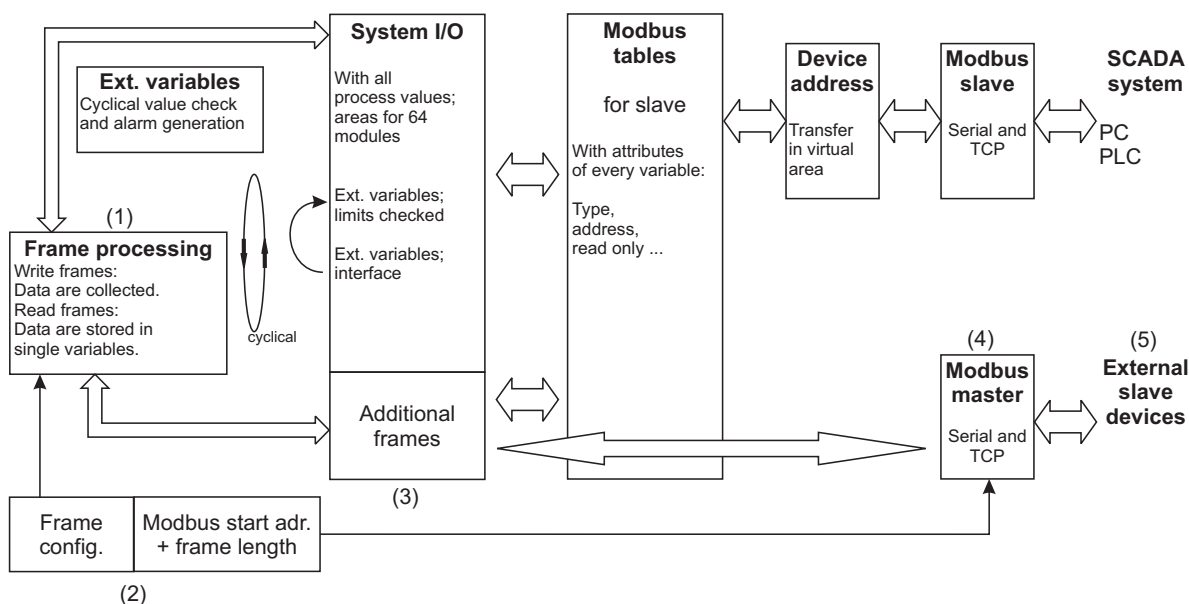


NOTE!

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 the other end.
(5)	The other end receives this frame and responds.

6 User frames



NOTE!

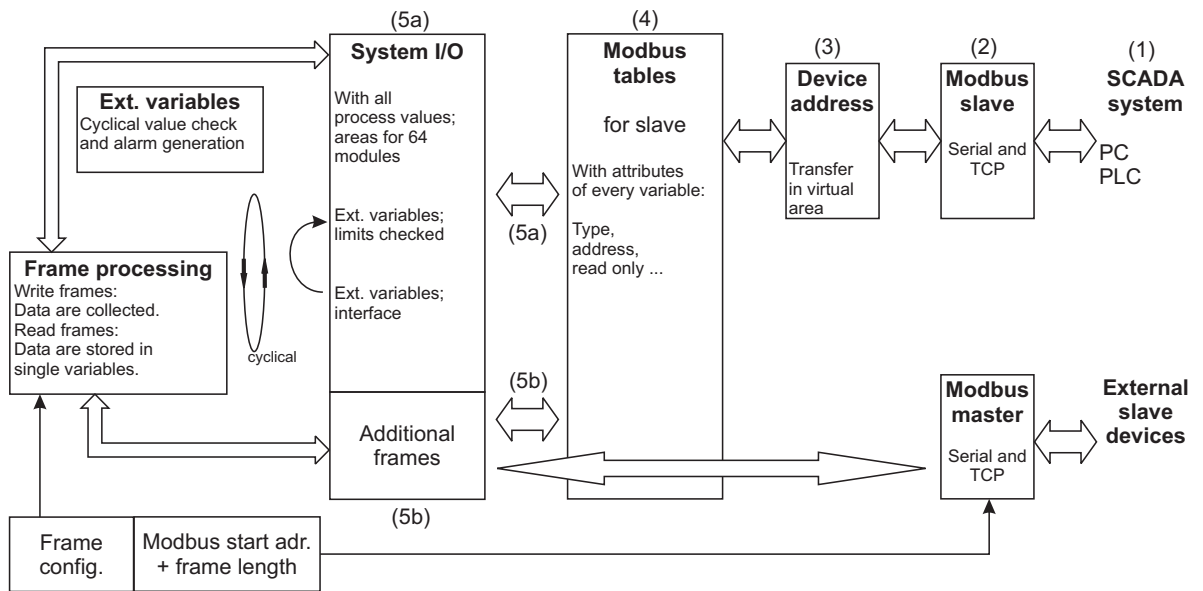
When transmitting as a Modbus master, repetitive errors are entered in the event list. Corresponding error codes can be found in chapter 3.9.3 "Error codes as integer return values".

Error monitoring

Each frame is monitored for data transmission. If an error occurs, the corresponding error flag is set, an error code is also saved 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 makes a request (reading or writing).
(2)	The interface receives the request.
(3)	The corresponding Modbus table is used based on the device address.
(4)	The Modbus table assigns the Modbus addresses to the device variables.
(5a)	The variable value is accessed and the Modbus response is generated.
(5b)	The frames configured for the Modbus master can also be accessed for checking or variable scaling/type conversions.

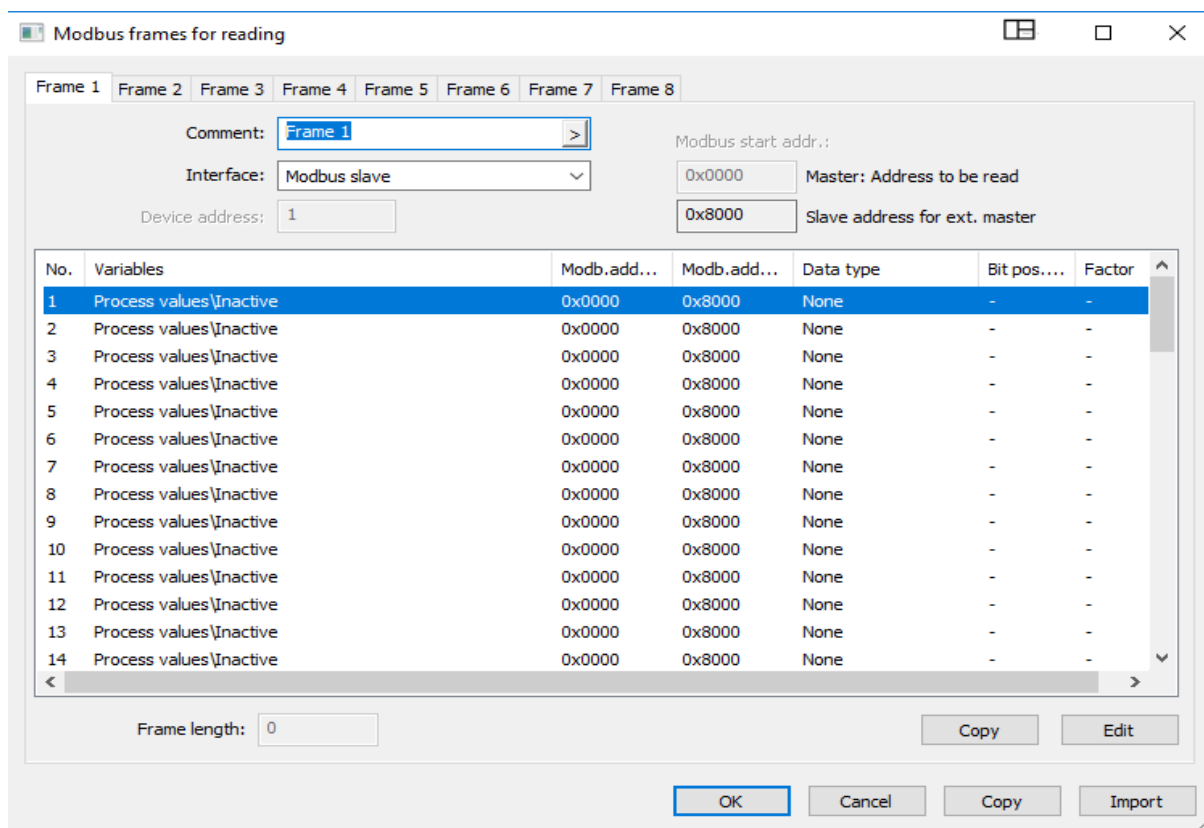
6.3 Compiling Modbus frames

The "Modbus frames for reading" and "Modbus frames for writing" options are found under **SETUP NAME > CPU > 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 side. 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 transferred in a Modbus telegram.

Setup dialog



Parameter

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 requests as a slave. In 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 requests 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 serial 2	Modbus master; Modbus RTU via Com1 Modbus master; Modbus RTU via Com2
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 (central processing unit 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) of the central processing unit if this frame is to be requested 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 marking and then confirm using the "Edit" button).	
Frame length	0 to 254	Displays the frame length (byte(s))

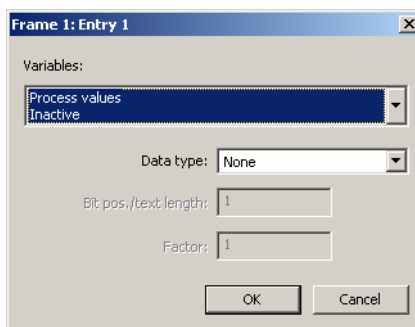
Copy (entries)

With this function entries within a Modbus frame can be copied.

As of system version 05 entries of a frame can be copied into other frames as well.

Editing

To open this window, use the "Edit" button:



Parameter

Parameter	Selection/settings	Description
Variables	Analog, digital, integer, and text variables	
	Inactive	No variable selected
	Select variable	Selector for selecting a variable
Data type	The data type depends on the type of the external input preset in the setup program. It can then 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 float 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
Bitpos./text length	Integer (1 byte): 0 to 7	The setting range of the bit position or text lengths depends on the selected data type. (This parameter is not active for the "float" data type.)
	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	

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 float range allowed, default value = 1.0	This factor is used to rescale values during the transmission, in particular, with the same type conversions. (This parameter is not active with the "text" data type.)

Copy (frames)

With this function Modbus frames within a project can be copied.

Import (as of system version 05)

With this function Modbus frames can be imported from another project. First, the setup file from which you want to import is temporarily opened. Then all frames to be copied have to be selected (individual frames or all). As a result, all entries of the relevant frames are transferred into your own project.

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 side. 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 transferred in a Modbus telegram.

Setup dialog

Parameter

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 requests as a slave. In 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 requests 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 Com1
	Modbus master serial 2	Modbus master; Modbus RTU via Com2
Device address	0 to 1 to 255	Device address of the external device (Modbus slave).

6 User frames

Parameter	Selection/settings	Description
Modbus start address (master)	0x0000 to 0xFFFF	Modbus start address (offset) of the external slave device (central processing unit 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) of the central processing unit if this frame is to be requested 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 marking and then confirm using 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 an error occurs (for measured values in the float format).

Replace (as of system version 05)

With this function the modules within the configured process value paths can be exchanged. The prerequisite is that several modules of the same type must exist in the project.

