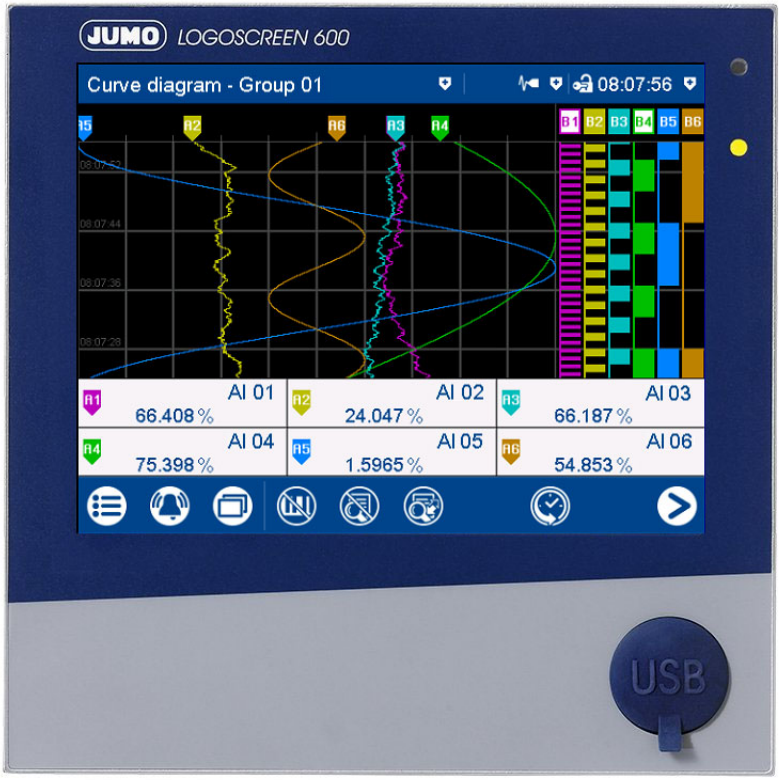
# JUMO LOGOSCREEN 600

## Paperless Recorder with Touchscreen




## Interface Description

70652000T92Z001K000

V2.00/EN/00625191

# Contents

# Contents

## 1.1 Warning symbols

**DANGER!**

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

**WARNING!**

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

**CAUTION!**

This symbol in connection with the signal word indicates that **material damage 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.

## 1.2 Note symbols

**NOTE!**

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

**REFERENCE!**

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

**FURTHER INFORMATION!**

This symbol is used in tables and indicates that **further information** is provided after the table.

**DISPOSAL!**

At the end of its service life, the device and any batteries present do not belong in the trash! Please ensure that they are **disposed of** properly and in an **environmentally friendly** manner.

# 1 Safety information

## 2.1 Location of interfaces

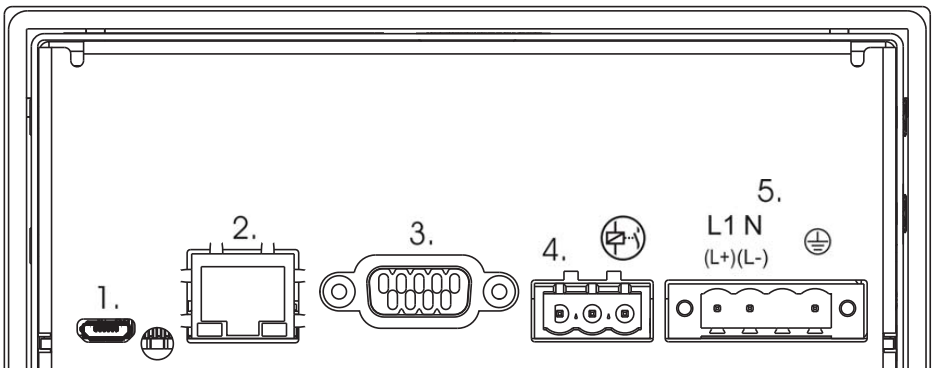
The paperless recorder features a serial interface and an Ethernet interface as standard.

The serial interface may be configured as an RS232 or RS485 interface and supports the Modbus protocol (Modbus RTU) as a master or slave. Alternatively, a barcode scanner may be connected.

