JUMO mTRON T

Measuring, Control, and Automation System
Multifunction Panel 840

Interface Description
Modbus

70506000T92Z001K000

V3.00/EN/00575642

Contents

Contents

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

1.1.3 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.

1.2.4 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. *The search results are listed.*
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). *The presentation starts.*

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 multifunction panel features an LAN interface as standard. 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 available as an option (Com1 and Com2, 9-pin), which can be used as either RS232 or RS422/485 as preferred. Both serial interfaces can be operated with the Modbus protocol (Modbus RTU; master or slave).

NOTE!
The type designation on the multifunction panel'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 705060.0 or the installation instructions B 705060.4 (the installation instructions are included in the scope of delivery of the multifunction panel).

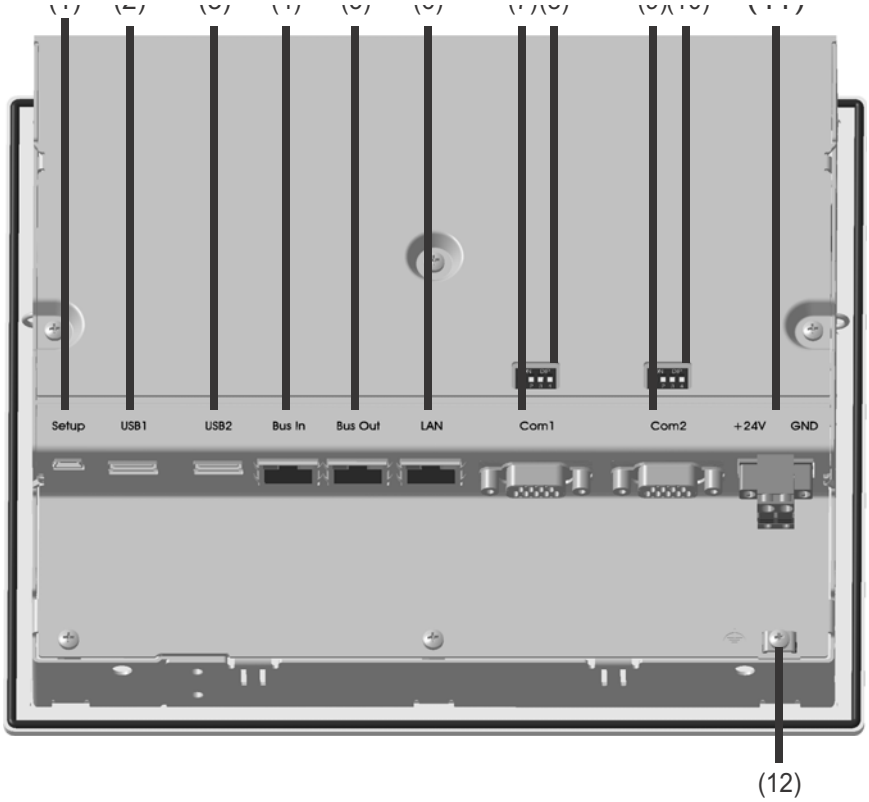
2.2 Display and control elements

(1) Front with decor foil

(2) Screen (touchscreen)

2 Connecting interfaces

2.3 Connection elements



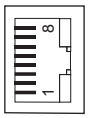
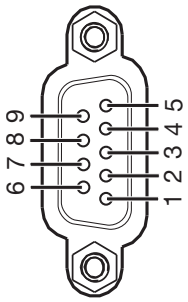
(1) USB device interface (setup)
(2) USB host interface 1
(3) USB host interface 2
(4) System bus In
(5) System bus Out
(6) LAN interface
(7) Com1 interface
(8) Com1 terminating resistor
(9) Com2 interface
(10) Com2 terminating resistor
(11) Voltage supply In, DC 24 V
(12) Functional grounding

CAUTION!
Functional grounding:
Connection terminal for functional ground.
To meet the specified EMC characteristics, this terminal must be connected to functional ground.

2.4 Use of the interfaces

Interface	Used for ...
USB device interface	• Setup program
USB host interface 1	• Connection of a USB memory stick
USB host interface 2	• Connection of a USB memory stick
System bus In	• Connection to base unit • Connection to router module
System bus Out	• Connection to router module
LAN interface	• Setup program • Web server • Mail server
Com1 interface	• Connection to Modbus master device • Connection to Modbus slave devices • Connection of a barcode reader • Connection of a modem
Com2 interface	• Connection to Modbus master device • Connection to Modbus slave devices • Connection of a barcode reader • Connection of a modem

2.5 Interface assignment

Connection	Description	Connection element		
Ethernet	LAN		1 TX+ 2 TX- 3 RX+ 6 RX-	Transmission data + Transmission data - Received data + Received data -
Serial interface (RS232)	Com1, Com2		2 RxD 3 TxD 5 GND	Received data Transmission data Ground
Serial interface (RS422)	Com1, Com2		3 TxD+ 4 RxD+ 5 GND 8 TxD- 9 RxD-	Transmission data + Received data + Ground Transmission data - Received data -
Serial interface (RS485)	Com1, Com2		3 TxD+/RxD+ 5 GND 8 TxD-/RxD-	Transmission/received data + Ground Transmission/received data -

2 Connecting interfaces

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.

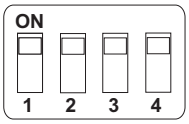
NOTE!
An RJ45 patch/crossover cable (CAT5 or higher) must be used to connect the LAN interface.

2.5.1 Terminating resistors

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

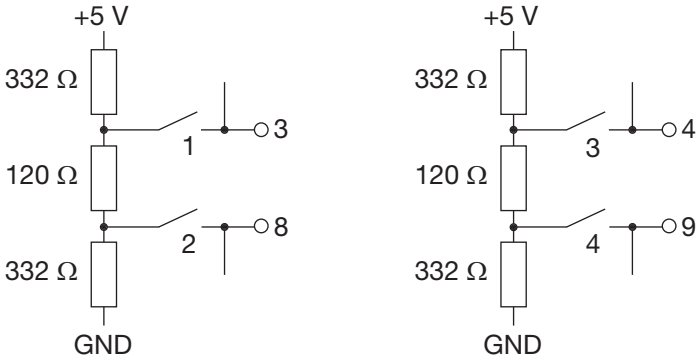
The terminating resistors are deactivated by default. To activate them, DIP switches 1 to 4 for the relevant interface must be pushed upward using a suitable tool such as a ballpoint pen (ON position).

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



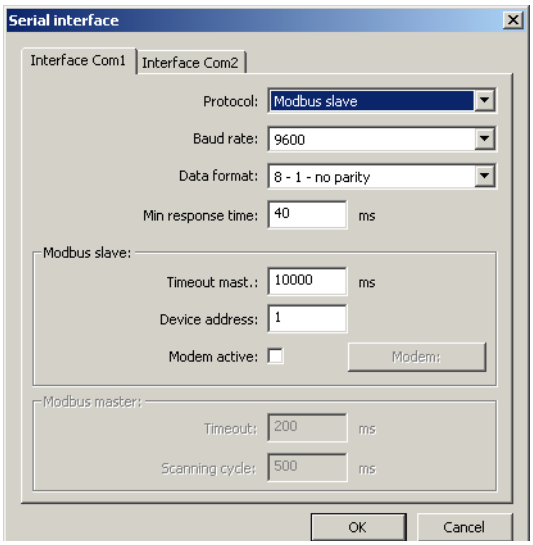
NOTE!
To ensure fault-free operation, terminating resistors are required at the start and end of an RS422/485 transmission path.

Internal terminating resistors



2.6 Serial interface

Setup dialog

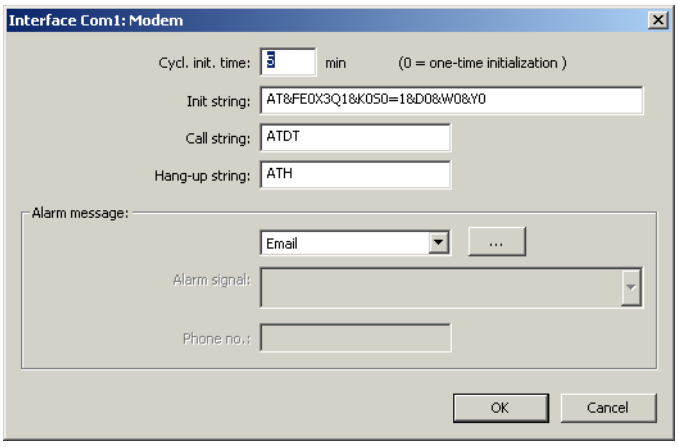


Parameter

Parameter	Selection/settings	Description
Protocol	**Modbus slave**	Modbus RTU (multifunction panel as Modbus slave)
	Modbus master	Modbus RTU (multifunction panel as Modbus master)
	Barcode	A barcode scanner should be operated on the interface.
Baud rate	Baud rate with which the interface is operated.	
	9600	9600 baud
	19200	19200 baud
	38400	38400 baud
Data format	Data format with which the interface is operated.	
	8 - 1 - no parity	8 data bits, 1 stop bit, no parity
	8 - 1 - odd parity	8 data bits, 1 stop bit, odd parity
	8 - 1 - even 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.

Parameter	Selection/settings	Description
Device address	**1** to 254	Device address The device address of the multifunction panel may only occur once within a connection for the interface type RS422/485 (multiple devices connected to a bus). This is of little importance for interface type RS232, as only one device may be connected to the serial interface.
Modem active	**No** (☐) Yes (☑)	No modem operation (Modbus slave is connected directly to the 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 data from the Modbus slave at these intervals.