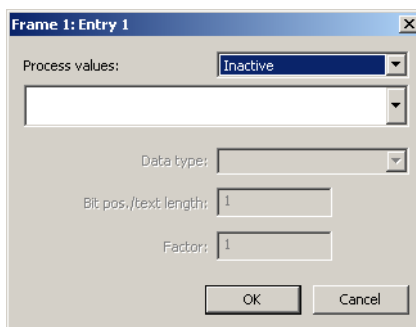
Copy (entries)

With this function entries within a Modbus frame can be copied.

As of system version 05 entries of a frame can be copied into other frames as well.

Editing

To open this window, use the "Edit" button:



Parameter

Parameter	Selection/settings	Description
Process value	Analog, integer, and digital signals and texts of the input/output modules and the central processing unit (incl. variables and PLC)	
	Inactive Select module (process value source)	No module selected The process value of the relevant module can be selected from a list in the following field.
Data type	The data type depends on the type of the process value preset in the setup program. It can then 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 float 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
Bitpos./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 range of the bit position or text lengths depends on the selected data type. (This parameter is not active for the "float" 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 float range allowed, default value = 1.0	This factor is used to rescale values during the transmission, in particular, with the same type conversions. (This parameter is not active with the "text" data type.)

Copy (frames)

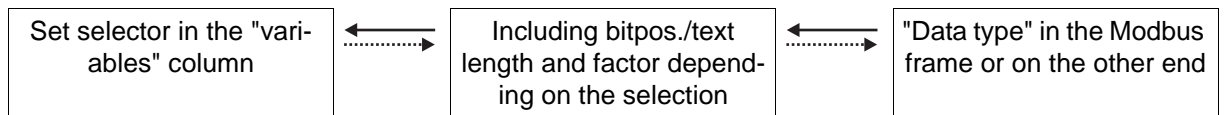
With this function Modbus frames within a project can be copied.

Import (as of system version 05)

With this function Modbus frames can be imported from another project. First, the setup file from which you want to import is temporarily opened. Then all frames to be copied have to be selected (individual frames or all). As a result, all entries of the relevant frames are transferred into your own project.

6.4 Examples for the data transmission options with frames

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



NOTE!

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 include:

Float value	←→	x Factor	←→	Float value
Float value	←→	x Factor	←→	Integer value
Integer value	←→	x Factor	←→	Float 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



NOTE!

These tables are important for external devices that access the central processing unit as a Modbus master (configured as a Modbus slave). Alternatively, external devices can also access the central processing unit using the Modbus frames.

7.1 Data types and access types

Data types

Bit x	Bit no. x (bit 0 is the lowest bit)
Boolean	Boolean value (TRUE or FALSE) can be read or written as a word. The value range is 0 to 1.
Byte	1 byte = 8 bi, 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
Char[60]	Text with 60 characters, each with 2 characters in a word
Bit field 32	Bit field 32 bits long

Access types

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



CAUTION!

Write operations to some R/W parameters result in them being saved to the EEPROM or Flash memory.

The storage components have only a limited number of writing cycles (approx. 10,000 or 100,000), which is why no fast cyclical writing operations should be performed, since there is otherwise a risk of a storage error in case of a power failure.

7.2 Central processing unit addresses

The following tables contain specifications of all process and device data for the central processing unit including their address, data type, and type of access.

Central processing unit - cycle time and device name

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x000A	10	Long	R/O	Software version
0x009D	157	Uint32	R/O	Modbus master TCP device 1: actual cycle time in 5 ms ticks
0x00B0	176	Uint32	R/O	Modbus master TCP device 2: actual cycle time in 5 ms ticks

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x00C3	195	Uint32	R/O	Modbus master TCP device 3: actual cycle time in 5 ms ticks
0x00D6	214	Uint32	R/O	Modbus master TCP device 4: actual cycle time in 5 ms ticks
0x00EC	236	Uint32	R/O	Com1: for Modbus master: actual cycle time in 5 ms ticks
0x010C	268	Uint32	R/O	Com2: for Modbus master: actual cycle time in 5 ms ticks
0x1000	4096	Char[60]	R/O	Configuration: device name

Central processing unit process values

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x106D	4205	Byte	R/O	Ethernet IP address 1st byte
0x106E	4206	Byte	R/O	Ethernet IP address 2nd byte
0x106F	4207	Byte	R/O	Ethernet IP address 3rd byte
0x1070	4208	Byte	R/O	Ethernet IP address 4th byte
0x1071	4209	Float	R/W	Analog variable 1
0x1073	4211	Float	R/W	Analog variable 2
0x...	...	Float	R/W	Analog variable ...
0x10EF	4335	Float	R/W	Analog variable 64
				As of system version 05:
0x6B4E	27470	Float	R/W	Analog variable 65
0x6B50	27472	Float	R/W	Analog variable 66
0x...		Float	R/W	Analog variable ...
0x6BCC	27596	Float	R/W	Analog variable 128
0x1171	4465	Boolean	R/O	Alarm 1 analog variable 1
0x1172	4466	Boolean	R/O	Alarm 1 analog variable 2
0x...	...	Boolean	R/O	Alarm 1 analog variable ...
0x11B0	4528	Boolean	R/O	Alarm 1 analog variable 64
				As of system version 05:
0x6C4E	27726	Boolean	R/O	Alarm 1 analog variable 65
0x6C4F	27727	Boolean	R/O	Alarm 1 analog variable 66
0x...	...	Boolean	R/O	Alarm 1 analog variable ...
0x6C8D	27789	Boolean	R/O	Alarm 1 analog variable 128
0x11B1	4529	Boolean	R/O	Alarm 2 analog variable 1
0x11B2	4530	Boolean	R/O	Alarm 2 analog variable 2
0x...	...	Boolean	R/O	Alarm 2 analog variable ...

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x11F0	4592	Boolean	R/O	Alarm 2 analog variable 64
				As of system version 05:
0x6C8E	27790	Boolean	R/O	Alarm 2 analog variable 65
0x6C8F	27791	Boolean	R/O	Alarm 2 analog variable 66
0x...	...	Boolean	R/O	Alarm 2 analog variable ...
0x6CCD	27853	Boolean	R/O	Alarm 2 analog variable 128
0x11F1	4593	Int32	R/W	Integer variable 1
0x11F3	4595	Int32	R/W	Integer variable 2
0x...	...	Int32	R/W	Integer variable ...
0x126F	4719	Int32	R/W	Integer variable 64
				As of system version 05:
0x6CCE	27854	Int32	R/W	Integer variable 65
0x6CD0	27856	Int32	R/W	Integer variable 66
0x...	...	Int32	R/W	Integer variable ...
0x6D4C	27980	Int32	R/W	Integer variable 128
0x12F1	4849	Boolean	R/O	Alarm 1 integer variable 1
0x12F2	4850	Boolean	R/O	Alarm 1 integer variable 2
0x...	...	Boolean	R/O	Alarm 1 integer variable ...
0x1330	4912	Boolean	R/O	Alarm 1 integer variable 64
				As of system version 05:
0x6DCE	28110	Boolean	R/O	Alarm 1 integer variable 65
0x6DCF	28111	Boolean	R/O	Alarm 1 integer variable 66
0x...	...	Boolean	R/O	Alarm 1 integer variable ...
0x6E0D	28173	Boolean	R/O	Alarm 1 integer variable 128
0x1331	4913	Boolean	R/O	Alarm 2 integer variable 1
0x1332	4914	Boolean	R/O	Alarm 2 integer variable 2
0x...	...	Boolean	R/O	Alarm 2 integer variable ...
0x1370	4976	Boolean	R/O	Alarm 2 integer variable 64
				As of system version 05:
0x6E0E	28174	Boolean	R/O	Alarm 2 integer variable 65
0x6E0F	28175	Boolean	R/O	Alarm 2 integer variable 66
0x...	...	Boolean	R/O	Alarm 2 integer variable ...
0x6E4D	28237	Boolean	R/O	Alarm 2 integer variable 128

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x1371	4977	Boolean	R/W	Digital variable 1
0x1372	4978	Boolean	R/W	Digital variable 2
0x...	...	Boolean	R/W	Digital variable ...
0x13B0	5040	Boolean	R/W	Digital variable 64
				As of system version 05:
0x6E4E	28238	Boolean	R/W	Digital variable 65
0x6E4f	28239	Boolean	R/W	Digital variable 66
0x...	...	Boolean	R/W	Digital variable ...
0x6E8D	28301	Boolean	R/W	Digital variable 128
0x13F1	5105	Char[64]	R/W	External batch text 1
0x1411	5137	Char[64]	R/W	External batch text 2
0x...	...	Char[64]	R/W	External batch text ...
0x14F1	5361	Char[64]	R/W	External batch text 9
				As of system version 02:
0x24B6	9398	Char[64]	R/W	External batch text 10
0x24D6	9430	Char[64]	R/W	External batch text 11
0x...	...	Char[64]	R/W	External batch text ...
0x26D6	9942	Char[64]	R/W	External batch text 27
				As of system version 04:
0x2936	10550	Char[187]	R/W	External batch text 1
0x2994	10644	Char[187]	R/W	External batch text 2
0x...	...	Char[187]	R/W	External batch text ...
0x49E4	18916	Char[187]	R/W	External batch text 90
0x1631	5681	Word	R/O	Time: year
0x1632	5682	Word	R/O	Time: month
0x1633	5683	Word	R/O	Time: day
0x1634	5684	Word	R/O	Time: hour
0x1635	5685	Word	R/O	Time: minute
0x1636	5686	Word	R/O	Time: second
0x1637	5687	Uint32	R/W	Start program no. x program generator 1
0x1639	5689	Uint32	R/W	Start program no. x program generator 2
0x...	...	Uint32	R/W	Start program no. x program generator ...
0x1647	5703	Uint32	R/W	Start program no. x program generator 9

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x1649	5705	Uint32	R/W	Start section no. x program generator 1
0x164B	5707	Uint32	R/W	Start section no. x program generator 2
0x...	...	Uint32	R/W	Start section no. x program generator ...
0x1659	5721	Uint32	R/W	Start section no. x program generator 9
0x165B	5723	Uint32	R/W	Start lead time of program generator 1
0x165D	5725	Uint32	R/W	Start lead time of program generator 2
0x...	...	Uint32	R/W	Start lead time of program generator ...
0x166B	5739	Uint32	R/W	Start lead time of program generator 9
0x166D	5741	Uint32	R/W	Start section of remaining time for program generator 1
0x166F	5743	Uint32	R/W	Start section of remaining time for program generator 2
0x...	...	Uint32	R/W	Start section of remaining time for program generator ...
0x167D	5757	Uint32	R/W	Start section of remaining time for program generator 9
0x167F	5759	Uint32	R/W	Start time for program generator 1
0x1681	5761	Uint32	R/W	Start time for program generator 2
0x...	...	Uint32	R/W	Start time for program generator ...
0x168F	5775	Uint32	R/W	Start time for program generator 9
0x1691	5777	Word	R/W	Command to program generator 1 0 = No command 1 = Stop 2 = Start 3 = Start-up at the current actual value 4 = Pause 5 = Manual mode 6 = Jump to next section 7 = Jump to previous section
0x1692	5778	Word	R/W	Command to program generator 2
0x...	...	Word	R/W	Command to program generator ...
0x1699	5785	Word	R/W	Command to program generator 9



NOTE!

As of system version 04, 90 external batch texts can be transmitted via Modbus. The texts 1 to 27 remain at the previous Modbus addresses. In addition, all 90 texts, including the texts 1 to 27, are transmitted in a separate address space. For the texts 1 to 27 it must be ensured in the application that only one of the two possible addresses is used.