The device may be integrated into a company network via the Ethernet interface. The following functions are provided for this purpose:

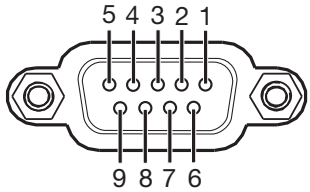
- Communication with PC software, such as a setup program, PCC, PCA3000 (HTTP)
- Web server for using a web browser (HTTP)
- Communication with SNTP server and email server (SMTP)
- Communication with a Modbus master or slave via Modbus/TCP

**Connection sockets (back)**




2. Ethernet interface
3. serial interface

## 2.2 Serial interface

| Version | Connection element.pin – assignment | Connection element |
|---|---|---|
| RS232<br>9-pin SUB-D socket<br>(switchable to RS485) | 3.2 – RxD (received data)<br>3.3 – TxD (transmission data)<br>3.5 – GND (ground) |  |
| RS485<br>9-pin SUB-D socket<br>(switchable to RS232) | 3.3 – TxD+/RxD+ (transmission/received data +)<br>3.5 – GND (ground)<br>3.8 – TxD-/RxD- (transmission/received data -) | |



**NOTE!**

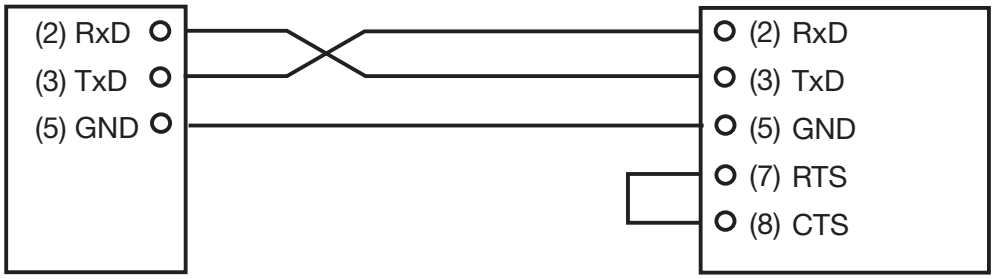
A connecting cable with shielding must be used to connect the RS232 interface.
A twisted connecting cable with shielding must be used to connect the RS485 interface.
To avoid transmission errors, only the signals listed above may be connected.

**NOTE!**

To ensure fault-free operation, terminating resistors are required at the beginning and end of an RS485 transmission path.

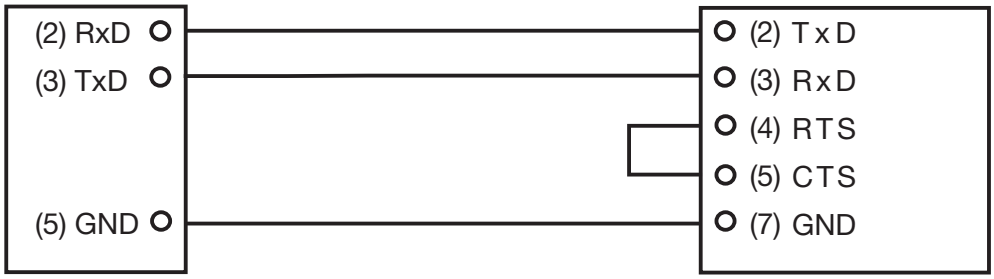
**Handshake lines for RS232**

No handshake signals (RTS, CTS) are used when using as an RS232 interface. The two pins are not assigned at the Sub D socket of the slave. If the signals are evaluated by the master (for example, through a PC program) they must be bridged in the connector.




Slave

Master with 9-pin Sub D socket




Slave

Master with 25-pin Sub D socket

**Configuration**

**NOTE!**

Refer to the operating manual of the paperless recorder for the configuration parameters and their settings (Configuration > Serial interface).

## 2.3 Ethernet interface

**Connection**

**NOTE!**

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

**CAUTION!**

**The device is not suitable for connection to a PoE port (power over Ethernet).**

There is the risk of damage to the device.

▶ Connect the device to an Ethernet port without PoE.

**Configuration**

**NOTE!**

Refer to the operating manual of the paperless recorder for the configuration parameters and their settings (Configuration > Ethernet).

## 2.4 Settings for Modbus/TCP

The settings for the Modbus/TCP operating mode are implemented in a separate configuration menu.