Setup dialog



Parameter

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)

Parameter	Selection/settings	Description
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.
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	Output 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 Digital selector	No alarm message Alarm message is triggered by a signal (high active), which must be selected from the list of digital signals.
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 via a modem:
AT&FE0X3Q1&K0S0=1&D0&W0&Y0

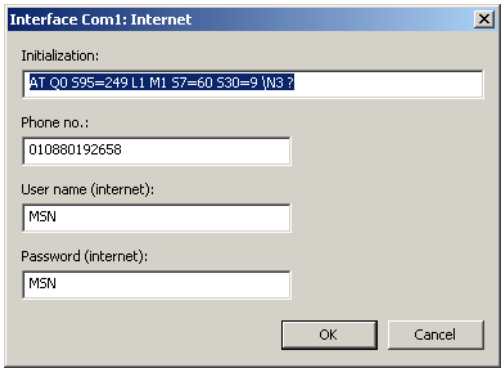
AT&F = Load current manufacturer profile
E0 = Switch off character echo
X3 = Switch off dial tone detection, activate busy tone detection
Q1 = Switch off command responses
&K0 = Switch off data flow control
S0=1 = Auto answer after first ring
&D0 = Ignore DTR signal
&W0 = Save current configuration as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

Call string, Hang-up string

The call string and hang-up string 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:



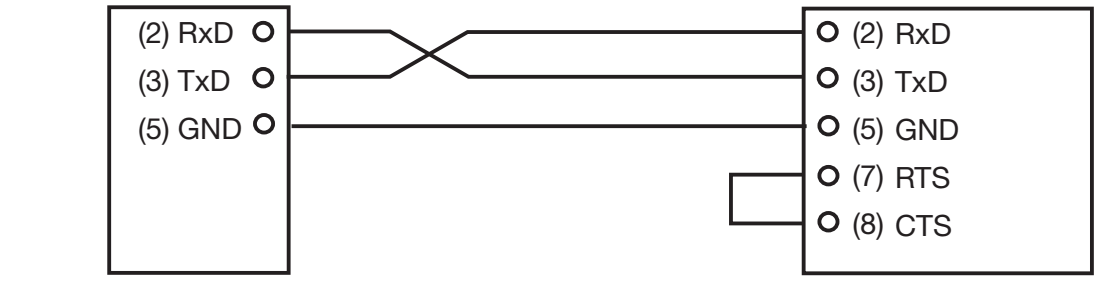
Parameter

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 name (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)

2.6.1 Handshake lines for RS232

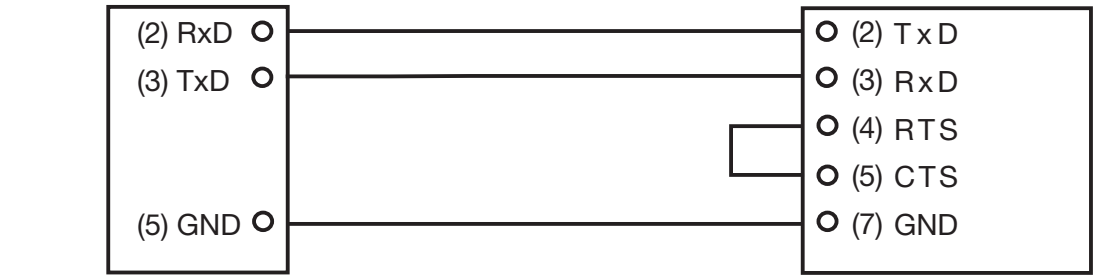
When using Com1/Com2 as an RS232 interface, the handshake lines (RTS, CTS) are not used. The RTS line coming from the master is ignored. The response is immediately transmitted from the slave. The CTS line of the master remains open.
If the program used evaluates the handshake lines, they must be bridged in the cable.

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



Com1/Com2 PC/Notebook

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



Com1/Com2 PC/Notebook

2 Connecting interfaces

2.7 Ethernet interface

General information

To use the Ethernet interface, a patch/crossover cable is required that is equipped with an RJ45 plug. The Ethernet interface is configured in the setup program.
Required parameters, such as DHCP functionality, IP address, subnet mask, gateway address, DNS device name, DNS server, and transfer rate, can be configured in the setup program under **PROJECT NAME** > **HMI** > **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

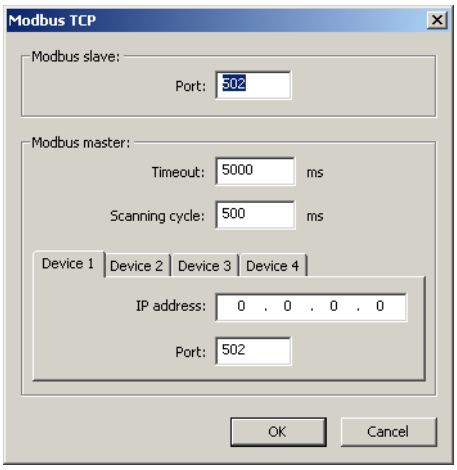
Speed	**Mode**
Auto negation	Default setting
10 Mbit/s	Half duplex
10 Mbit/s	Full duplex
100 Mbit/s	Half duplex
100 Mbit/s	Full duplex

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 multifunction panel 840 (B 705060.0).

2.8 Ethernet settings for Modbus/TCP

This menu is used to implement settings for the Modbus/TCP operating mode. If the multifunction panel functions as a Modbus master, it can communicate with up to four external devices (Modbus slaves; device 1 to 4). If it functions as a Modbus slave, two external devices (Modbus masters) can access the multifunction panel at the same time.

Setup dialog



Parameter

Parameter	Selection/settings	Description
Modbus slave (multifunction panel 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 (multifunction panel as Modbus master)		
Timeout	4000 to **5000** 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 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

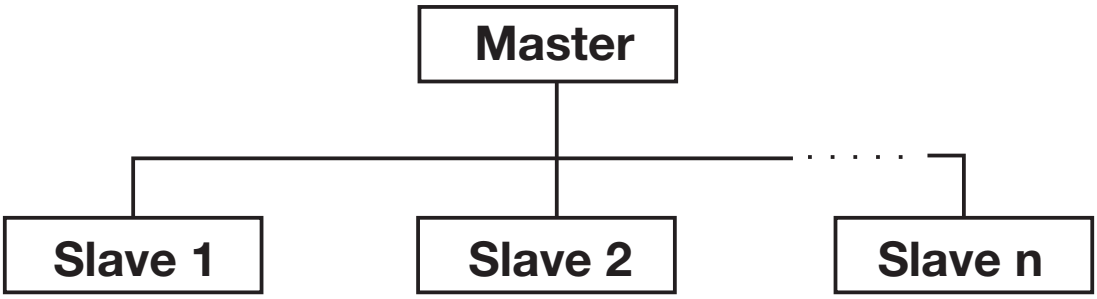
NOTE!
To ensure that fixed IP addresses are used, DHCP must be deactivated in the devices involved, if applicable.

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.

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 multifunction panel 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 multifunction panel (slave) as well as actively have the multifunction panel (master) import them from one or several slaves.

If the multifunction panel 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, 16, or 32 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

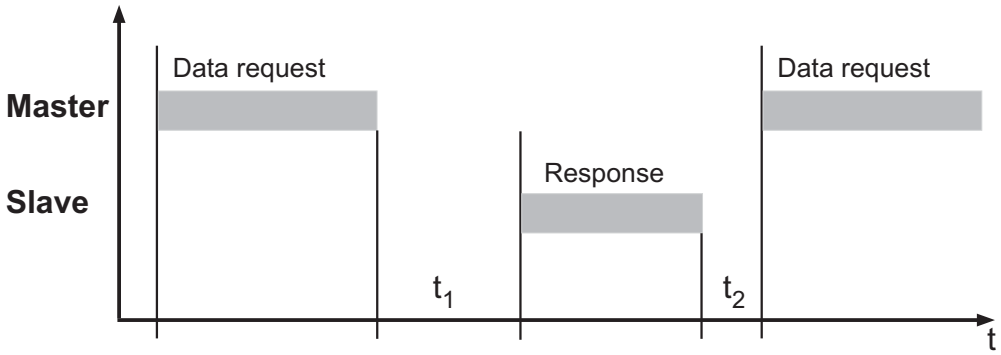
The character transmission time (time taken to transmit one character with 8 data bits) depends on the baud rate and the number of bits of the character (see table for data format):

Character transmission time [ms] = 1000 × number of bits / 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 t1 and t2 also contain the end identifier (3.5 × character transmission time), which
follows after each data request or response.

NOTE!
A minimum response time can be set in the multifunction panel via the setup program, under **HMI** > **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 multifunction panel answers as soon as internal processing is completed. If a time of 0 ms is set, the device will answer at the maximum possible speed.
The minimum response time that 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 needed with 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 Modbus protocol description

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 the following 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

3.5 Device address

The device address can be set to between 1_{DEC} and 254_{DEC}. Each Modbus station must have a unique device address. For interface connectors inserted directly at the multifunction panel, the device address is assigned in the setup program under **HMI > CONFIGURATION LEVEL > SERIAL INTERFACE > MODBUS SLAVE**.

A subset of the multifunction panel data can also be requested via an interface connector inserted at the central processing unit. For all modules visible from the central processing unit, the device addresses are assigned in the setup program under **PROJECT > HARDWARE ARRANGEMENT**.

⇨ Modbus interface description for the central processing unit B 705001.2.0

NOTE!
If the multifunction panel is controlled directly via its IP, only data specific to the multifunction panel can be exchanged. Communication with other modules is not possible in such a case. In this case the device address of the multifunction panel is set to 255.

The following data exchange variants 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 accessed slave responds.

Broadcast

The broadcast is an instruction from the master to all slaves via the device address 0 (e.g. for transferring a certain 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. Data request with the device address 0 does not make sense.

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, and for writing specific data.

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

NOTE!
A hexadecimal number is marked by a preceding "0x".
Example: 0x0010 (= 16_{DEC})

NOTE!
If the multifunction panel does not respond to these functions or outputs an error code, it can be evaluated.

⇨ Chapter 3.9 "Error messages", page 42

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 75

The word addresses are specified in hexadecimal form in the Modbus address tables in the left column. The bit address 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 $_{DEC}$

3.6.1 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 the Modbus interface description for the central processing unit (B 705001.2.0), chapter 7.3.5 "Modbus addresses for each 12-channel digital input/output module", page 100, the word address here is 0x0002. The values in bits 1 and 2 are included.