7 Modbus address tables

Central processing unit process values for Modbus error

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x169A	5786	Boolean	R/O	COM1 error state
0x169B	5787	Word	R/O	COM1 last error code
0x169C	5788	Boolean	R/O	COM2 error state
0x169D	5789	Word	R/O	COM2 last error code
0x169E	5790	Boolean	R/O	Modbus/TCP slave 1 error state
0x169F	5791	Word	R/O	Modbus/TCP slave 1 last error code
0x16A0	5792	Boolean	R/O	Modbus/TCP slave 2 error state
0x16A1	5793	Word	R/O	Modbus/TCP slave 2 last error code
0x16A2	5794	Boolean	R/O	Modbus/TCP master 1 error state
0x16A3	5795	Word	R/O	Modbus/TCP master 1 last error code
0x16A4	5796	Boolean	R/O	Modbus/TCP master 2 error state
0x16A5	5797	Word	R/O	Modbus/TCP master 2 last error code
0x16A6	5798	Boolean	R/O	Modbus/TCP master 3 error state
0x16A7	5799	Word	R/O	Modbus/TCP master 3 last error code
0x16A8	5800	Boolean	R/O	Modbus/TCP master 4 error state
0x16A9	5801	Word	R/O	Modbus/TCP master 4 last error code

Central processing unit hardware configuration (current)

Address		Data type/ bit number	Access	Signal designation
Hex	Dec.			
0x16AA	5802	Word	R/O	System bus status (actual state)
0x16AB	5803	Word	R/O	Switch position
0x16AC	5804	Word	R/O	Switch position setup
0x16AD	5805	Word	R/O	Internal switch position
0x16AE	5806	Word	R/O	Target state of PLC switch position
0x16AF	5807	Word	R/O	Actual state of PLC switch position
0x16B0	5808	Word	R/O	PLC switch position in stop
0x16B1	5809	Bit field 16	R/O	Alarm bits
		Bit 0	R/O	Collective alarm/fault
		Bit 1	R/O	Collective alarm/fault with acknowledgement
		Bit 2	R/O	System collective alarm
		Bit 3	R/O	System collective alarm with acknowledgement
		Bit 4	R/O	System fault
		Bit 5	R/O	System fault with acknowledgement
		Bit 6	R/O	Base unit (central processing unit) collective alarm
		Bit 7	R/O	"Run" system state

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex	Dec.			
		Bit 8	R/O	HW configuration conflict
		Bit 9	R/O	Fieldbus error
		Bit 10	R/O	Error of a mandatory module
		Bit 11	R/O	Error of an optional module
		Bit 12	R/O	No PLC program
		Bit 13	R/O	"Stop" system state
		Bit 14	R/O	Battery empty
		Bit 15	R/O	Battery low
0x16B2	5810	Bit field 16	R/O	External alarm bits
		Bit 0	R/O	Collective alarm of Ext. Analog (analog variables)
		Bit 1	R/O	Collective alarm of Ext. Integer (integer variables)
		Bit 2	R/O	Collective alarm of Ext. Digital (digital variables)
0x16B3	5811	Word	R/O	Target state of system bus status
0x16B4	5812	Word	R/O	System bus error

PLC analog outputs

The meaning of PLC analog outputs (18 blocks each with 16 outputs) is determined by the PLC program. They are named in the setup program under **PROJECT NAME > CPU > SETUP ONLY > ANALOG PLC OUTPUT TEXTS**.

PLC process values (name provided at the factory in brackets)

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x16C1	5825	Float	R/O	Block 1 of analog output 1 (channel 1 SP 1)
0x16C3	5827	Float	R/O	Block 1 of analog output 2 (channel 1 SP 2)
0x16C5	5829	Float	R/O	Block 1 of analog output 3 (channel 1 SP 3)
0x16C7	5831	Float	R/O	Block 1 of analog output 4 (channel 1 SP 4)
0x16C9	5833	Float	R/O	Block 1 of analog output 5 (channel 2 SP 1)
0x16CB	5835	Float	R/O	Block 1 of analog output 6 (channel 2 SP 2)
0x16CD	5837	Float	R/O	Block 1 of analog output 7 (channel 2 SP 3)
0x16CF	5839	Float	R/O	Block 1 of analog output 8 (channel 2 SP 4)
0x16D1	5841	Float	R/O	Block 1 of analog output 9 (channel 3 SP 1)
0x16D3	5843	Float	R/O	Block 1 of analog output 10 (channel 3 SP 2)
0x16D5	5845	Float	R/O	Block 1 of analog output 11 (channel 3 SP 3)
0x16D7	5847	Float	R/O	Block 1 of analog output 12 (channel 3 SP 4)
0x16D9	5849	Float	R/O	Block 1 of analog output 13
0x16DB	5851	Float	R/O	Block 1 of analog output 14
0x16DD	5853	Float	R/O	Block 1 of analog output 15

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x16DF	5855	Float	R/O	Block 1 of analog output 16
0x16E1	5857	Float	R/O	Block 2 of analog output 1; see name of block 1 provided by factory
0x16E3	5859	Float	R/O	Block 2 of analog output 2
0x...	...	Float	R/O	Block 2 of analog output ...
0x16FF	5887	Float	R/O	Block 2 of analog output 16
0x1701	5889	Float	R/O	Block 3 of analog output 1; see name of block 1 provided by factory
0x1703	5891	Float	R/O	Block 3 of analog output 2
0x...	...	Float	R/O	Block 3 of analog output ...
0x171F	5919	Float	R/O	Block 3 of analog output 16
0x1721	5921	Float	R/O	Block 4 of analog output 1; see name of block 1 provided by factory
0x1723	5923	Float	R/O	Block 4 of analog output 2
0x...	...	Float	R/O	Block 4 of analog output ...
0x173F	5951	Float	R/O	Block 4 of analog output 16
0x1741	5953	Float	R/O	Block 5 of analog output 1; see name of block 1 provided by factory
0x1743	5955	Float	R/O	Block 5 of analog output 2
0x...	...	Float	R/O	Block 5 of analog output ...
0x175F	5983	Float	R/O	Block 5 of analog output 16
0x1761	5985	Float	R/O	Block 6 of analog output 1; see name of block 1 provided by factory
0x1763	5987	Float	R/O	Block 6 of analog output 2
0x...	...	Float	R/O	Block 6 of analog output ...
0x177F	6015	Float	R/O	Block 6 of analog output 16
0x1781	6017	Float	R/O	Block 7 of analog output 1; see name of block 1 provided by factory
0x1783	6019	Float	R/O	Block 7 of analog output 2
0x...	...	Float	R/O	Block 7 of analog output ...
0x179F	6047	Float	R/O	Block 7 of analog output 16

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x17A1	6049	Float	R/O	Block 8 of analog output 1; see name of block 1 provided by factory
0x17A3	6051	Float	R/O	Block 8 of analog output 2
0x...	...	Float	R/O	Block 8 of analog output ...
0x17BF	6079	Float	R/O	Block 8 of analog output 16
0x17C1	6081	Float	R/O	Block 9 of analog output 1; see name of block 1 provided by factory
0x17C3	6083	Float	R/O	Block 9 of analog output 2
0x...	...	Float	R/O	Block 9 of analog output ...
0x17DF	6111	Float	R/O	Block 9 of analog output 16
0x17E1	6113	Float	R/O	Block 10 of analog output 1
0x17E3	6115	Float	R/O	Block 10 of analog output 2
0x...	...	Float	R/O	Block 10 of analog output...
0x17FF	6143	Float	R/O	Block 10 of analog output 16
0x1801	6145	Float	R/O	Block 11 of analog output 1
0x1803	6147	Float	R/O	Block 11 of analog output 2
0x...	...	Float	R/O	Block 11 of analog output ...
0x181F	6175	Float	R/O	Block 11 of analog output 16
0x1821	6177	Float	R/O	Block 12 of analog output 1
0x1823	6179	Float	R/O	Block 12 of analog output 2
0x...	...	Float	R/O	Block 12 of analog output ...
0x183F	6207	Float	R/O	Block 12 of analog output 16
0x1841	6209	Float	R/O	Block 13 of analog output 1
0x1843	6211	Float	R/O	Block 13 of analog output 2
0x...	...	Float	R/O	Block 13 of analog output ...
0x185F	6239	Float	R/O	Block 13 of analog output 16
0x1861	6241	Float	R/O	Block 14 of analog output 1
0x1863	6243	Float	R/O	Block 14 of analog output 2
0x...	...	Float	R/O	Block 14 of analog output ...
0x187F	6271	Float	R/O	Block 14 of analog output 16
0x1881	6273	Float	R/O	Block 15 of analog output 1

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x1883	6275	Float	R/O	Block 15 of analog output 2
0x...	...	Float	R/O	Block 15 of analog output ...
0x189F	6303	Float	R/O	Block 15 of analog output 16
0x18A1	6305	Float	R/O	Block 16 of analog output 1
0x18A3	6307	Float	R/O	Block 16 of analog output 2
0x...	...	Float	R/O	Block 16 of analog output ...
0x18BF	6335	Float	R/O	Block 16 of analog output 16
0x18C1	6337	Float	R/O	Block 17 of analog output 1
0x18C3	6339	Float	R/O	Block 17 of analog output 2
0x...	...	Float	R/O	Block 17 of analog output ...
0x18DF	6367	Float	R/O	Block 17 of analog output 16
0x18E1	6369	Float	R/O	Block 18 of analog output 1
0x18E3	6371	Float	R/O	Block 18 of analog output 2
0x...	...	Float	R/O	Block 18 of analog output ...
0x18FF	6399	Float	R/O	Block 18 of analog output 16

7 Modbus address tables

PLC integer outputs

The meaning of PLC integer outputs (12 blocks each with 24 outputs) is determined by the PLC program. They are named in the setup program under **CPU > SETUP ONLY > INTEGER PLC OUTPUT TEXTS**.