If the paperless recorder functions as a Modbus slave, two external devices (masters 1 and 2) can simultaneously access the paperless recorder. If it functions as a Modbus master, it can communicate with up to four external devices (Modbus slaves: device 1 to 4).

**Configuration**

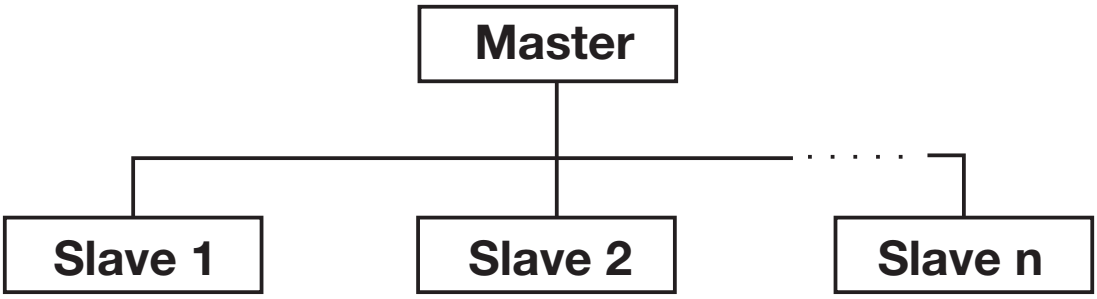
**NOTE!**

Refer to the operating manual of the paperless recorder for the configuration parameters and their settings (Configuration > Modbus/TCP).

# 2 Interfaces on the device

## 3.1 Master-slave principle

Communication between a master and a slave device with 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 paperless recorder can be operated either as a Modbus master or as a Modbus slave. The master and slave function can also be used in parallel.

This means external analog and digital signals, as well as texts, may be transmitted from a master to a paperless recorder (slave) as well as actively read in from one or more slaves by the paperless recorder (master). Conversely, the external device may access the process values of the paperless recorder as a master or slave.

If the paperless recorder functions as a master, the corresponding Modbus and device addresses must be assigned using the setup settings of the Modbus frames.

## 3.2 RTU transmission mode

In addition to Modbus/TCP, the RTU mode (Remote Terminal Unit) can also be used as the transmission mode. The transmission of a character is thus performed in binary format with 8 data bits, 1 start bit, 1 stop bit, and, if necessary, 1 parity bit. The highest value bit (MSB, most significant bit) is transmitted first.

The ASCII operating mode is not supported.

### Data format

The data format describes the structure of a transmitted character.

| 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 Chronological sequence of communication

### Character transmission time

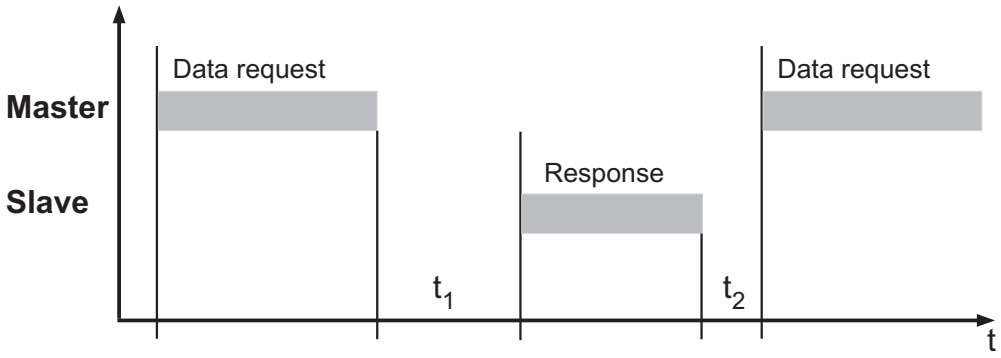
The character transmission time (time taken to transmit one character with 8 data bits) depends on the baud rate and the number of bits of the character (see table for data format):

**Character transmission time [ms] ='1000 × number of bits/baud rate**

| **Baud rate**[Bd] | **Number of bits** | **Character transmission time**[ms] |
|---|---|---|
| 115200 | 11 | 0.095 |
| | 10 | 0.087 |
| 38400 | 11 | 0.286 |
| | 10 | 0.26 |
| 19200 | 11 | 0.573 |
| | 10 | 0.521 |
| 9600 | 11 | 1.146 |
| | 10 | 1.042 |