Hexadecimal calculation of bit address for bit 1	**Decimal calculation of bit address for bit 1**
Bit address = (0x0002 × 0x10) + 0x1	Bit address = (2 × 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 × 0x10) + 0x2	Bit address = (2 × 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:

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, the Modbus address of which is 7 in the example here.
According to the Modbus interface description for the central processing unit (B 705001.2.0), chapter 7.3.1 "Modbus addresses for each HMI module", page 83, the word address here is 0x0002. The values in bits 0 to 53 are included:

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. Then the word with the next highest value until the lowest value 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 word	...	...	Lowest-value word
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 word
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 word
00	0	0000 0000	High byte	

This analysis and arrangement from the table yields:

- first low byte (highest-value 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-value 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.

3.6.2 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

Reading of the IP address of the multifunction panel. 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.

Data request:

01	03	21 9C	00 04	8E 1B
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				

3.6.3 Writing 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

3.6.4 Writing 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

Writing of ext. binary value 1 at word address 0x15F0 to value 1.

⇨ Chapter 7.2 "Multifunction panel addresses", page 76

Instruction:

01	06	15 F0	00 01	4C 35
Slave	Function	Word address	Value	CRC

Response:

01	06	15 F0	00 01	4C 35
Slave	Function	Word address	Value	CRC

3.6.5 Writing 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 of the word "test" (ASCII encoding: 0x54 0x65 0x73 0x74 0x00) to external batch text 1 from word address 0x1209.

⇨ Chapter 7.2 "Multifunction panel addresses", page 76

Instruction:

01	10	12 09	00 03	06	54 65 73 74 00 00	95 72
Slave	Function	Address of the 1st word	Number of words	Number of bytes	Text in ASCII	CRC

Response:

01	10	12 09	00 03	55 72
Slave	Function	Address of the 1st word	Number of words	CRC

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

3.7.2 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

SEEEEEEE	EMMMMMMM	MMMMMMMM	MMMMMMMM
Byte 1	Byte 2	Byte 3	Byte 4

S - sign bit

E - exponent (two's complement)

M - 23 bits normalized mantissa

Modbus float format

Modbus address x		**Modbus address x+1**	
MMMMMMMM	MMMMMMMM	SEEEEEEE	EMMMMMMM
Byte 3	Byte 4	Byte 1	Byte 2

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:

01	03	10 71	00 02	05 D4
Slave	Function	Address of the 1st word	Number of words	CRC

Response (values in the Modbus float format):

01	03	04	80 00	44 09	F5 20
Slave	Function	Bytes read	Float value		CRC

Once transmission from the multifunction panel 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

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

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

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

SEEEEEEE	EEEEMMMM	MMMMMMMM	MMMMMMMM
Byte 1	Byte 2	Byte 3	Byte 4

MMMMMMMM	MMMMMMMM	MMMMMMMM	MMMMMMMM
Byte 5	Byte 6	Byte 7	Byte 8

S - sign bit

E - exponent (two's complement)

M - 52 bits normalized mantissa

Modbus double format

Modbus address x		**Modbus address x+1**	
SEEEEEEE	EEEEMMMM	MMMMMMMM	MMMMMMMM
Byte 1	Byte 2	Byte 3	Byte 4

Modbus address x+2		**Modbus address x+3**	
MMMMMMMM	MMMMMMMM	MMMMMMMM	MMMMMMMM
Byte 5	Byte 6	Byte 7	Byte 8

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.

⇨ Modbus interface description for the central processing unit (B 705001.2.0), "Modbus address tables" chapter

Example

Requesting of 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".

3.8 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 CRC = CRC XOR 0xA0001	else
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 malfunction

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

The function code is ORed with 0x80. This sets the most significant bit (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.

3.9.2 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 (=3.0×10^{37}) supplied by analog input 1 indicates that the input value is invalid.

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: keypad and program lock	
26	Keypad locked
27	Programming locked
28	Write error in the ser. EEPROM (calib)
29	Hardware error: MANUAL + AUTO locked

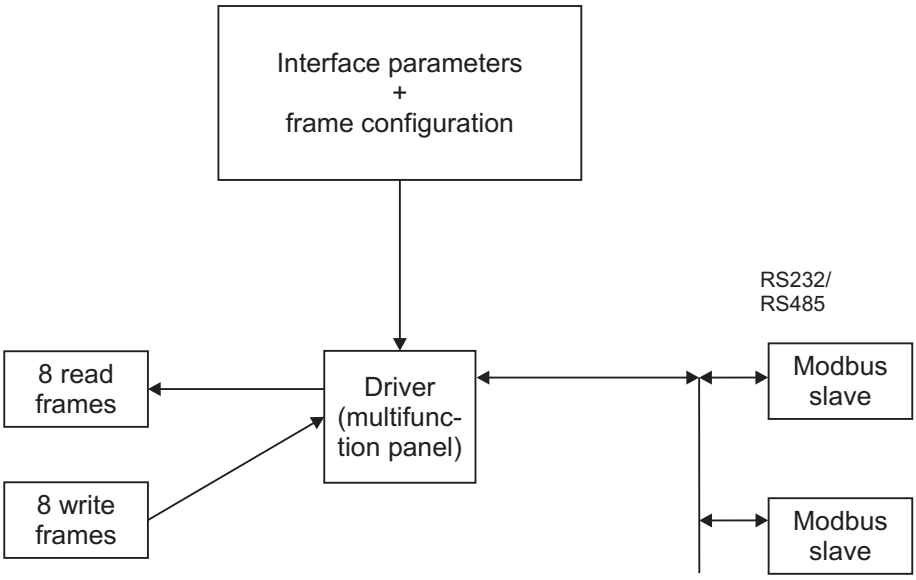
Error code	Description
30	Edit is inadmissible when the program is active
31	Copy is inadmissible when the program is active
32	MANUAL is inadmissible during AUTO lead time
33	Segment change! Image update required
34	No DB number, image update by PLC
35	No DB number for process values of PLC
36	Printer in use 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

Error code	Description
72	Message pool cannot be generated
73	Memory from message pool cannot be requested
74	Message cannot be transmitted
Error list: processing MQX functions	
80	Task creation failed
81	Hardware timer not created
Error list: flash processing	
90	Flash memory 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

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	Received as client DNS request
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

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 uninstalling the partition manager
203	Error when uninstalling the MFS file system

4.1 Modbus master operation via serial interface



If the multifunction panel 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.

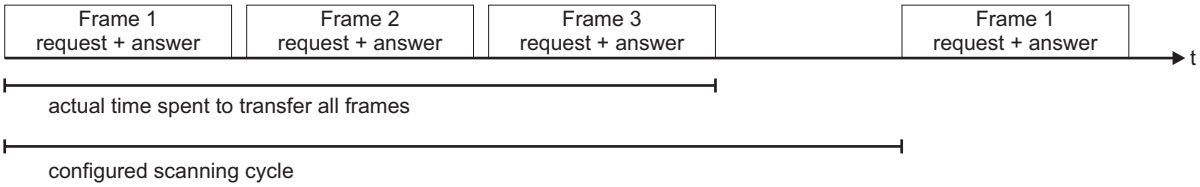
Number of external inputs during Modbus master operation

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

- 54 float values, 16 integer values, and 54 binary values, as well as
- 9 batch texts (27 texts as of system version 02; 90 texts as of system version 04).

Chronological sequence

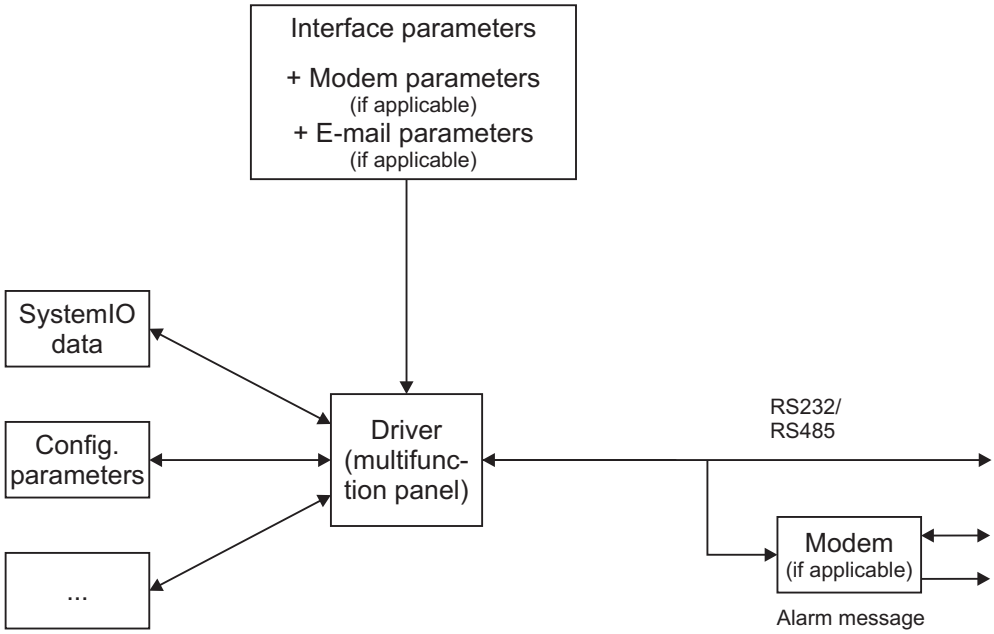
The respective interface searches for all frames configured for it (setup program: **HMI** > **SETUP ONLY** > **MODBUS FRAMES FOR READING/WRITING** > menu item **INTERFACE**) and cyclically transmits them in sequence to frame 1, frame 2, and frame 3 as displayed below.