PLC process values (name provided at the factory in brackets)

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x1901	6401	Int32	R/O	Block 1 of integer 1 (program generator state) 0 = No status 1 = Basic status (program is not running) 2 = Program start 3 = Delay (delay time begins running) 4 = Start-up at the actual value 5 = Automatic (program runs in automatic operation) 6 = Stopped (program stopped during automatic mode) 7 = Manual mode 8 = Program end (program end time expires)
0x1903	6403	Int32	R/O	Block 1 of integer 2 = program number
0x1905	6405	Int32	R/O	Block 1 of integer 3 = Maximum program run time
0x1907	6407	Int32	R/O	Block 1 of integer 4 = program run time
0x1909	6409	Int32	R/O	Block 1 of integer 5 = remaining program time
0x190B	6411	Int32	R/O	Block 1 of integer 6 = delay time
0x190D	6413	Int32	R/O	Block 1 of integer 7 = program channel 1: section number
0x190F	6415	Int32	R/O	Block 1 of integer 8 = program channel 1: max. program run time
0x1911	6417	Int32	R/O	Block 1 of integer 9 = program channel 1: program run time
0x1913	6419	Int32	R/O	Block 1 of integer 10 = program channel 1: max. section run time
0x1915	6421	Int32	R/O	Block 1 of integer 11 = program channel 1: section run time
0x1917	6423	Int32	R/O	Block 1 of integer 12 = program channel 1: remaining section time
0x1919	6425	Int32	R/O	Block 1 of integer 13 = program channel 2: section number
0x191B	6427	Int32	R/O	Block 1 of integer 14 = program channel 2: max. program run time
0x191D	6429	Int32	R/O	Block 1 of integer 15 = program channel 2: program run time
0x191F	6431	Int32	R/O	Block 1 of integer 16 = program channel 2: max. section run time
0x1921	6433	Int32	R/O	Block 1 of integer 17 = program channel 2: section run time
0x1923	6435	Int32	R/O	Block 1 of integer 18 = program channel 2: remaining section time
0x1925	6437	Int32	R/O	Block 1 of integer 19 = program channel 3: section number
0x1927	6439	Int32	R/O	Block 1 of integer 20 = program channel 3: max. program run time
0x1929	6441	Int32	R/O	Block 1 of integer 21 = program channel 3: program run time
0x192B	6443	Int32	R/O	Block 1 of integer 22 = program channel 3: max. section run time
0x192D	6445	Int32	R/O	Block 1 of integer 23 = program channel 3: section run time

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x192F	6447	Int32	R/O	Block 1 of integer 24 = program channel 3: remaining section time
0x1931	6449	Int32	R/O	Block 2 of integer 1 = diverse program data
0x1933	6451	Int32	R/O	Block 2 of integer 2
0x...	...	Int32	R/O	Block 2 of integer ...
0x195F	6495	Int32	R/O	Block 2 of integer 24
0x1961	6497	Int32	R/O	Block 3 of integer 1 = diverse program data
0x1963	6499	Int32	R/O	Block 3 of integer 2
0x...	...	Int32	R/O	Block 3 of integer ...
0x198F	6543	Int32	R/O	Block 3 of integer 24
0x1991	6545	Int32	R/O	Block 4 of integer 1 = diverse program data
0x1993	6547	Int32	R/O	Block 4 of integer 2
0x...	...	Int32	R/O	Block 4 of integer ...
0x19BF	6591	Int32	R/O	Block 4 of integer 24
0x19C1	6593	Int32	R/O	Block 5 of integer 1 = diverse program data
0x19C3	6597	Int32	R/O	Block 5 of integer 2
0x...	...	Int32	R/O	Block 5 of integer ...
0x19EF	6639	Int32	R/O	Block 5 of integer 24
0x19F1	6641	Int32	R/O	Block 6 of integer 1 = diverse program data
0x19F3	6643	Int32	R/O	Block 6 of integer 2
0x...	...	Int32	R/O	Block 6 of integer ...
0x1A1F	6687	Int32	R/O	Block 6 of integer 24
0x1A21	6689	Int32	R/O	Block 7 of integer 1 = diverse program data
0x1A23	6691	Int32	R/O	Block 7 of integer 2
0x...	...	Int32	R/O	Block 7 of integer ...
0x1A4F	6735	Int32	R/O	Block 7 of integer 24
0x1A51	6737	Int32	R/O	Block 8 of integer 1 = diverse program data
0x1A53	6739	Int32	R/O	Block 8 of integer 2
0x...	...	Int32	R/O	Block 8 of integer ...
0x1A7F	6783	Int32	R/O	Block 8 of integer 24

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x1A81	6785	Int32	R/O	Block 9 of integer 1 = diverse program data
0x1A83	6787	Int32	R/O	Block 9 of integer 2
0x...	...	Int32	R/O	Block 9 of integer ...
0x1AAF	6831	Int32	R/O	Block 9 of integer 24
0x1AB1	6833	Int32	R/O	Block 10 of integer 1
0x1AB3	6835	Int32	R/O	Block 10 of integer 2
0x...	...	Int32	R/O	Block 10 of integer ...
0x1ADF	6879	Int32	R/O	Block 10 of integer 24
0x1AE1	6881	Int32	R/O	Block 11 of integer 1
0x1AE3	6883	Int32	R/O	Block 11 of integer 2
0x...	...	Int32	R/O	Block 11 of integer ...
0x1B0F	6927	Int32	R/O	Block 11 of integer 24
0x1B11	6929	Int32	R/O	Block 12 of integer 1
0x1B13	6931	Int32	R/O	Block 12 of integer 2
0x...	...	Int32	R/O	Block 12 of integer ...
0x1B3F	6975	Int32	R/O	Block 12 of integer 24

7 Modbus address tables

PLC digital outputs

The meaning of PLC digital outputs (18 blocks each with 32 outputs) is determined by the PLC program. They are named in the setup program under **CPU > SETUP ONLY > DIGITAL PLC OUTPUT TEXTS**.

PLC digital outputs (name provided at factory)

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x1B41	6977	Bit field 32	R/O	Block 1 digital outputs; intended for operating contacts, status, and tolerance band signals of program generator 1
		Bit 0	R/O	Digital output 1, (operating contact 1)
		Bit 1	R/O	Digital output 2, (operating contact 2)
		Bit 2	R/O	Digital output 3, (operating contact 3)
		Bit 3	R/O	Digital output 4, (operating contact 4)
		Bit 4	R/O	Digital output 5, (operating contact 5)
		Bit 5	R/O	Digital output 6, (operating contact 6)
		Bit 6	R/O	Digital output 7, (operating contact 7)
		Bit 7	R/O	Digital output 8, (operating contact 8)
		Bit 8	R/O	Digital output 9, (operating contact 9)
		Bit 9	R/O	Digital output 10, (operating contact 10)
		Bit 10	R/O	Digital output 11, (operating contact 11)
		Bit 11	R/O	Digital output 12, (operating contact 12)
		Bit 12	R/O	Digital output 13, (operating contact 13)
		Bit 13	R/O	Digital output 14, (operating contact 14)
		Bit 14	R/O	Digital output 15, (operating contact 15)
		Bit 15	R/O	Digital output 16, (operating contact 16)
		Bit 16	R/O	Digital output 17 (basic setting)
		Bit 17	R/O	Digital output 18, (generator status: "Automatic" operating mode)
		Bit 18	R/O	Digital output 19, (generator status: "Extended automatic" operating mode)
		Bit 19	R/O	Digital output 20, (generator status: "Standstill" operating mode)
		Bit 20	R/O	Digital output 21, (generator status: "Delay" operating mode)
		Bit 21	R/O	Digital output 22, (generator status: "Program end" operating mode)
		Bit 22	R/O	Digital output 23, (generator status: "Manual mode" operating mode)
		Bit 23	R/O	Digital output 24, (tol.range channel 1)
		Bit 24	R/O	Digital output 25, (tol.range channel 2)
Bit 25	R/O	Digital output 26, (tol.range channel 3)		

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x1B43	6979	Bit field 32	R/O	Block 2 digital outputs; intended for program generator 2
0x1B45	6981	Bit field 32	R/O	Block 3 digital outputs; intended for program generator 3
0x1B47	6983	Bit field 32	R/O	Block 4 digital outputs; intended for program generator 4
0x1B49	6985	Bit field 32	R/O	Block 5 digital outputs; intended for program generator 5
0x1B4B	6987	Bit field 32	R/O	Block 6 digital outputs; intended for program generator 6
0x1B4D	6989	Bit field 32	R/O	Block 7 digital outputs; intended for program generator 7
0x1B4F	6991	Bit field 32	R/O	Block 8 digital outputs; intended for program generator 8
0x1B51	6993	Bit field 32	R/O	Block 9 digital outputs; intended for program generator 9
0x1B53	6995	Bit field 32	R/O	Block 10 digital outputs; intended for limit value monitoring 1 to 32
		Bit 0	R/O	Output 1 (limit value monitoring 1)
		Bit ...	R/O	Output ...
		Bit 31	R/O	Output 32 (limit value monitoring 32)
0x1B55	6997	Bit field 32	R/O	Block 11 digital outputs; intended for limit value monitoring 33 to 64
		Bit 0	R/O	Output 1 (limit value monitoring 33)
		Bit ...	R/O	Output ...
		Bit 31	R/O	Output 32 (limit value monitoring 64)
0x1B57	6999	Bit field 32	R/O	Block 12 digital outputs; intended for binary linking 1 to 8
		Bit 0	R/O	Block 12 of digital output 1 (binary linking 1)
		Bit ...	R/O	Block 12 of digital output ...
		Bit 7	R/O	Block 12 of digital output 8 (binary linking 8)
		Bit 8 to 31	R/O	Block 12 of digital outputs 9 to 32
0x1B59	7001	Bit field 32	R/O	Block 13 of digital outputs 1 to 32
0x1B5B	7003	Bit field 32	R/O	Block 14 of digital outputs 1 to 32
0x1B5D	7005	Bit field 32	R/O	Block 15 of digital outputs 1 to 32
0x1B5F	7007	Bit field 32	R/O	Block 16 of digital outputs 1 to 32
0x1B61	7009	Bit field 32	R/O	Block 17 of digital outputs 1 to 32
0x1B63	7011	Bit field 32	R/O	Block 18 of digital outputs 1 to 32

PLC program text for generator 1 to 9

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x1B65	7013	Char[73]	R/O	Program text for generator 1 (= Unicode)
0x1B8A	7050	Char[73]	R/O	Program text for generator 2 (= Unicode)
0x...	...	Char[73]	R/O	Program text for generator ... (= Unicode)
0x1C8D	7309	Char[73]	R/O	Program text for generator 9 (= Unicode)

7 Modbus address tables

PLC program texts for program channels 1 to 3

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x1CB2	7346	Char[73]	R/O	Text for program channel 1 for generator 1 (= Unicode)
0x1CD7	7383	Char[73]	R/O	Text for program channel 1 for generator 2 (= Unicode)
0x...	...	Char[73]	R/O	Text for program channel 1 for generator ... (= Unicode)
0x1DDA	7642	Char[73]	R/O	Text for program channel 1 for generator 9 (= Unicode)
0x1DFF	7679	Char[73]	R/O	Text for program channel 2 for generator 1 (= Unicode)
0x1E24	7716	Char[73]	R/O	Text for program channel 2 for generator 2 (= Unicode)
0x...	...	Char[73]	R/O	Text for program channel 2 for generator ... (= Unicode)
0x1F27	7975	Char[73]	R/O	Text for program channel 2 for generator 9 (= Unicode)
0x1F4C	8012	Char[73]	R/O	Text for program channel 3 for generator 1 (= Unicode)
0x1F71	8049	Char[73]	R/O	Text for program channel 3 for generator 2 (= Unicode)
0x...	...	Char[73]	R/O	Text for program channel 3 for generator ... (= Unicode)
0x2074	8308	Char[73]	R/O	Text for program channel 3 for generator 9 (= Unicode)

PLC limit value 1 and 2 for alarms of the analog variables 1 to 64

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x20B6	8374	Float	R/W	Limit value 1 for alarm of analog variable 1
0x20B8	8376	Float	R/W	Limit value 1 for alarm of analog variable 2
0x...	...	Float	R/W	Limit value 1 for alarm of analog variable ...
0x2134	8500	Float	R/W	Limit value 1 for alarm of analog variable 64
0x2136	8502	Float	R/W	Limit value 2 for alarm of analog variable 1
0x2138	8504	Float	R/W	Limit value 2 for alarm of analog variable 2
0x...	...	Float	R/W	Limit value 2 for alarm of analog variable ...
0x21B4	8628	Float	R/W	Limit value 2 for alarm of analog variable 64

PLC switching differentials for alarms of the analog variables 1 to 64

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x21B6	8630	Float	R/W	Switching differential for alarm of analog variable 1
0x21B8	8632	Float	R/W	Switching differential for alarm of analog variable 2
0x...	...	Float	R/W	Switching differential for alarm of analog variable ...
0x2234	8756	Float	R/W	Switching differential for alarm of analog variable 64