### Time diagram of a data request

A data request runs 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.<br><br>min.: 5 ms<br>typical: 5 to 35ms<br>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.<br><br>For RS232: at least 3.5 times the character transmission time (end identifier)<br>For RS485: 35 ms |

**NOTE!**

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

**NOTE!**

In the configuration of the serial interface of the paperless recorder, a minimum response time can be set (0 to 500 ms). This set time is the minimum waiting period before an answer is transmitted. If a smaller value is set, then the response time may be longer than the set value (internal processing takes longer); the paperless recorder responds as soon as internal processing is completed. A set time of 0 ms

means that the paperless recorder responds at the maximum possible speed.
The adjustable minimum response time is required by the master for the RS485 interface in order to switch the interface drivers from transmitting to receiving. This parameter is not required for the RS232 interface.

**NOTE!**

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

## 3.4 Structure of a Modbus telegram

### Data structure

All telegrams have the same structure:

| Slave address | Function code | Data field | Checksum<br>CRC |
|---|---|---|---|
| 1 byte | 1 byte | x bytes | 2 bytes |

Each telegram has four fields:

**Slave address** Device address of a specific slave

**Function code** Function selection (read/write words)

**Data field** Contains information (according to the function code)

- Word address
- Number of words
- Word value(s)

**Checksum** Detection of transmission errors

# 3 Modbus protocol description

## 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.

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 addressed slave responds.

**Broadcast**

The broadcast is an instruction by the master to all slaves via the device address 0 (for example, to transmit a specific value to all slaves).
The connected slaves do not respond. In such a case, the correct acceptance of the value by the slaves should be checked by a subsequent readout at each individual slave. A data request with the device address 0 is meaningless.

**NOTE!**

A maximum of 31 slaves can be addressed via the RS485 interface.
The device address 0 is reserved as the Modbus broadcast address.
An instruction by the master to address 0 is carried out by all slaves, but no response is transmitted by them (as this would result in a data collision).
In the transmission protocol the address is specified in binary format.

## 3.6 Function codes

**Function overview**

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

| Function number | Function | Limit |
|---|---|---|
| 0x03 or 0x04 | Reading n words | Max. 127 words (254 bytes) |
| 0x06 | Writing one word | Max. 1 word (2 bytes) |
| 0x10 | Writing n words | Max. 127 words (254 bytes) |

**NOTE!**

A hexadecimal number is identified by "0x" preceding the actual number.
Example: 0x0010 (= $16_{DEC}$)

**NOTE!**

If the paperless recorder does not respond to these functions or outputs an error code, it can be evaluated.

⇨ chapter 3.9 "Error messages", page 19

### 3.6.1 Reading n words

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

**Data request**

| Slave address | Function<br>0x03 or 0x04 | Address<br>First word | Number of words | Checksum CRC |
|---|---|---|---|---|
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

**Response**

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

**Example**

Reading the IP address of the paperless recorder (slave, device address 1). Here, the example deals with the address 10.10.1.69. Since each IP address range is stored in a word, 4 words (8 bytes) must be read in.

Data request:

| 01 | 03 | 00 1B | 00 04 | 34 0E |
|---|---|---|---|---|
| Slave | Function | Address<br>1st word | Number of<br>words | CRC |

Response (values in Modbus floating-point format):

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

### 3.6.2 Writing one word

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

**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 value 1 to external binary input 1 to word address 0x1144.

Instruction:

| 01 | 06 | 11 44 | 00 01 | 06 23 |
|---|---|---|---|---|
| Slave | Function | Word address | Value | CRC |

Response:

| 01 | 06 | 11 44 | 00 01 | 06 23 |
|---|---|---|---|---|
| Slave | Function | Word address | Value | CRC |

### 3.6.3 Writing n words

**Instruction**

| Slave address | Function 0x10 | Address of first word | Number of words | Number of bytes | Word val-ue(s) | Checksum CRC |
|---|---|---|---|---|---|---|
| 1 byte | 1 byte | 2 bytes | 2 bytes | 1 byte | x bytes | 2 bytes |

**Response**

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

**Example**

Writing the word "Test" (ASCII encoding: 0x54 0x65 0x73 0x74 0x00) to external text variable 1 from word address 0x1218.