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 multifunction panel 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 multifunction panel 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 75

4.3 RS232 and RS422/485

The serial interface is implemented using two different additional cards for the multifunction panel. The interface board type used, RS232 or RS422/485, is automatically detected by the multifunction panel 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 75

Example: reading n words

Reading of the IP address of the multifunction panel. 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:

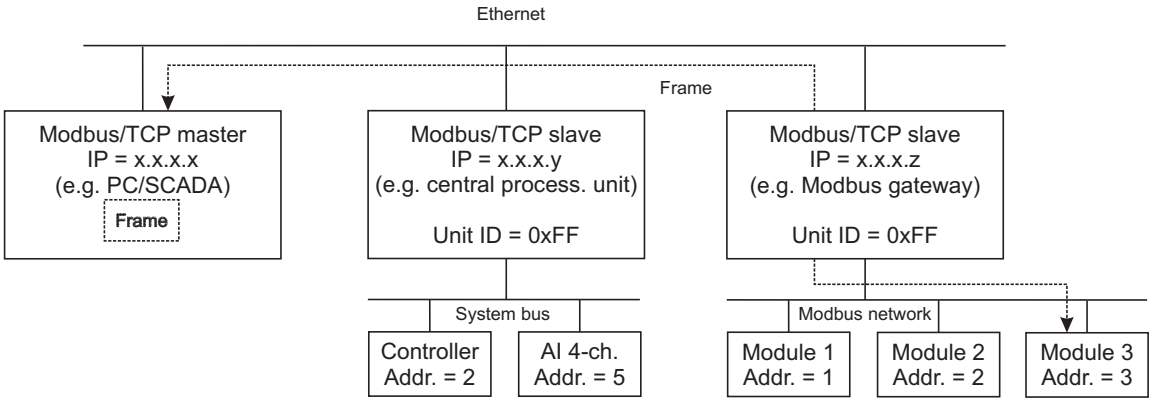
MBAP header				**Modbus telegram (without slave address and CRC)**		
00 01	00 00	00 06	FF	03	21 9C	00 04
2-byte transaction ID	2-byte protocol ID	2-byte length	1-byte unit ID	1-byte function code	2-byte address first word	2-byte number of words
Response assignment to the request (consecutive numbering)	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

5.2 Networking with Modbus/TCP

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



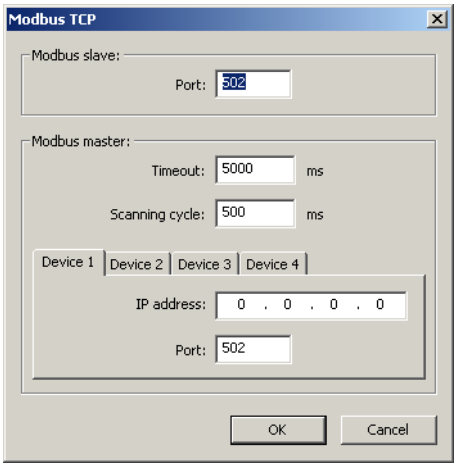
It is possible to directly address every Modbus/TCP node via an IP address. The "unit ID" (= device address) also contained in the protocol must be 0xFF for the node directly attached to the network. However, a node can present itself virtually as several subordinate devices, including the central processing unit (in centre of image) for all connected modules. These can then be accessed with device addresses 1 to 254 (assigned as in the setup program, under **HARDWARE ARRANGEMENT**).

No subordinate modules can be accessed from the multifunction panel. Via Modbus/TCP, the multifunction panel can therefore only be addressed with device address = 0xFF!

5.3 Modbus master with Modbus/TCP

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

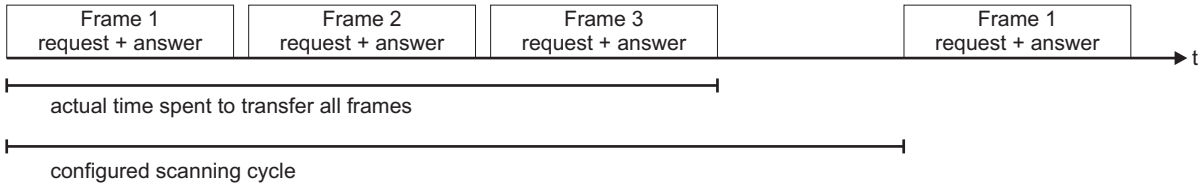
⇨ Chapter 4 "Serial transmission modes", page 49



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.

Chronological sequence

The respective interface searches for all frames configured for it (setup program: **HMI** > **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 multifunction panel 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 multifunction panel 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 75

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

5.6 Browser connection and Web server

The multifunction panel can also be accessed by a browser using the HTTP protocol. The required URL for this is the IP address of the multifunction panel (in the above example 10.13.6.233). 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 to the multifunction panel via the setup program, under **SETUP NAME** > **HMI** > **CONFIGURATION LEVEL** > **WEB SERVER**. The Microsoft plugin "Silverlight" is required for the browser in order to use the index.htm supplied ex-works.

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 multifunction panel's operating manual B 705060.0.

5.7 E-mail (SMTP and POP3)

The multifunction panel 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).

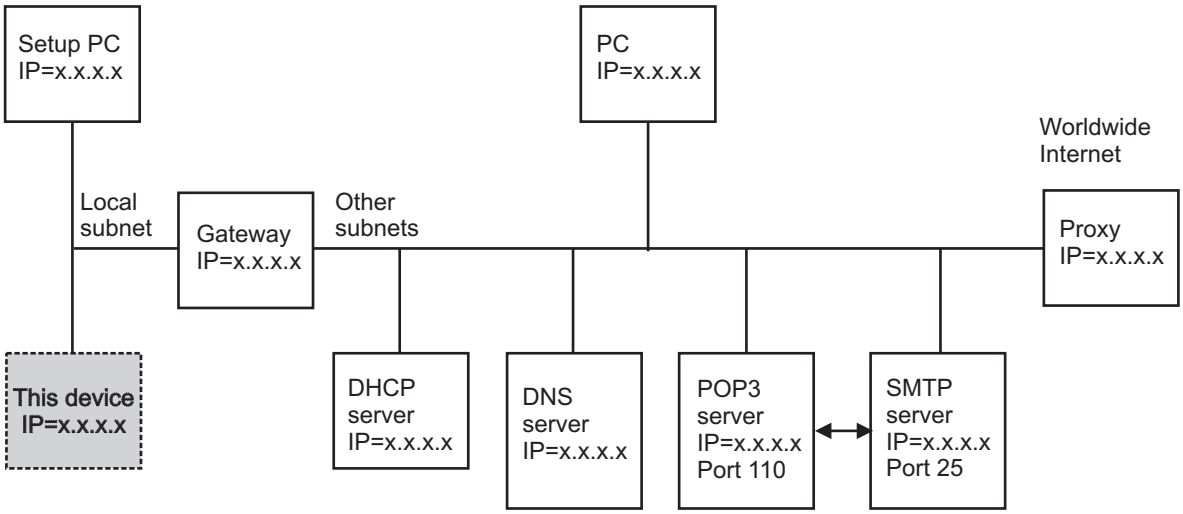


Figure 1: Typical networking in the company network

Function of the individual nodes

Gateway:

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

DHCP server:

can automatically assign an IP address, subnet mask, and gateway address to other nodes when switching on. These parameters can also be entered manually, in which case a DHCP server.

DNS server:

changes symbolic names into IP addresses. Example: the request "www.name.de" will generate the response "www.name.de has IP=10.12.32.45".

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.

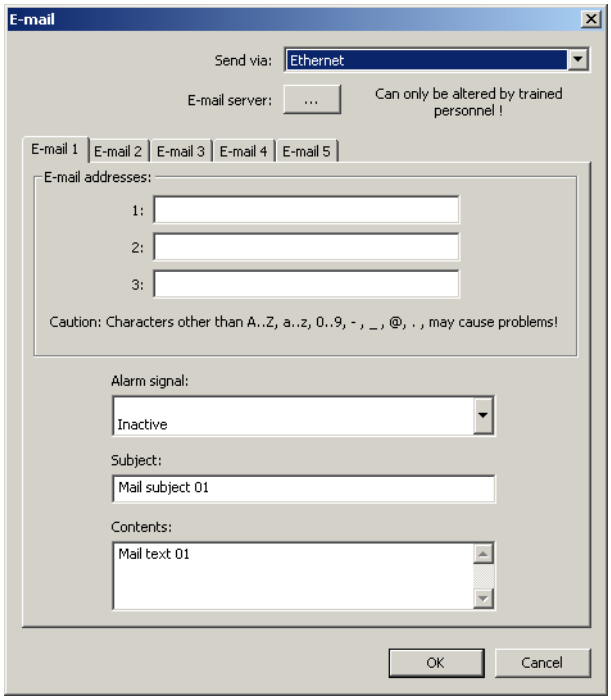
[1] As of system version 05: SMTP port number changeable

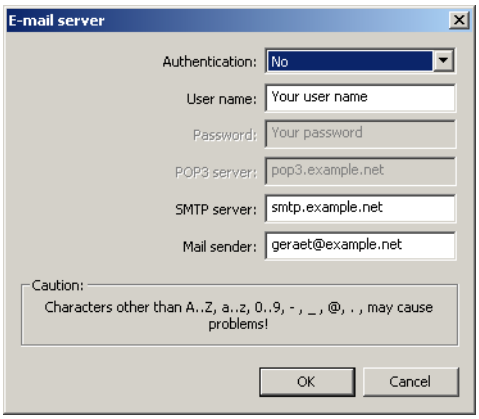
Proxy:

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 relevant screens can be accessed via **HMI** > **SETUP ONLY** > **E-MAIL**.





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.