7 Modbus address tables

PLC limit value 1 and 2 for alarms of the integer variables 1 to 64

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x2236	8758	Int32	R/W	Limit value 1 for alarm of integer variable 1
0x2238	8760	Int32	R/W	Limit value 1 for alarm of integer variable 2
0x...	...	Int32	R/W	Limit value 1 for alarm of integer variable ...
0x22B4	8884	Int32	R/W	Limit value 1 for alarm of integer variable 64
0x22B6	8886	Int32	R/W	Limit value 2 for alarm of integer variable 1
0x22B8	8888	Int32	R/W	Limit value 2 for alarm of integer variable 2
0x...	...	Int32	R/W	Limit value 2 for alarm of integer variable ...
0x2334	9012	Int32	R/W	Limit value 2 for alarm of integer variable 64

PLC switching differential for alarms of the integer variables 1 to 64

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x2336	9014	Int32	R/W	Switching differential for alarm of integer variable 1
0x2338	9016	Int32	R/W	Switching differential for alarm of integer variable 2
0x...	...	Int32	R/W	Switching differential for alarm of integer variable ...
0x23B4	9140	Int32	R/W	Switching differential for alarm of integer variable 64

PLC limit values for limit value monitoring

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x23B6	9142	Float	R/W	Limit value for limit value monitoring 1
0x23B8	9144	Float	R/W	Limit value for limit value monitoring 2
0x...	...	Float	R/W	Limit value for limit value monitoring ...
0x2434	9268	Float	R/W	Limit value for limit value monitoring 64

Configuration, switching differential for limit value monitoring

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x2436	9270	Float	R/W	Switching differential for limit value monitoring 1
0x2438	9272	Float	R/W	Switching differential for limit value monitoring 2
0x...	...	Float	R/W	Switching differential for limit value monitoring ...
0x24B4	9396	Float	R/W	Switching differential for limit value monitoring 64

7 Modbus address tables

Compiled reading frames

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

Compiled writing frames

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

7.3 Modbus addresses of other modules

If the central processing unit detects other modules, they can be accessed using the following addresses depending on the module type.

The additional modules are available for configuration under virtual Modbus device addresses (1 to 254) as with the setup program under **PROJECT > HARDWARE ARRANGEMENT**.

7.3.1 Modbus addresses for each HMI module

Process values based on a HMI module

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0002	5	Bit field 64	R/O	Bit 0 to 53 = 0x3FFFFFFFFFFFFFFF : binary spur 1 to 54
0x0006	9	Bit field 64	R/O	Bit 0 to 8 = 0x1FF : batch start 1 to 9
0x0011	17	Float	R/O	Analog spur 1
0x0013	19	Float	R/O	Analog spur 2
0x...	...	Float	R/O	Analog spur ...
0x007B	123	Float	R/O	Analog spur 54

Process values of a HMI module for the basis

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x008C	140	Bit field 64	R/O	Bit 0 to 8 = 0x1FF: collective alarm for batch 1 to 9 Bit 9 = 0x200: collective alarm of recording function Bit 10 = 0x400: fault with recording function Bit 11 to 19= 0xFF800: active batch 1 to 9
0x009D	157	Float	R/O	Current value of counter 1 (current value of the counter or integrator)
0x009F	159	Float	R/O	Current value of counter 2
0x...	...	Float	R/O	Current value of counter ...
0x00D1	209	Float	R/O	Current value of counter 27
0x00D3	211	Float	R/O	Last value of counter 1 (value of the counter or integrator during the last recording period)
0x00D5	213	Float	R/O	Last value of counter 2
0x...	...	Float	R/O	Last value of counter ...
0x0107	263	Float	R/O	Last value of counter 27

7 Modbus address tables

7.3.2 Modbus addresses for each multichannel controller module



NOTE!

Signals with labels "NV_..." are intended for the function that corresponds to the respective default names. However, the function ultimately depends on how the signals are used in the configuration.

Signals from the central processing unit to a multichannel controller module

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0002	2	Bit field 64	R/O	The bits of this bit field are also available as individual bool values (R/W) from Modbus address 0x01F8. The meaning of the respective bit or value is listed there.
0x0008	8	Float	R/O	Actual value for controller channel 1 (NV_C01ActualValue)
0x000A	10	Float	R/O	Setpoint value for controller channel 1 (NV_C01Setpoint)
0x000C	12	Float	R/O	Actual value for controller channel 2 (NV_C02ActualValue)
0x000E	14	Float	R/O	Setpoint value for controller channel 2 (NV_C02Setpoint)
0x0010	16	Float	R/O	Actual value for controller channel 3 (NV_C03ActualValue)
0x0012	18	Float	R/O	Setpoint value for controller channel 3 (NV_C03Setpoint)
0x0014	20	Float	R/O	Actual value for controller channel 4 (NV_C04ActualValue)
0x0016	22	Float	R/O	Setpoint value for controller channel 4 (NV_C04Setpoint)
0x0018	24	Float	R/O	External setpoint value for controller channel 1 (NV_SP01Ext)
0x001A	26	Float	R/O	External setpoint value for controller channel 2 (NV_SP02Ext)
0x001C	28	Float	R/O	External setpoint value for controller channel 3 (NV_SP03Ext)
0x001E	30	Float	R/O	External setpoint value for controller channel 4 (NV_SP04Ext)
0x0020	32	Float	R/O	Flag of controller channel 1 (NV_M01Flag)
0x0022	34	Float	R/O	Flag of controller channel 2 (NV_M02Flag)
0x0024	36	Float	R/O	Flag of controller channel 3 (NV_M03Flag)
0x0026	38	Float	R/O	Flag of controller channel 4 (NV_M04Flag)

7 Modbus address tables

Signals from the controller module to the central processing unit

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0042	66	Bit field 16	R/O	Module error
0x0045	69	Bit field 8	R/O	Bit 0 = 0x01: digital input 1 Bit 1 = 0x02: digital input 2 Bit 2 = 0x04: digital input 5 Bit 3 = 0x08: digital input 6 Bit 4 = 0x10: digital input 7 Bit 5 = 0x20: digital input 8 Bit 6 = 0x40: digital input 9 Bit 7 = 0x80: digital input 10
0x0046	70	Bit field 8	R/O	Bit 0 = 0x01: logic result 1 Bit 1 = 0x02: logic result 2 Bit 2 = 0x04: logic result 3 Bit 3 = 0x08: logic result 4
0x0047	71	Bit field 8	R/O	Bit 0 = 0x01: limit value monitoring 1 Bit 1 = 0x02: limit value monitoring 2 Bit 2 = 0x04: limit value monitoring 3 Bit 3 = 0x08: limit value monitoring 4
0x0048	72	Bit field 8	R/O	Bit 0 = 0x01: alarm 1 of analog input 1 Bit 1 = 0x02: alarm 1 of analog input 2 Bit 2 = 0x04: alarm 1 of analog input 3 Bit 3 = 0x08: alarm 1 of analog input 4 Bit 4 = 0x10: alarm 2 of analog input 1 Bit 5 = 0x20: alarm 2 of analog input 2 Bit 6 = 0x40: alarm 2 of analog input 3 Bit 7 = 0x80: alarm 2 of analog input 4
0x0049	73	Bit field 32	R/O	Bit 0 to 3 = 0x0F: manual mode controller channel 1 to 4 Bit 4 to 7 = 0xF0: self-optimization of active controller channel 1 to 4 Bit 8 to 11 = 0x0F00: output 1 of controller channel 1 to 4 Bit 12 to 15 = 0xF000: output 2 of controller channel 1 to 4 Bit 16 to 19 = 0xF0000: collective alarm of controller channel 1 to 4
0x004B	75	Bit field 16	R/O	Bit 0 to 3 = 0x0F: ramp function active for controller channel 1 to 4 Bit 4 to 11 = 0xFF0: setpoint changeover for controller 1 to 4 (bit 0 and bit 1 each)
0x004C	76	Bit field 8	R/O	Bit 1 = 0x02: controller module collective alarm
0x004D	77	Bit field 8	R/O	Bit 0 to 7 = 0xFF: digital output 3 to 10

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0052	82	Float	R/O	HW_counter
0x0054	84	Float	R/O	Analog input 1
0x0056	86	Float	R/O	Analog input 2
0x0058	88	Float	R/O	Analog input 3
0x005A	90	Float	R/O	Analog input 4
0x005C	92	Float	R/O	Math 1
0x005E	94	Float	R/O	Math 2
0x0060	96	Float	R/O	Math 3
0x0062	98	Float	R/O	Math 4
0x0064	100	Float	R/O	Output level for controller channel 1
0x0066	102	Float	R/O	Output level for controller channel 2
0x0068	104	Float	R/O	Output level for controller channel 3
0x006A	106	Float	R/O	Output level for controller channel 4
0x006C	108	Float	R/O	Actual value for controller channel 1
0x006E	110	Float	R/O	Actual value for controller channel 2
0x0070	112	Float	R/O	Actual value for controller channel 3
0x0072	114	Float	R/O	Actual value for controller channel 4
0x0074	116	Float	R/O	Setpoint value for controller channel 1
0x0076	118	Float	R/O	Setpoint value for controller channel 2
0x0078	120	Float	R/O	Setpoint value for controller channel 3
0x007A	122	Float	R/O	Setpoint value for controller channel 4
0x007C	124	Float	R/O	Setpoint value function 1
0x007E	126	Float	R/O	Setpoint value function 2
0x0080	128	Float	R/O	Setpoint value function 3
0x0082	130	Float	R/O	Setpoint value function 4

Multichannel controller module limit value 1 and 2 for alarms of the analog variables 1 to 4

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x008C	140	Float	R/W	Limit value 1 for alarm of analog variable 1
0x008E	142	Float	R/W	Limit value 1 for alarm of analog variable 2
0x0090	144	Float	R/W	Limit value 1 for alarm of analog variable 3
0x0092	146	Float	R/W	Limit value 1 for alarm of analog variable 4
0x0094	148	Float	R/W	Limit value 2 for alarm of analog variable 1
0x0096	150	Float	R/W	Limit value 2 for alarm of analog variable 2
0x0098	152	Float	R/W	Limit value 2 for alarm of analog variable 3

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x009A	154	Float	R/W	Limit value 2 for alarm of analog variable 4

Multichannel controller module, switching differentials for alarms of the analog variables 1 to 4

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x009C	156	Float	R/W	Switching differential for alarm of analog variable 1
0x009E	158	Float	R/W	Switching differential for alarm of analog variable 2
0x00A0	160	Float	R/W	Switching differential for alarm of analog variable 3
0x00A2	162	Float	R/W	Switching differential for alarm of analog variable 4

Multichannel controller module parameter XP1

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x00A4	164	Float	R/W	Parameter table: XP1 1 => controller 1: parameter block 1
0x00A6	166	Float	R/W	Parameter table: XP1 2 => controller 1: parameter block 2
0x00A8	168	Float	R/W	Parameter table: XP1 3 => controller 2: parameter block 1
0x00AA	170	Float	R/W	Parameter table: XP1 4 => controller 2: parameter block 2
0x00AC	172	Float	R/W	Parameter table: XP1 5 => controller 3: parameter block 1
0x00AE	174	Float	R/W	Parameter table: XP1 6 => controller 3: parameter block 2
0x00B0	176	Float	R/W	Parameter table: XP1 7 => controller 4: parameter block 1
0x00B2	178	Float	R/W	Parameter table: XP1 8 => controller 4: parameter block 2