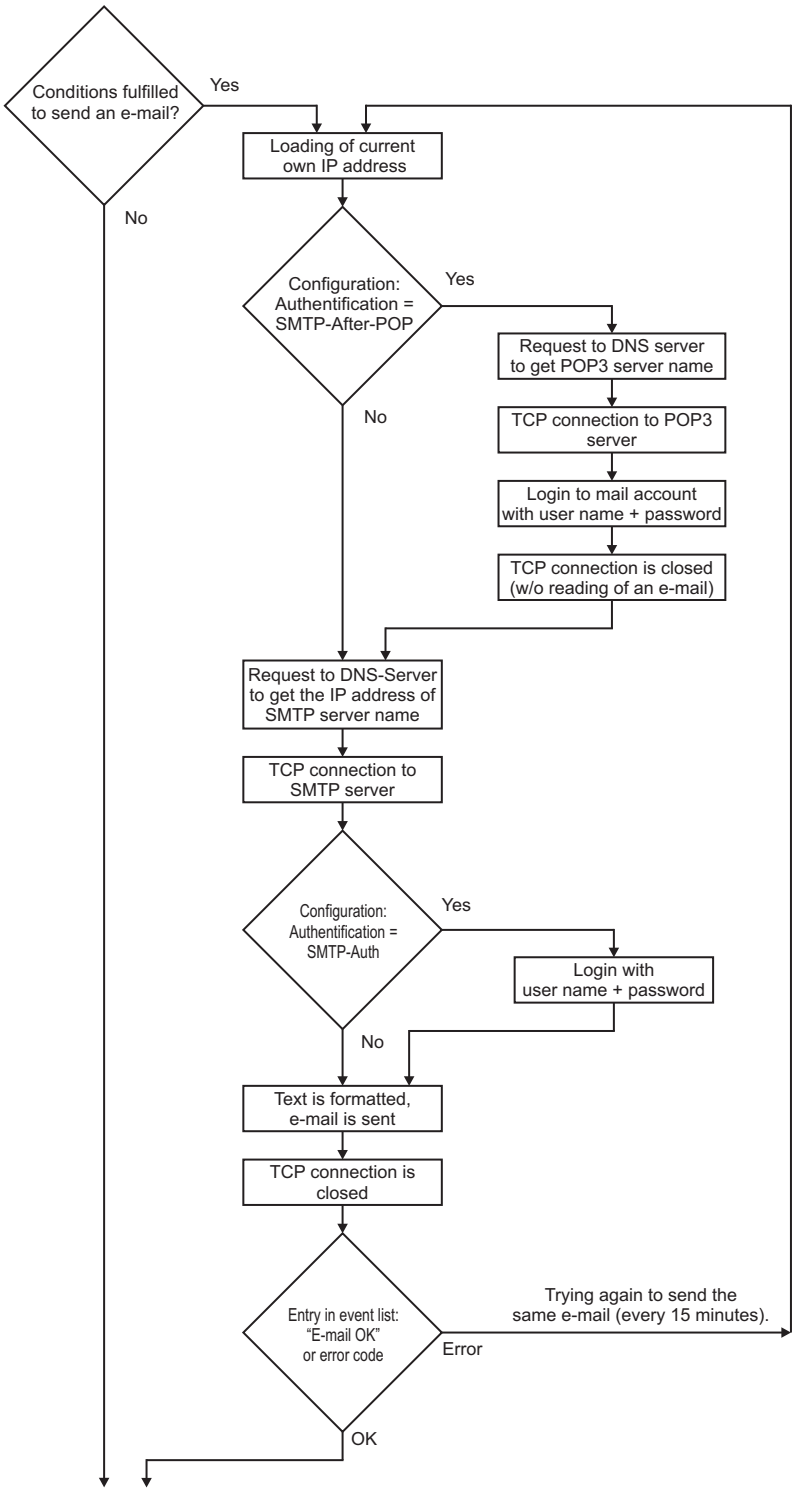
5 Ethernet transmission modes

NOTE!
Information on configuring the e-mail functionality can be found in the multifunction panel's operating manual B 705060.0.

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


Conditions fulfilled to send an e-mail?
Yes
No
Loading of current own IP address
Configuration: Authentification = SMTP-After-POP
Yes
No
Request to DNS server to get POP3 server name
TCP connection to POP3 server
Login to mail account with user name + password
TCP connection is closed (w/o reading of an e-mail)
Request to DNS-Server to get the IP address of SMTP server name
TCP connection to SMTP server
Configuration: Authentification = SMTP-Auth
Yes
No
Login with user name + password
Text is formatted, e-mail is sent
TCP connection is closed
Entry in event list: "E-mail OK" or error code
Trying again to send the same e-mail (every 15 minutes).
Error
OK

5 Ethernet transmission modes

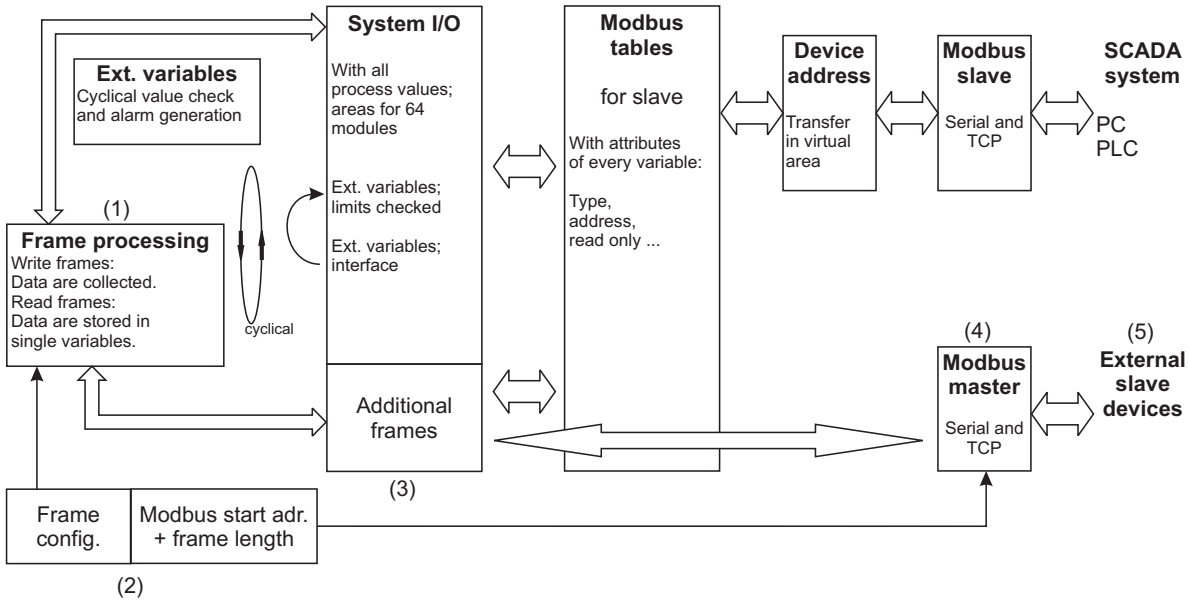
6.1 General information

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

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

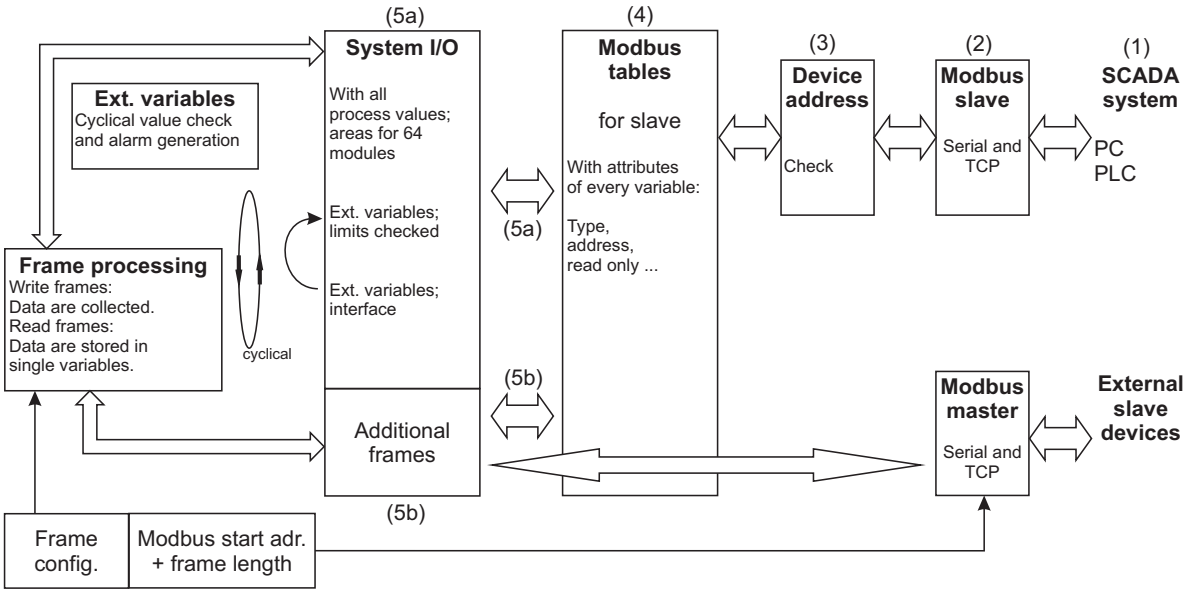
6 User frames

NOTE!
When actively 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 device address is checked for validity.
(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.

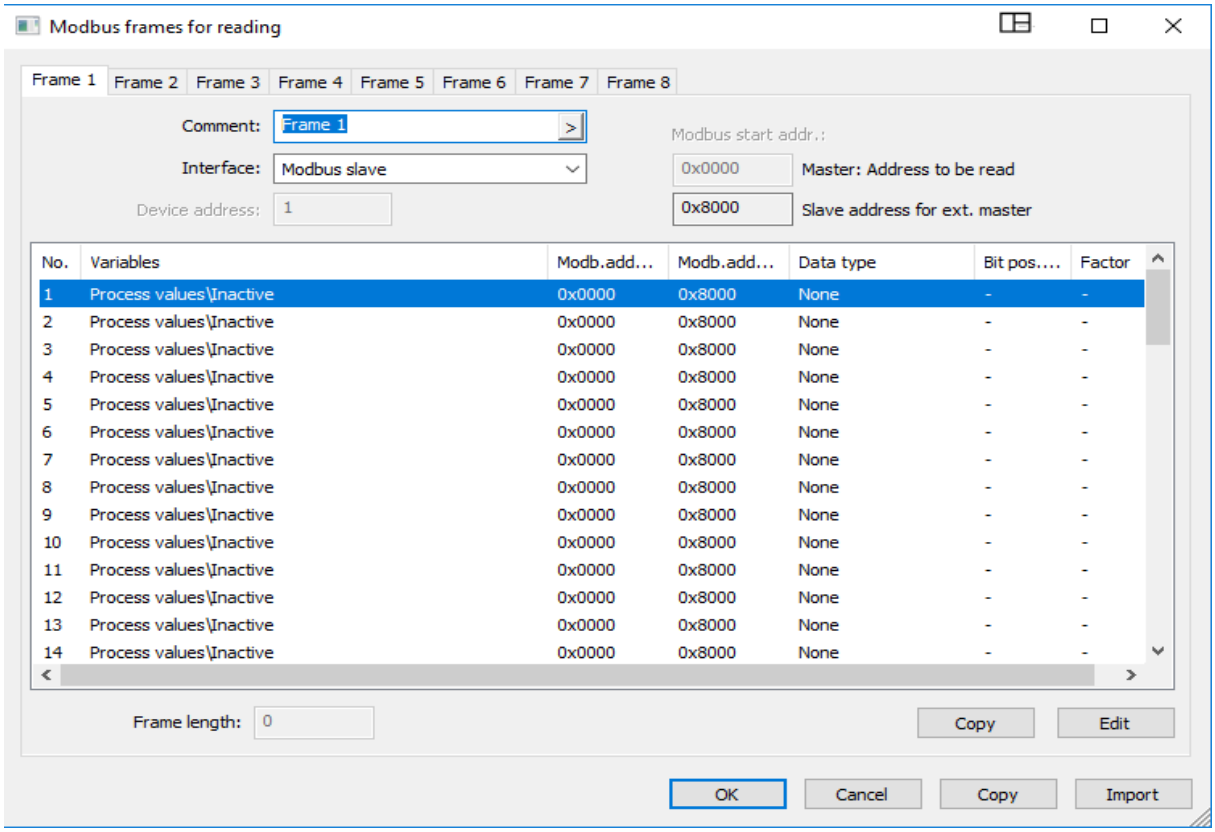
6.3 Compiling Modbus frames

The "Modbus frames for reading" and "Modbus frames for writing" options are found under **SETUP NAME** > **HMI** > **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



6 User frames

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	**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 (multifunction panel 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 multifunction panel 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))

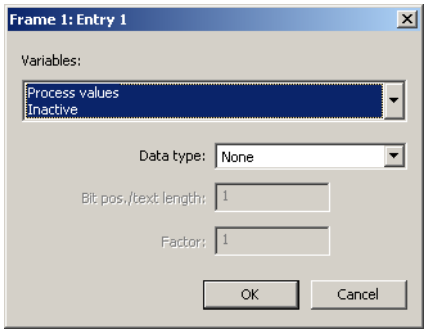
Copy (entries)

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 character per word
	Text (2 char./word)	Text, 2 characters per word
Bit pos./text length	Integer (1 byte): **0** to 7 Integer (2 bytes): **0** to 15 Integer (4 bytes): **0** to 31 Text (1 char./word): **1** to 127 Text (2 char./word): **1** to 254	The setting range of the bit position or text length depends on the selected data type. (This parameter is not active for the "float" data type.)

Parameter	Selection/settings	Description
Factor	Using the factor makes it possible to transmit floating point values in the integer format. The transmitter multiplies the data with the corresponding factor before transmission. The data must be divided by the same value in the receiver.	
	Complete 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)

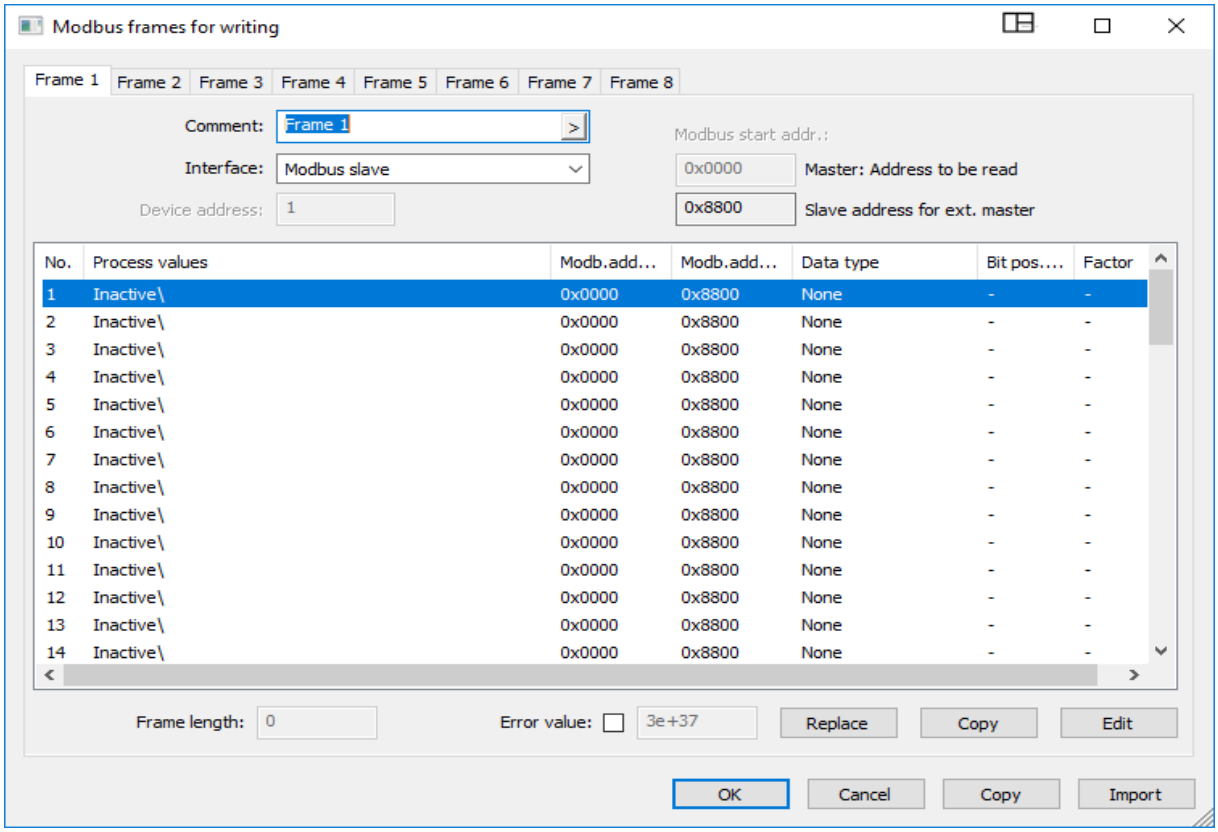
With this function Modbus frames can be imported from another project. First, the setup file from which you want to import is temporarily opened. Then all frames to be copied have to be selected (individual frames or all). As a result, all entries of the relevant frames are transferred into your own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

Parameter	Selection/settings	Description
Device address	0 to **1** to 255	Device address of the external device (Modbus slave). With writing frames, the 0 is permitted as the device address.
Modbus start address (Master)	**0x0000** to 0xFFFF	Modbus start address (offset) of the external slave device (multifunction panel 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 multifunction panel 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 a malfunction 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.

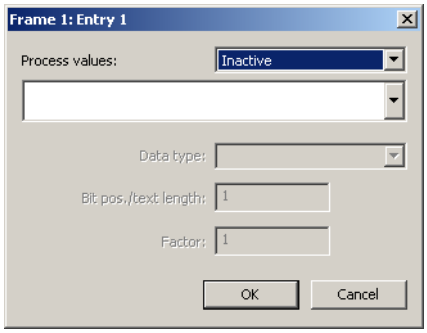
Copy (entries)

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 multifunction panel (incl. variables and PLC)	
	Inactive	No module selected
	Select module (process value source)	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, 1 character per word
	Text (2 char./word)	Text, 2 characters per word
Bit pos./text length	Integer (1 byte): **0** to 7 Integer (2 bytes): **0** to 15 Integer (4 bytes): **0** to 31 Text (1 char./word): **1** to 127 Text (2 char./word): **1** to 254	The setting range of the bit position or text length depends on the selected data type. (This parameter is not active for the "float" data type.)

Parameter	Selection/settings	Description
Factor	Using the factor makes it possible to transmit floating point values in the integer format. The transmitter multiplies the data with the corresponding factor before transmission. The data must be divided by the same value in the receiver.	
	Complete 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)

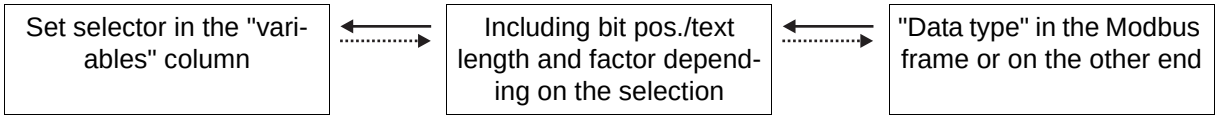
With this function Modbus frames within a project can be copied.

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 multifunction panel and the data format on the other end. The "variables" column (see above screenshot) determines the data format within the multifunction panel; 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

NOTE!
These tables are important for external devices that access the multifunction panel as a Modbus master (configured as a Modbus slave). Alternatively, external devices can also access the multifunction panel using the Modbus frames.

NOTE!
When the multifunction panel is accessed directly via its IP address (Modbus slave), it can be reached via the device address 255. The Modbus address given in the hardware arrangement for the multifunction panel is only valid if the multifunction panel is accessed via the central processing unit.