Multichannel controller module parameter XP2

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x00B4	180	Float	R/W	Parameter table: XP2 1 => controller 1: parameter block 1
0x00B6	182	Float	R/W	Parameter table: XP2 2 => controller 1: parameter block 2
0x00B8	184	Float	R/W	Parameter table: XP2 3 => controller 2: parameter block 1
0x00BA	186	Float	R/W	Parameter table: XP2 4 => controller 2: parameter block 2
0x00BC	188	Float	R/W	Parameter table: XP2 5 => controller 3: parameter block 1
0x00BE	190	Float	R/W	Parameter table: XP2 6 => controller 3: parameter block 2
0x00C0	192	Float	R/W	Parameter table: XP2 7 => controller 4: parameter block 1
0x00C2	194	Float	R/W	Parameter table: XP2 8 => controller 4: parameter block 2

7 Modbus address tables

Multichannel controller module parameter TV1

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x00C4	196	Float	R/W	Parameter table: TV1 1 => controller 1: parameter block 1
0x00C6	198	Float	R/W	Parameter table: TV1 2 => controller 1: parameter block 2
0x00C8	200	Float	R/W	Parameter table: TV1 3 => controller 2: parameter block 1
0x00CA	202	Float	R/W	Parameter table: TV1 4 => controller 2: parameter block 2
0x00CC	204	Float	R/W	Parameter table: TV1 5 => controller 3: parameter block 1
0x00CE	206	Float	R/W	Parameter table: TV1 6 => controller 3: parameter block 2
0x00D0	208	Float	R/W	Parameter table: TV1 7 => controller 4: parameter block 1
0x00D2	210	Float	R/W	Parameter table: TV1 8 => controller 4: parameter block 2

Multichannel controller module parameter TV2

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x00D4	212	Float	R/W	Parameter table: TV2 1 => controller 1: parameter block 1
0x00D6	214	Float	R/W	Parameter table: TV2 2 => controller 1: parameter block 2
0x00D8	216	Float	R/W	Parameter table: TV2 3 => controller 2: parameter block 1
0x00DA	218	Float	R/W	Parameter table: TV2 4 => controller 2: parameter block 2
0x00DC	220	Float	R/W	Parameter table: TV2 5 => controller 3: parameter block 1
0x00DE	222	Float	R/W	Parameter table: TV2 6 => controller 3: parameter block 2
0x00E0	224	Float	R/W	Parameter table: TV2 7 => controller 4: parameter block 1
0x00E2	226	Float	R/W	Parameter table: TV2 8 => controller 4: parameter block 2

Multichannel controller module parameter TN1

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x00E4	228	Float	R/W	Parameter table: TN1 1 => controller 1: parameter block 1
0x00E6	230	Float	R/W	Parameter table: TN1 2 => controller 1: parameter block 2
0x00E8	232	Float	R/W	Parameter table: TN1 3 => controller 2: parameter block 1
0x00EA	234	Float	R/W	Parameter table: TN1 4 => controller 2: parameter block 2
0x00EC	236	Float	R/W	Parameter table: TN1 5 => controller 3: parameter block 1
0x00EE	238	Float	R/W	Parameter table: TN1 6 => controller 3: parameter block 2
0x00F0	240	Float	R/W	Parameter table: TN1 7 => controller 4: parameter block 1
0x00F2	242	Float	R/W	Parameter table: TN1 8 => controller 4: parameter block 2

7 Modbus address tables

Multichannel controller module parameter TN2

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x00F4	244	Float	R/W	Parameter table: TN2 1 => controller 1: parameter block 1
0x00F6	246	Float	R/W	Parameter table: TN2 2 => controller 1: parameter block 2
0x00F8	248	Float	R/W	Parameter table: TN2 3 => controller 2: parameter block 1
0x00FA	250	Float	R/W	Parameter table: TN2 4 => controller 2: parameter block 2
0x00FC	252	Float	R/W	Parameter table: TN2 5 => controller 3: parameter block 1
0x00FE	254	Float	R/W	Parameter table: TN2 6 => controller 3: parameter block 2
0x0100	256	Float	R/W	Parameter table: TN2 7 => controller 4: parameter block 1
0x0102	258	Float	R/W	Parameter table: TN2 8 => controller 4: parameter block 2

Multichannel controller module parameter CY1

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0104	260	Float	R/W	Parameter table: CY1 1 => controller 1: parameter block 1
0x0106	262	Float	R/W	Parameter table: CY1 2 => controller 1: parameter block 2
0x0108	264	Float	R/W	Parameter table: CY1 3 => controller 2: parameter block 1
0x010A	266	Float	R/W	Parameter table: CY1 4 => controller 2: parameter block 2
0x010C	268	Float	R/W	Parameter table: CY1 5 => controller 3: parameter block 1
0x010E	270	Float	R/W	Parameter table: CY1 6 => controller 3: parameter block 2
0x0110	272	Float	R/W	Parameter table: CY1 7 => controller 4: parameter block 1
0x0112	274	Float	R/W	Parameter table: CY1 8 => controller 4: parameter block 2

Multichannel controller module parameter CY2

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0114	276	Float	R/W	Parameter table: CY2 1 => controller 1: parameter block 1
0x0116	278	Float	R/W	Parameter table: CY2 2 => controller 1: parameter block 2
0x0118	280	Float	R/W	Parameter table: CY2 3 => controller 2: parameter block 1
0x011A	282	Float	R/W	Parameter table: CY2 4 => controller 2: parameter block 2
0x011C	284	Float	R/W	Parameter table: CY2 5 => controller 3: parameter block 1
0x011E	286	Float	R/W	Parameter table: CY2 6 => controller 3: parameter block 2
0x0120	288	Float	R/W	Parameter table: CY2 7 => controller 4: parameter block 1
0x0122	290	Float	R/W	Parameter table: CY2 8 => controller 4: parameter block 2

7 Modbus address tables

Multichannel controller module parameter XSH 1 to 8

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0124	292	Float	R/W	Parameter table: XSH 1 => controller 1: parameter block 1
0x0126	294	Float	R/W	Parameter table: XSH 2 => controller 1: parameter block 2
0x0128	296	Float	R/W	Parameter table: XSH 3 => controller 2: parameter block 1
0x012A	298	Float	R/W	Parameter table: XSH 4 => controller 2: parameter block 2
0x012C	300	Float	R/W	Parameter table: XSH 5 => controller 3: parameter block 1
0x012E	302	Float	R/W	Parameter table: XSH 6 => controller 3: parameter block 2
0x0130	304	Float	R/W	Parameter table: XSH 7 => controller 4: parameter block 1
0x0132	306	Float	R/W	Parameter table: XSH 8 => controller 4: parameter block 2

Multichannel controller module parameter XD1

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0134	308	Float	R/W	Parameter table: XD1 1 => controller 1: parameter block 1
0x0136	310	Float	R/W	Parameter table: XD1 2 => controller 1: parameter block 2
0x0138	312	Float	R/W	Parameter table: XD1 3 => controller 2: parameter block 1
0x013A	314	Float	R/W	Parameter table: XD1 4 => controller 2: parameter block 2
0x013C	316	Float	R/W	Parameter table: XD1 5 => controller 3: parameter block 1
0x013E	318	Float	R/W	Parameter table: XD1 6 => controller 3: parameter block 2
0x0140	320	Float	R/W	Parameter table: XD1 7 => controller 4: parameter block 1
0x0142	322	Float	R/W	Parameter table: XD1 8 => controller 4: parameter block 2

Multichannel controller module parameter XD2

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0144	324	Float	R/W	Parameter table: XD2 1 => controller 1: parameter block 1
0x0146	326	Float	R/W	Parameter table: XD2 2 => controller 1: parameter block 2
0x0148	328	Float	R/W	Parameter table: XD2 3 => controller 2: parameter block 1
0x014A	330	Float	R/W	Parameter table: XD2 4 => controller 2: parameter block 2
0x014C	332	Float	R/W	Parameter table: XD2 5 => controller 3: parameter block 1
0x014E	334	Float	R/W	Parameter table: XD2 6 => controller 3: parameter block 2
0x0150	336	Float	R/W	Parameter table: XD2 7 => controller 4: parameter block 1
0x0152	338	Float	R/W	Parameter table: XD2 8 => controller 4: parameter block 2

7 Modbus address tables

Multichannel controller module parameter TT 1 to 8

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0154	340	Word	R/W	Parameter table: TT1 => controller 1: parameter block 1
0x0155	341	Word	R/W	Parameter table: TT 2 => controller 1: parameter block 2
0x0156	342	Word	R/W	Parameter table: TT 3 => controller 2: parameter block 1
0x0157	343	Word	R/W	Parameter table: TT 4 => controller 2: parameter block 2
0x0158	344	Word	R/W	Parameter table: TT 5 => controller 3: parameter block 1
0x0159	345	Word	R/W	Parameter table: TT 6 => controller 3: parameter block 2
0x015A	346	Word	R/W	Parameter table: TT 7 => controller 4: parameter block 1
0x015B	347	Word	R/W	Parameter table: TT 8 => controller 4: parameter block 2

Multichannel controller module parameter Y0

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x015C	348	Byte	R/W	Parameter table: Y0 1 => controller 1: parameter block 1
0x015D	349	Byte	R/W	Parameter table: Y0 2 => controller 1: parameter block 2
0x015E	350	Byte	R/W	Parameter table: Y0 3 => controller 2: parameter block 1
0x015F	351	Byte	R/W	Parameter table: Y0 4 => controller 2: parameter block 2
0x0160	352	Byte	R/W	Parameter table: Y0 5 => controller 3: parameter block 1
0x0161	353	Byte	R/W	Parameter table: Y0 6 => controller 3: parameter block 2
0x0162	354	Byte	R/W	Parameter table: Y0 7 => controller 4: parameter block 1
0x0163	355	Byte	R/W	Parameter table: Y0 8 => controller 4: parameter block 2

Multichannel controller module parameter Y1

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0164	356	Byte	R/W	Parameter table: Y1 1 => controller 1: parameter block 1
0x0165	357	Byte	R/W	Parameter table: Y1 2 => controller 1: parameter block 2
0x0166	358	Byte	R/W	Parameter table: Y1 3 => controller 2: parameter block 1
0x0167	359	Byte	R/W	Parameter table: Y1 4 => controller 2: parameter block 2
0x0168	360	Byte	R/W	Parameter table: Y1 5 => controller 3: parameter block 1
0x0169	361	Byte	R/W	Parameter table: Y1 6 => controller 3: parameter block 2
0x016A	362	Byte	R/W	Parameter table: Y1 7 => controller 4: parameter block 1
0x016B	363	Byte	R/W	Parameter table: Y1 8 => controller 4: parameter block 2

7 Modbus address tables

Multichannel controller module parameter Y2

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x016C	364	Byte	R/W	Parameter table: Y2 1 => controller 1: parameter block 1
0x016D	365	Byte	R/W	Parameter table: Y2 2 => controller 1: parameter block 2
0x016E	366	Byte	R/W	Parameter table: Y2 3 => controller 2: parameter block 1
0x016F	367	Byte	R/W	Parameter table: Y2 4 => controller 2: parameter block 2
0x0170	368	Byte	R/W	Parameter table: Y2 5 => controller 3: parameter block 1
0x0171	369	Byte	R/W	Parameter table: Y2 6 => controller 3: parameter block 2
0x0172	370	Byte	R/W	Parameter table: Y2 7 => controller 4: parameter block 1
0x0173	371	Byte	R/W	Parameter table: Y2 8 => controller 4: parameter block 2

Multichannel controller module parameter TK1

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0174	372	Float	R/W	Parameter table: TK1 1 => controller 1: parameter block 1
0x0176	374	Float	R/W	Parameter table: TK1 2 => controller 1: parameter block 2
0x0178	376	Float	R/W	Parameter table: TK1 3 => controller 2: parameter block 1
0x017A	378	Float	R/W	Parameter table: TK1 4 => controller 2: parameter block 2
0x017C	380	Float	R/W	Parameter table: TK1 5 => controller 3: parameter block 1
0x017E	382	Float	R/W	Parameter table: TK1 6 => controller 3: parameter block 2
0x0180	384	Float	R/W	Parameter table: TK1 7 => controller 4: parameter block 1
0x0182	386	Float	R/W	Parameter table: TK1 8 => controller 4: parameter block 2

Multichannel controller module parameter TK2

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0184	388	Float	R/W	Parameter table: TK2 1 => controller 1: parameter block 1
0x0186	390	Float	R/W	Parameter table: TK2 2 => controller 1: parameter block 2
0x0188	392	Float	R/W	Parameter table: TK2 3 => controller 2: parameter block 1
0x018A	394	Float	R/W	Parameter table: TK2 4 => controller 2: parameter block 2
0x018C	396	Float	R/W	Parameter table: TK2 5 => controller 3: parameter block 1
0x018E	398	Float	R/W	Parameter table: TK2 6 => controller 3: parameter block 2
0x0190	400	Float	R/W	Parameter table: TK2 7 => controller 4: parameter block 1
0x0192	402	Float	R/W	Parameter table: TK2 8 => controller 4: parameter block 2

7 Modbus address tables

Multichannel controller module offset 1 to 4, each for setpoint value 1 to 4

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0194	404	Float	R/W	Setpoint value offset 1 for controller channel 1 (or fixed setpoint value 1)
0x0196	406	Float	R/W	Setpoint value offset 2 for controller channel 1 (or fixed setpoint value 2)
0x0198	408	Float	R/W	Setpoint value offset 3 for controller channel 1 (or fixed setpoint value 3)
0x019A	410	Float	R/W	Setpoint value offset 4 for controller channel 1 (or fixed setpoint value 4)
0x019C	412	Float	R/W	Setpoint value offset 1 for controller channel 2 (or fixed setpoint value 1)
0x019E	414	Float	R/W	Setpoint value offset 2 for controller channel 2 (or fixed setpoint value 2)
0x01A0	416	Float	R/W	Setpoint value offset 3 for controller channel 2 (or fixed setpoint value 3)
0x01A2	418	Float	R/W	Setpoint value offset 4 for controller channel 2 (or fixed setpoint value 4)
0x01A4	420	Float	R/W	Setpoint value offset 1 for controller channel 3 (or fixed setpoint value 1)
0x01A6	422	Float	R/W	Setpoint value offset 2 for controller channel 3 (or fixed setpoint value 2)
0x01A8	424	Float	R/W	Setpoint value offset 3 for controller channel 3 (or fixed setpoint value 3)
0x01AA	426	Float	R/W	Setpoint value offset 4 for controller channel 3 (or fixed setpoint value 4)
0x01AC	428	Float	R/W	Setpoint value offset 1 for controller channel 4 (or fixed setpoint value 1)
0x01AE	430	Float	R/W	Setpoint value offset 2 for controller channel 4 (or fixed setpoint value 2)
0x01B0	432	Float	R/W	Setpoint value offset 3 for controller channel 4 (or fixed setpoint value 3)
0x01B2	434	Float	R/W	Setpoint value offset 4 for controller channel 4 (or fixed setpoint value 4)

Multichannel controller module limit value monitoring 1 to 4

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x01B4	436	Float	R/W	Limit value for limit value monitoring 1

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x01B6	438	Float	R/W	Limit value for limit value monitoring 2
0x01B8	440	Float	R/W	Limit value for limit value monitoring 3
0x01BA	442	Float	R/W	Limit value for limit value monitoring 4

Multichannel controller module switching difference for limit value monitoring 1 to 4

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x01BC	444	Float	R/W	Switching difference for limit value monitoring 1
0x01BE	446	Float	R/W	Switching difference for limit value monitoring 2
0x01C0	448	Float	R/W	Switching difference for limit value monitoring 3
0x01C2	450	Float	R/W	Switching difference for limit value monitoring 4

Multichannel controller module - Start of self-optimization for controller channel 1 to 4

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x01C4	452	Boolean	R/W	Start of self-optimization for controller channel 1
0x01C5	453	Boolean	R/W	Start of self-optimization for controller channel 2
0x01C6	454	Boolean	R/W	Start of self-optimization for controller channel 3
0x01C7	455	Boolean	R/W	Start of self-optimization for controller channel 4

Multichannel controller module - activation of manual mode for controller channel 1 to 4

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x01C8	456	Boolean	R/W	Manual mode of controller channel 1
0x01C9	457	Boolean	R/W	Manual mode of controller channel 2
0x01CA	458	Boolean	R/W	Manual mode of controller channel 3
0x01CB	459	Boolean	R/W	Manual mode of controller channel 4

Multichannel controller module - manual output level 1 to 4 of controller 2

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x01CC	460	Float	R/W	Manual output level 1 of controller 2
0x01CE	462	Float	R/W	Manual output level 2 of controller 2
0x01D0	464	Float	R/W	Manual output level 3 of controller 2
0x01D2	466	Float	R/W	Manual output level 4 of controller 2

7 Modbus address tables

Multichannel controller module - additional configuration parameters (as of system version 05)

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x01D4	468	Float	R/W	Setpoint value function 1: positive gradient of the ramp function
0x01D6	470	Float	R/W	Setpoint value function 2: positive gradient of the ramp function
0x01D8	472	Float	R/W	Setpoint value function 3: positive gradient of the ramp function
0x01DA	474	Float	R/W	Setpoint value function 4: positive gradient of the ramp function
0x01DC	476	Float	R/W	Setpoint value function 1: negative gradient of the ramp function
0x01DE	478	Float	R/W	Setpoint value function 2: negative gradient of the ramp function
0x01E0	480	Float	R/W	Setpoint value function 3: negative gradient of the ramp function
0x01E2	482	Float	R/W	Setpoint value function 4: negative gradient of the ramp function
0x01E4	484	Float	R/W	Setpoint value function 1: start value of the ramp function
0x01E6	486	Float	R/W	Setpoint value function 2: start value of the ramp function
0x01E8	488	Float	R/W	Setpoint value function 3: start value of the ramp function
0x01EA	490	Float	R/W	Setpoint value function 4: start value of the ramp function
0x01EC	492	Word	R/W	Limit value monitoring 1: switch-on delay
0x01ED	493	Word	R/W	Limit value monitoring 2: switch-on delay
0x01EE	494	Word	R/W	Limit value monitoring 3: switch-on delay
0x01EF	495	Word	R/W	Limit value monitoring 4: switch-on delay
0x01F0	496	Word	R/W	Limit value monitoring 1: switch-off delay
0x01F1	497	Word	R/W	Limit value monitoring 2: switch-off delay
0x01F2	498	Word	R/W	Limit value monitoring 3: switch-off delay
0x01F3	499	Word	R/W	Limit value monitoring 4: switch-off delay
0x01F4	500	Word	R/W	Limit value monitoring 1: pulse time
0x01F5	501	Word	R/W	Limit value monitoring 2: pulse time
0x01F6	502	Word	R/W	Limit value monitoring 3: pulse time
0x01F7	503	Word	R/W	Limit value monitoring 4: pulse time

7 Modbus address tables

Multichannel controller module - additional process values (as of system version 05)

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
				The following values are also available under Modbus address 0x0002 as Bit field 64 (R/O). The corresponding bits are indicated here in square brackets [...].
0x01F8	504	Boolean	R/W	[bit 0] Controller channel 1: self-optimization start (1 = start); (NV_C01TuneStart)
0x01F9	505	Boolean	R/W	[bit 1] Controller channel 1: self-optimization stop (1 = stop); (NV_C01TuneStop)
0x01FA	506	Boolean	R/W	[bit 2] Controller channel 1: parameter block (0 = param. block 1, 1 = param. block 2); (NV_C01ParamSet)
0x01FB	507	Boolean	R/W	[bit 3] Controller channel 1: manual mode (0 = inactive, 1 = active); (NV_C01ManualMode)
0x01FC	508	Boolean	R/W	[bit 4] Controller channel 1: controller off (0 = on, 1 = off); (NV_C01Stop)
0x01FD	509	Boolean	R/W	[bit 5] Controller channel 2: self-optimization start (1 = start); (NV_C02TuneStart)
0x01FE	510	Boolean	R/W	[bit 6] Controller channel 2: self-optimization stop (1 = stop); (NV_C02TuneStop)
0x01FF	511	Boolean	R/W	[bit 7] Controller channel 2: parameter block (0 = param. block 1, 1 = param. block 2); (NV_C02ParamSet)
0x0200	512	Boolean	R/W	[bit 8] Controller channel 2: manual mode (0 = inactive, 1 = active); (NV_C02ManualMode)
0x0201	513	Boolean	R/W	[bit 9] Controller channel 2: controller off (0 = on, 1 = off); (NV_C02Stop)
0x0202	514	Boolean	R/W	[bit 10] Controller channel 3: self-optimization start (1 = start); (NV_C03TuneStart)
0x0203	515	Boolean	R/W	[bit 11] Controller channel 3: self-optimization stop (1 = stop); (NV_C03TuneStop)
0x0204	516	Boolean	R/W	[bit 12] Controller channel 3: parameter block (0 = param. block 1, 1 = param. block 2); (NV_C03ParamSet)
0x0205	517	Boolean	R/W	[bit 13] Controller channel 3: manual mode (0 = inactive, 1 = active); (NV_C03ManualMode)
0x0206	518	Boolean	R/W	[bit 14] Controller channel 3: controller off (0 = on, 1 = off); (NV_C03Stop)

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0207	519	Boolean	R/W	[bit 15] Controller channel 4: self-optimization start (1 = start); (NV_C04TuneStart)
0x0208	520	Boolean	R/W	[bit 16] Controller channel 4: self-optimization stop (1 = stop); (NV_C01TuneStop)
0x0209	521	Boolean	R/W	[bit 17] Controller channel 4: parameter block (0 = param. block 1, 1 = param. block 2); (NV_C04ParamSet)
0x020A	522	Boolean	R/W	[bit 18] Controller channel 4: manual mode (0 = inactive, 1 = active); (NV_C04ManualMode)
0x020B	523	Boolean	R/W	[bit 19] Controller channel 4: controller off (0 = on, 1 = off); (NV_C04Stop)
0x020C	524	Boolean	R/W	[bit 20] Setpoint value function 1: setpoint changeover 1 (bit 0); (NV_SP01Changeover1)
0x020D	525	Boolean	R/W	[bit 21] Setpoint value function 1: setpoint changeover 2 (bit 1); (NV_SP01Changeover2)
0x020E	526	Boolean	R/W	[bit 22] Setpoint value function 2: setpoint changeover 1 (bit 0); (NV_SP02Changeover1)
0x020F	527	Boolean	R/W	[bit 23] Setpoint value function 2: setpoint changeover 2 (bit 1); (NV_SP02Changeover2)
0x0210	528	Boolean	R/W	[bit 24] Setpoint value function 3: setpoint changeover 1 (bit 0); (NV_SP03Changeover1)
0x0211	529	Boolean	R/W	[bit 25] Setpoint value function 3: setpoint changeover 2 (bit 1); (NV_SP03Changeover2)
0x0212	530	Boolean	R/W	[bit 26] Setpoint value function 4: setpoint changeover 1 (bit 0); (NV_SP04Changeover1)
0x0213	531	Boolean	R/W	[bit 27] Setpoint value function 4: setpoint changeover 2 (bit 1); (NV_SP04Changeover2)
0x0214	532	Boolean	R/W	[bit 28] Digital output 3: process value; (NV_DO3)
0x0215	533	Boolean	R/W	[bit 29] Digital output 4: process value; (NV_DO4)
0x0216	534	Boolean	R/W	[bit 30] Digital output 5: process value; (NV_DO5)
0x0217	535	Boolean	R/W	[bit 32] Digital output 6: process value; (NV_DO6)
0x0218	536	Boolean	R/W	[bit 32] Digital output 7: process value; (NV_DO7)