7.1 Data types and access types

Data types

Bit x	Bit No. x (bit 0 is the bit with the lowest value)
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) according 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 Modbus address tables

7.2 Multifunction panel addresses

The following tables contain important process data and device data for the multifunction panel including address, data type, and type of access.

Cycle times and device names

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
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

Multifunction panel process values

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x103F	4156	Byte	R/O	Software version for optional board 1
0x1040	4160	Byte	R/O	Software version for optional board 2
0x1041	4161	Word	R/O	Time: year
0x1042	4162	Word	R/O	Time: month
0x1043	4163	Word	R/O	Time: day
0x1044	4164	Word	R/O	Time: hour
0x1045	4165	Word	R/O	Time: minute
0x1046	4166	Word	R/O	Time: second
0x1047	4167	Float	R/W	External actual value, interface 1
0x1049	4169	Float	R/W	External actual value, interface 2
0x...	...	Float	R/W	External actual value, interface...
0x10B1	4273	Float	R/W	External actual value, interface 54
0x111F	4383	Boolean	R/O	External analog alarm, min 1
0x1120	4384	Boolean	R/O	External analog alarm, min 2
0x...	...	Boolean	R/O	External analog alarm, min...
0x1154	4436	Boolean	R/O	External analog alarm, min 54

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x1155	4437	Boolean	R/O	External analog alarm, max 1
0x1156	4438	Boolean	R/O	External analog alarm, max 2
0x...	...	Boolean	R/O	External analog alarm, max...
0x118A	4490	Boolean	R/O	External analog alarm, max 54
0x118B	4491	Int32	R/W	External integer 1
0x118D	4493	Int32	R/W	External integer 2
0x...	...	Int32	R/W	External integer...
0x11A9	4521	Int32	R/W	External integer 16
0x11AB	4523	Boolean	R/O	External integer alarm, min 1
0x11AC	4524	Boolean	R/O	External integer alarm, min 2
0x...	...	Boolean	R/O	External integer alarm, min...
0x11BA	4538	Boolean	R/O	External integer alarm, min 16
0x11BB	4539	Boolean	R/O	External integer alarm, max 1
0x11BC	4540	Boolean	R/O	External integer alarm, max 2
0x...	...	Boolean	R/O	External integer alarm...
0x11CA	4554	Boolean	R/O	External integer alarm, max 16
0x11CB	4555	Boolean	R/O	Alarm output 1 of counter 1
0x11CC	4556	Boolean	R/O	Alarm output 1 of counter 2
0x...	...	Boolean	R/O	Alarm output 1 of counter...
0x11E5	4581	Boolean	R/O	Alarm output 1 of counter 27
0x11E6	4582	Boolean	R/O	Alarm output 2 of counter 1
0x11E7	4583	Boolean	R/O	Alarm output 2 of counter 2
0x...	...	Boolean	R/O	Alarm output 2 of counter...
0x1200	4608	Boolean	R/O	Alarm output 2 of counter 27
0x1201	4609	Boolean	R/O	Collective alarm
0x1202	4610	Boolean	R/O	Memory alarm of the internal flash memory
0x1203	4611	Boolean	R/O	Memory alarm of the interface
0x1204	4612	Boolean	R/O	Fault

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x1209	4617	Char[64]	R/W	External batch text 1
0x1229	4649	Char[64]	R/W	External batch text 2
0x...	...	Char[64]	R/W	External batch text ...
0x1309	4873	Char[64]	R/W	External batch text 9
				As of system version 02:
0x232C	9004	Char[64]	R/W	External batch text 10
0x234C	9036	Char[64]	R/W	External batch text 11
0x...	...	Char[64]	R/W	External batch text ...
0x254C	9548	Char[64]	R/W	External batch text 27
				As of system version 04:
0x27AC	10156	Char[64]	R/W	External batch text 1
0x27CC	10188	Char[64]	R/W	External batch text 2
0x...	...	Char[64]	R/W	External batch text ...
0x2F6C	12140	Char[64]	R/W	External batch text 90
0x1449	5193	Char[94]	R/W	External event text for group 1
0x1478	5240	Char[94]	R/W	External event text for group 2
0x...	...	Char[94]	R/W	External event text for group...
0x15C1	5569	Char[94]	R/W	External event text for group 9
0x15F0	5616	Boolean	R/W	External binary value 1
0x15F1	5617	Boolean	R/W	External binary value 2
0x...	...	Boolean	R/W	External binary value...
0x1625	5669	Boolean	R/W	External binary value 54
0x1626	5670	Boolean	R/O	External binary alarm 1
0x1627	5671	Boolean	R/O	External binary alarm 2
0x...	...	Boolean	R/O	External binary alarm...
0x165B	5723	Boolean	R/O	External binary alarm 54
0x165C	5724	Char[64]	R/O	Current batch mask 1: right line 1
0x167C	5756	Char[64]	R/O	Current batch mask 1: right line 2
0x...	...	Char[64]	R/O	Current batch mask 1: right line...
0x177C	6012	Char[64]	R/O	Current batch mask 1: right line 10
0x179C	6044	Char[64]	R/O	Current batch mask 2: right line 1
0x17BC	6076	Char[64]	R/O	Current batch mask 2: right line 2

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x...	...	Char[64]	R/O	Current batch mask 2: right line...
0x18BC	6332	Char[64]	R/O	Current batch mask 2: right line 10
0x18DC	6364	Char[64]	R/O	Current batch mask 3: right line 1
0x18FC	6396	Char[64]	R/O	Current batch mask 3: right line 2
0x...	...	Char[64]	R/O	Current batch mask 3: right line...
0x19FC	6652	Char[64]	R/O	Current batch mask 3: right line 10
0x1A1C	6684	Char[64]	R/O	Current batch mask 4: right line 1
0x1A3C	6716	Char[64]	R/O	Current batch mask 4: right line 2
0x...	...	Char[64]	R/O	Current batch mask 4: right line...
0x1B3C	6972	Char[64]	R/O	Current batch mask 4: right line 10
0x1B5C	7004	Char[64]	R/O	Current batch mask 5: right line 1
0x1B7C	7036	Char[64]	R/O	Current batch mask 5: right line 2
0x...	...	Char[64]	R/O	Current batch mask 5: right line...
0x1C7C	7292	Char[64]	R/O	Current batch mask 5: right line 10
0x1C9C	7324	Char[64]	R/O	Current batch mask 6: right line 1
0x1CBC	7356	Char[64]	R/O	Current batch mask 6: right line 2
0x...	...	Char[64]	R/O	Current batch mask 6: right line...
0x1DBC	7612	Char[64]	R/O	Current batch mask 6: right line 10
0x1DDC	7644	Char[64]	R/O	Current batch mask 7: right line 1
0x1DFC	7676	Char[64]	R/O	Current batch mask 7: right line 2
0x...	...	Char[64]	R/O	Current batch mask 7: right line...
0x1EFC	7932	Char[64]	R/O	Current batch mask 7: right line 10
0x1F1C	7964	Char[64]	R/O	Current batch mask 8: right line 1
0x1F3C	7996	Char[64]	R/O	Current batch mask 8: right line 2
0x...	...	Char[64]	R/O	Current batch mask 8: right line...
0x203C	8252	Char[64]	R/O	Current batch mask 8: right line 10
0x205C	8284	Char[64]	R/O	Current batch mask 9: right line 1
0x207C	8316	Char[64]	R/O	Current batch mask 9: right line 2
0x...	...	Char[64]	R/O	Current batch mask 9: right line...
0x217C	8572	Char[64]	R/O	Current batch mask 9: right line 10

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x219C	8604	Byte	R/O	Ethernet IP address 1st byte
0x219D	8605	Byte	R/O	Ethernet IP address 2nd byte
0x219E	8606	Byte	R/O	Ethernet IP address 3rd byte
0x219F	8607	Byte	R/O	Ethernet IP address 4th byte

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.

Process values of the groups

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x21AC	8620	Boolean	R/O	Group alarm for group 1
0x21AD	8621	Boolean	R/O	Group alarm for group 2
0x...	...	Boolean	R/O	Group alarm for group...
0x21B4	8628	Boolean	R/O	Group alarm for group 9
0x21B5	8629	Boolean	R/O	Alarm: violation of positive tolerance, group 1, analog 1
0x21B6	8630	Boolean	R/O	Alarm: violation of positive tolerance, group 1, analog 2
0x...	...	Boolean	R/O	Alarm: violation of positive tolerance, group 1, analog...
0x21BA	8634	Boolean	R/O	Alarm: violation of positive tolerance, group 1, analog 6
0x21BB	8635	Boolean	R/O	Alarm: violation of positive tolerance, group 2, analog 1
0x21BC	8636	Boolean	R/O	Alarm: violation of positive tolerance, group 2, analog 2
0x...	...	Boolean	R/O	Alarm: violation of positive tolerance, group 2, analog...
0x21C0	8640	Boolean	R/O	Alarm: violation of positive tolerance, group 2, analog 6
0x21C1	8641	Boolean	R/O	Alarm: violation of positive tolerance, group 3, analog 1
0x21C2	8642	Boolean	R/O	Alarm: violation of positive tolerance, group 3, analog 2
0x...	...	Boolean	R/O	Alarm: violation of positive tolerance, group 3, analog...
0x21C6	8646	Boolean	R/O	Alarm: violation of positive tolerance, group 3, analog 6
0x21C7	8647	Boolean	R/O	Alarm: violation of positive tolerance, group 4, analog 1
0x21C8	8648	Boolean	R/O	Alarm: violation of positive tolerance, group 4, analog 2
0x...	...	Boolean	R/O	Alarm: violation of positive tolerance, group 4, analog...