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0219	537	Boolean	R/W	[bit 33] Digital output 8: process value; (NV_DO8)
0x021A	538	Boolean	R/W	[bit 34] Digital output 9: process value; (NV_DO9)
0x021B	539	Boolean	R/W	[bit 35] Digital output 10: process value; (NV_DO10)
0x021C	540	Boolean	R/W	[bit 36] Limit value monitoring 1: acknowledgement (1 = active); (NV_LI01Confirmation)
0x021D	541	Boolean	R/W	[bit 37] Limit value monitoring 2: acknowledgement; (NV_LI02Confirmation)
0x021E	542	Boolean	R/W	[bit 38] Limit value monitoring 3: acknowledgement; (NV_LI03Confirmation)
0x021F	543	Boolean	R/W	[bit 39] Limit value monitoring 4: acknowledgement; (NV_LI04Confirmation)
0x0220	544	Boolean	R/W	[bit 40] Digital input 1: signal suppression (1 = active); (NV_SR01)
0x0221	545	Boolean	R/W	[bit 41] Digital input 2: signal suppression; (NV_SR02)
0x0222	546	Boolean	R/W	[bit 42] Digital input 5: signal suppression; (NV_SR03)
0x0223	547	Boolean	R/W	[bit 43] Digital input 6: signal suppression; (NV_SR04)
0x0224	548	Boolean	R/W	[bit 44] Digital input 7: signal suppression; (NV_SR05)
0x0225	549	Boolean	R/W	[bit 45] Digital input 8: signal suppression; (NV_SR06)
0x0226	550	Boolean	R/W	[bit 46] Digital input 9: signal suppression; (NV_SR07)
0x0227	551	Boolean	R/W	[bit 47] Digital input 10: signal suppression; (NV_SR08)

7 Modbus address tables

7.3.3 Modbus addresses for each 4-channel analog input module

Signals from the central processing unit to the 4-channel analog input module

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0002	2	Bit field 64	R/O	Bit 0 = 0x01: signal suppression 1 (NV_SR01) Bit 1 = 0x02: signal suppression 2 (NV_SR02) Bit 2 = 0x04: signal suppression 3 (NV_SR03) Bit 3 = 0x08: signal suppression 4 (NV_SR04) Bit 4 = 0x10: signal suppression 5 (NV_SR05)

Signals from the 4-channel analog input module to the central processing unit

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0042	66	Bit field 16	R/O	Module error
0x0045	69	Bit field 8	R/O	Bit 0 = 0x01: alarm 1 for channel 1 => alarm 1 for analog input 1 Bit 1 = 0x02: alarm 1 for channel 2 => alarm 1 for analog input 2 Bit 2 = 0x04: alarm 1 for channel 3 => alarm 1 for analog input 3 Bit 3 = 0x08: alarm 1 for channel 4 => alarm 1 for analog input 4 Bit 4 = 0x10: alarm 2 for channel 1 => alarm 2 for analog input 1 Bit 5 = 0x20: alarm 2 for channel 2 => alarm 2 for analog input 2 Bit 6 = 0x40: alarm 2 for channel 3 => alarm 2 for analog input 3 Bit 7 = 0x80: alarm 2 for channel 4 => alarm 2 for analog input 4
0x0046	70	Bit field 8	R/O	Bit 0 = 0x01 : digital input
0x0047	71	Byte	R/O	Collective alarm
0x0052	82	Float	R/O	Analog input 1
0x0054	84	Float	R/O	Analog input 2
0x0056	86	Float	R/O	Analog input 3
0x0058	88	Float	R/O	Analog input 4

4-channel analog input module - limit value 1 and 2 for alarms of the analog variables 1 to 4

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x008C	140	Float	R/W	Limit value 1 for alarm of analog variable 1
0x008E	142	Float	R/W	Limit value 1 for alarm of analog variable 2
0x0090	144	Float	R/W	Limit value 1 for alarm of analog variable 3
0x0092	146	Float	R/W	Limit value 1 for alarm of analog variable 4

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0094	148	Float	R/W	Limit value 2 for alarm of analog variable 1
0x0096	150	Float	R/W	Limit value 2 for alarm of analog variable 2
0x0098	152	Float	R/W	Limit value 2 for alarm of analog variable 3
0x009A	154	Float	R/W	Limit value 2 for alarm of analog variable 4

4-channel analog input module, switching differential for alarms of the analog variables 1 to 4

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x009C	156	Float	R/W	Switching differential for alarm of analog variable 1
0x009E	158	Float	R/W	Switching differential for alarm of analog variable 2
0x00A0	160	Float	R/W	Switching differential for alarm of analog variable 3
0x00A2	162	Float	R/W	Switching differential for alarm of analog variable 4

7 Modbus address tables

7.3.4 Modbus addresses for each 8-channel analog input module

Digital signals from the central processing unit to the 8-channel analog input module

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0002	2	Bit field 64	R/O	Bit 0 = 0x01: signal suppression 1 (NV_SR01) Bit 1 = 0x02: signal suppression 2 (NV_SR02) Bit 2 = 0x04: signal suppression 3 (NV_SR03) Bit 3 = 0x08: signal suppression 4 (NV_SR04) Bit 4 = 0x10: signal suppression 5 (NV_SR05) Bit 5 = 0x20: signal suppression 6 (NV_SR06) Bit 6 = 0x40: signal suppression 7 (NV_SR07) Bit 7 = 0x80: signal suppression 8 (NV_SR08) Bit 8 = 0x100: signal suppression 9 (NV_SR09)

Digital signals from the 8-channel analog input module to the central processing unit

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0042	66	Bit field 16	R/O	Module error
0x0045	69	Bit field 16	R/O	Bit 0 = 0x01: alarm 1 of analog input 1 Bit 1 = 0x02: alarm 1 of analog input 2 Bit 2 = 0x04: alarm 1 of analog input 3 Bit 3 = 0x08: alarm 1 of analog input 4 Bit 4 = 0x10: alarm 1 of analog input 5 Bit 5 = 0x20: alarm 1 of analog input 6 Bit 6 = 0x40: alarm 1 of analog input 7 Bit 7 = 0x80: alarm 1 of analog input 8 Bit 8 = 0x100: alarm 2 of analog input 1 Bit 9 = 0x200: alarm 2 of analog input 2 Bit 10 = 0x400: alarm 2 of analog input 3 Bit 11 = 0x800: alarm 2 of analog input 4 Bit 12 = 0x1000: alarm 2 of analog input 5 Bit 13 = 0x2000: alarm 2 of analog input 6 Bit 14 = 0x4000: alarm 2 of analog input 7 Bit 15 = 0x8000: alarm 2 of analog input 8

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0046	70	Bit field 8	R/O	Bit 0 = 0x01: digital input
0x0047	71	Byte	R/O	Collective alarm
0x0050	80	Float	R/O	Analog input 1
0x0052	82	Float	R/O	Analog input 2
0x...	...	Float	R/O	Analog input ...
0x005E	94	Float	R/O	Analog input ´8

8-channel analog input module - limit value 1 and 2 for alarms of the analog variables 1 to 8

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x008A	138	Float	R/W	Limit value 1 for alarm of analog variable 1
0x008C	140	Float	R/W	Limit value 1 for alarm of analog variable 2
0x...	...	Float	R/W	Limit value 1 for alarm of analog variable ...
0x0098	152	Float	R/W	Limit value 1 for alarm of analog variable 8
0x009A	154	Float	R/W	Limit value 2 for alarm of analog variable 1
0x009C	156	Float	R/W	Limit value 2 for alarm of analog variable 2
0x...	...	Float	R/W	Limit value 2 for alarm of analog variable ...
0x00A8	168	Float	R/W	Limit value 2 for alarm of analog variable 8

8-channel analog input module, switching differential for alarms of the analog variables 1 to 8

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x00AA	170	Float	R/W	Switching differential for alarm of analog variable 1
0x00AC	172	Float	R/W	Switching differential for alarm of analog variable 2
0x...	...	Float	R/W	Switching differential for alarm of analog variable ...
0x00B8	184	Float	R/W	Switching differential for alarm of analog variable 8

7 Modbus address tables

7.3.5 Modbus addresses for each 12-channel digital input/output module

Process values from the central processing unit to the 12-channel digital input/output module

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0000	0	Bit field 32	R/O	Bit 6 = 0x40: digital output 1 Bit 7 = 0x80: digital output 2 Bit 8 = 0x100: digital output 3 Bit 9 = 0x200: digital output 4 Bit 10 = 0x400: digital output 5 Bit 11 = 0x800: digital output 6 Bit 12 = 0x1000: digital output 7 Bit 13 = 0x2000: digital output 8 Bit 14 = 0x4000: digital output 9 Bit 15 = 0x8000: digital output 10 Bit 16 = 0x10000: digital output 11 Bit 17 = 0x20000: digital output 12

Process values from the 12-channel digital input/output module of the central processing unit

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0002	2	Bit field 32	R/O	Bit 0 = 0x01: digital input 1 Bit 1 = 0x02: digital input 2 Bit 2 = 0x04: digital input 3 Bit 3 = 0x08: digital input 4 Bit 4 = 0x10: digital input 5 Bit 5 = 0x20: digital input 6 Bit 6 = 0x40: digital input 7 Bit 7 = 0x80: digital input 8 Bit 8 = 0x100: digital input 9 Bit 9 = 0x200: digital input 10 Bit 10 = 0x400: digital input 11 Bit 11 = 0x800: digital input 12

7 Modbus address tables

7.3.6 Modbus addresses for each relay module 4-channel

Process values from the central processing unit to the relay module 4-channel

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0000	0	Byte	R/O	Bit 4 = 0x10: relay 1 (NV_relay01) Bit 5 = 0x20: relay 2 (NV_relay02) Bit 6 = 0x40: relay 3 (NV_relay03) Bit 7 = 0x80: relay 4 (NV_relay04)

7.3.7 Modbus addresses for each analog output module 4-channel

Process values from the central processing unit to the analog output module 4-channel (as of system version 03)

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x0008	8	Float	R/O	Analog output 1
0x000A	10	Float	R/O	Analog output 2
0x000C	12	Float	R/O	Analog output 3
0x000E	14	Float	R/O	Analog output 4



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
Email: 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 62 50 29
Email: sales@jumo.co.uk
Internet: www.jumo.co.uk

JUMO Process Control, Inc.

6733 Myers Road
East Syracuse, NY 13057, USA

Phone: +1 315 437 5866
Fax: +1 315 437 5860
Email: info.us@jumo.net
Internet: www.jumousa.com