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x21CC	8652	Boolean	R/O	Alarm: violation of positive tolerance, group 4, analog 6
0x21CD	8653	Boolean	R/O	Alarm: violation of positive tolerance, group 5, analog 1
0x21CE	8654	Boolean	R/O	Alarm: violation of positive tolerance, group 5, analog 2
0x...	...	Boolean	R/O	Alarm: violation of positive tolerance, group 5, analog...
0x21D2	8658	Boolean	R/O	Alarm: violation of positive tolerance, group 5, analog 6
0x21D3	8659	Boolean	R/O	Alarm: violation of positive tolerance, group 6, analog 1
0x21D4	8660	Boolean	R/O	Alarm: violation of positive tolerance, group 6, analog 2
0x...	...	Boolean	R/O	Alarm: violation of positive tolerance, group 6, analog...
0x21D8	8664	Boolean	R/O	Alarm: violation of positive tolerance, group 6, analog 6
0x21D9	8665	Boolean	R/O	Alarm: violation of positive tolerance, group 7, analog 1
0x21DA	8666	Boolean	R/O	Alarm: violation of positive tolerance, group 7, analog 2
0x...	...	Boolean	R/O	Alarm: violation of positive tolerance, group 7, analog...
0x21DE	8670	Boolean	R/O	Alarm: violation of positive tolerance, group 7, analog 6
0x21DF	8671	Boolean	R/O	Alarm: violation of positive tolerance, group 8, analog 1
0x21E0	8672	Boolean	R/O	Alarm: violation of positive tolerance, group 8, analog 2
0x...	...	Boolean	R/O	Alarm: violation of positive tolerance, group 8, analog...
0x21E4	8676	Boolean	R/O	Alarm: violation of positive tolerance, group 8, analog 6
0x21E5	8677	Boolean	R/O	Alarm: violation of positive tolerance, group 9, analog 1
0x21E6	8678	Boolean	R/O	Alarm: violation of positive tolerance, group 9, analog 2
0x...	...	Boolean	R/O	Alarm: violation of positive tolerance, group 9, analog...
0x21EA	8682	Boolean	R/O	Alarm: violation of positive tolerance, group 9, analog 6
0x21EB	8683	Boolean	R/O	Collective alarm: violation of positive tolerance, group 1
0x21EC	8684	Boolean	R/O	Collective alarm: violation of positive tolerance, group 2
0x...	...	Boolean	R/O	Collective alarm: violation of positive tolerance, group...
0x21F3	8691	Boolean	R/O	Collective alarm: violation of positive tolerance, group 9
0x21F4	8692	Boolean	R/O	Alarm: violation of negative tolerance, group 1, analog 1
0x21F5	8693	Boolean	R/O	Alarm: violation of negative tolerance, group 1, analog 2
0x...	...	Boolean	R/O	Alarm: violation of negative tolerance, group 1, analog...
0x21F9	8697	Boolean	R/O	Alarm: violation of negative tolerance, group 1, analog 6

7 Modbus address tables

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x21FA	8698	Boolean	R/O	Alarm: violation of negative tolerance, group 2, analog 1
0x21FB	8699	Boolean	R/O	Alarm: violation of negative tolerance, group 2, analog 2
0x...	...	Boolean	R/O	Alarm: violation of negative tolerance, group 2, analog...
0x21FF	8703	Boolean	R/O	Alarm: violation of negative tolerance, group 2, analog 6
0x2200	8704	Boolean	R/O	Alarm: violation of negative tolerance, group 3, analog 1
0x2201	8705	Boolean	R/O	Alarm: violation of negative tolerance, group 3, analog 2
0x...	...	Boolean	R/O	Alarm: violation of negative tolerance, group 3, analog...
0x2205	8709	Boolean	R/O	Alarm: violation of negative tolerance, group 3, analog 6
0x2206	8710	Boolean	R/O	Alarm: violation of negative tolerance, group 4, analog 1
0x2207	8711	Boolean	R/O	Alarm: violation of negative tolerance, group 4, analog 2
0x...	...	Boolean	R/O	Alarm: violation of negative tolerance, group 4, analog...
0x220B	8715	Boolean	R/O	Alarm: violation of negative tolerance, group 4, analog 6
0x220C	8716	Boolean	R/O	Alarm: violation of negative tolerance, group 5, analog 1
0x220D	8717	Boolean	R/O	Alarm: violation of negative tolerance, group 5, analog 2
0x...	...	Boolean	R/O	Alarm: violation of negative tolerance, group 5, analog...
0x2211	8721	Boolean	R/O	Alarm: violation of negative tolerance, group 5, analog 6
0x2212	8722	Boolean	R/O	Alarm: violation of negative tolerance, group 6, analog 1
0x2213	8723	Boolean	R/O	Alarm: violation of negative tolerance, group 6, analog 2
0x...	...	Boolean	R/O	Alarm: violation of negative tolerance, group 6, analog...
0x2217	8727	Boolean	R/O	Alarm: violation of negative tolerance, group 6, analog 6
0x2218	8728	Boolean	R/O	Alarm: violation of negative tolerance, group 7, analog 1
0x2219	8729	Boolean	R/O	Alarm: violation of negative tolerance, group 7, analog 2
0x...	...	Boolean	R/O	Alarm: violation of negative tolerance, group 7, analog...
0x221D	8733	Boolean	R/O	Alarm: violation of negative tolerance, group 7, analog 6
0x221E	8734	Boolean	R/O	Alarm: violation of negative tolerance, group 8, analog 1
0x221F	8735	Boolean	R/O	Alarm: violation of negative tolerance, group 8, analog 2
0x...	...	Boolean	R/O	Alarm: violation of negative tolerance, group 8, analog...
0x2223	8739	Boolean	R/O	Alarm: violation of negative tolerance, group 8, analog 6
0x2224	8740	Boolean	R/O	Alarm: violation of negative tolerance, group 9, analog 1
0x2225	8741	Boolean	R/O	Alarm: violation of negative tolerance, group 9, analog 2

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x...	...	Boolean	R/O	Alarm: violation of negative tolerance, group 9, analog...
0x2229	8745	Boolean	R/O	Alarm: violation of negative tolerance, group 9, analog 6
0x222A	8746	Boolean	R/O	Collective alarm: violation of negative tolerance, group 1
0x222B	8747	Boolean	R/O	Collective alarm: violation of negative tolerance, group 2
0x...	...	Boolean	R/O	Collective alarm: violation of negative tolerance, group...
0x2232	8754	Boolean	R/O	Collective alarm: violation of negative tolerance, group 9
0x2233	8755	Word	R/O	Operating mode, group 1
0x2234	8756	Word	R/O	Operating mode, group 2
0x...	...	Word	R/O	Operating mode, group...
0x223B	8763	Word	R/O	Operating mode, group 9

Ethercat process values

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x223C	8764	Float	R/O	Actual value 1
0x223E	8766	Float	R/O	Actual value 2
0x...	...	Float	R/O	Actual value...
0x22A6	8870	Float	R/O	Actual value 54
0x22A8	8872	Boolean	R/O	Analog alarm, min. 1
0x22A9	8873	Boolean	R/O	Analog alarm, min 2
0x...	...	Boolean	R/O	Analog alarm, min...
0x22DD	8925	Boolean	R/O	Analog alarm, min 54
0x22DE	8926	Boolean	R/O	Analog alarm max. 1
0x22DF	8927	Boolean	R/O	Analog alarm max. 2
0x...	...	Boolean	R/O	Analog alarm max....
0x2313	8979	Boolean	R/O	Analog alarm max. 54
0x2314	8980	Bit field 64	R/O	Bit 0 to 53 = 0x3FFFFFFFFFFFF: binary trace 1 to 54
0x2318	8984	Bit field 64	R/O	Bit 0 to 53 = 0x3FFFFFFFFFFFF: binary alarm 1 to 54

7 Modbus address tables

Process values for Modbus error

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x231C	8988	Boolean	R/O	Com1 error state
0x231D	8989	Word	R/O	Com1 last error code
0x231E	8990	Boolean	R/O	Com2 error state
0x231F	8991	Word	R/O	Com2 last error code
0x2320	8992	Boolean	R/O	Modbus/TCP slave 1 error state
0x2321	8993	Word	R/O	Modbus/TCP slave 1 last error code
0x2322	8994	Boolean	R/O	Modbus/TCP slave 2 error state
0x2323	8995	Word	R/O	Modbus/TCP slave 2 last error code
0x2324	8996	Boolean	R/O	Modbus/TCP master 1 error state
0x2325	8997	Word	R/O	Modbus/TCP master 1 last error code
0x2326	8998	Boolean	R/O	Modbus/TCP master 2 error state
0x2327	8999	Word	R/O	Modbus/TCP master 2 last error code
0x2328	9000	Boolean	R/O	Modbus/TCP master 3 error state
0x2329	9001	Word	R/O	Modbus/TCP master 3 last error code
0x232A	9002	Boolean	R/O	Modbus/TCP master 4 error state
0x232B	9003	Word	R/O	Modbus/TCP master 4 last error code

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

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
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

SystemIO network variables (NV), recipe of current batch 0 to 8

Address		Data type/ bit number	Access	Signal designation
Hex.	Dec.			
0x9000	36864	Char[1204]	R/W	Recipe of current batch 0
0x9400	37888	Char[1204]	R/W	Recipe of current batch 1
0x9800	38912	Char[1204]	R/W	Recipe of current batch 2
0x9C00	39936	Char[1204]	R/W	Recipe of current batch 3
0xA000	40960	Char[1204]	R/W	Recipe of current batch 4
0xA400	41984	Char[1204]	R/W	Recipe of current batch 5
0xA800	43008	Char[1204]	R/W	Recipe of current batch 6
0xAC00	44032	Char[1204]	R/W	Recipe of current batch 7
0xB000	45056	Char[1204]	R/W	Recipe of current batch 8

7 Modbus address tables

JUMO GmbH & Co. KG
Street address:
Moritz-Juchheim-Straße 1
36039 Fulda, Germany
Delivery address:
Mackenrodtstraße 14
36039 Fulda, Germany
Postal address:
36035 Fulda, Germany
Phone: +49 661 6003-0
Fax: +49 661 6003-607
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