Instruction:

| 01 | 10 | 12 18 | 00 03 | 06 | 54 65 73 74 00 00 | 22 C5 |
|---|---|---|---|---|---|---|
| Slave | Function | Address 1st word | Number of words | Number of bytes | Text in ASCII | CRC |

Response:

| 01 | 10 | 12 18 | 00 03 | 05 77 |
|---|---|---|---|---|
| Slave | Function | Address 1st word | Number of words | CRC |

## 3.7 Transmission formats (integer, floating-point, and text values)

Functions 0x03 or 0x04 (reading n words) are used to read out integer, floating-point, double, and text values.

**Data request**

| Slave address | Function<br>0x03 or 0x04 | Address<br>First word | Number of words | Checksum CRC |
|---|---|---|---|---|
| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes |

Integer values are transmitted via Modbus in the following format:
The high byte first, followed by the low byte.

**Response**

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

### 3.7.1 Integer values

**Example**

In this hypothetical example, the integer value should be read out at address 0x11F1. The value here should be "4" (word value 0x0004).

Data request:

| 01 | 03 | 11 F1 | 00 01 | D0 C5 |
|---|---|---|---|---|
| Slave | Function | Address<br>1st word | Number of<br>words | CRC |

Response:

| 01 | 03 | 02 | 00 04 | B9 87 |
|---|---|---|---|---|
| Slave | Function | Bytes<br>read | Integer value | CRC |

### 3.7.2 Floating-point values

For floating-point values, Modbus functions with the IEEE 754 standard format (32-bit), but with the difference that bytes 1 and 2 are interchanged with bytes 3 and 4.

**Single floating-point format (32-bit) acc. to standard IEEE 754**

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

S - Prefix sign bit
E - Exponent (two's complement)
M - 23-bit normalized mantissa

**Modbus floating-point 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 input 1 should be read out at address 0x1006. The value here should be 550.0 (0x44098000 in the IEEE 754 format).

Data request:

| | | | | |
|---|---|---|---|---|
| 01 | 03 | 10 06 | 00 02 | 20 CA |
| Slave | Function | Address 1st word | Number of words | CRC |

Response (values in Modbus floating-point format):

| | | | | | |
|---|---|---|---|---|---|
| 01 | 03 | 04 | 80 00 | 44 09 | 20 F5 |
| Slave | Function | Bytes read | Floating-point value | | CRC |

After being transmitted from the paperless recorder, the bytes for the floating-point value must be interchanged accordingly.

Many compilers (for example, Microsoft® Visual C++) store the floating-point values in the following sequence:

**Floating-point value**

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

**NOTE!**

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

### 3.7.3 Text values (character strings)

Character strings are transmitted in ASCII format.

**NOTE!**

A "\0" (ASCII code 0x00) must always be transmitted as the terminating code. Characters after this mark have no significance.
Knowing that the transmission of texts takes place word by word (16-bit), 0x0 is additionally appended where an odd number of characters is used (incl. "\00").
The maximum lengths specified in the address tables for character strings also include the terminating "/0". This means the text can consist of max. 63 readable characters for "char 64".

**Example**

Text query from address 0x1218 that may be 64 characters long (incl. terminating code). In the example, the character string "Test" (ASCII code: 0x54, 0x65, 0x73, 0x74, 0x00) is under this address. However, in the request the maximum value is assumed for the number of words to be read: 64 characters per 1 byte = 64 bytes = $32_{DEC}$ words = 0x20 words.

Request: 010312180020 (+ 2 bytes CRC16)

Response: 010340**5465737**400 ... 00 (+ 2 bytes CRC16)

All 64 characters are read here ($64_{DEC}$ bytes = 0x40 bytes): 4 text characters (0x54, 0x65, 0x73, 0x74) + 1 terminating code character (0x00) + 59 further characters. The characters following the terminating code are not evaluated.

## 3.8 Checksum (CRC16)

**Calculation principle**

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

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

**NOTE!**

The low byte of the checksum is the first to be transmitted!
Example: The CRC16 checksum 0x3291 is transmitted and presented in the sequence 0x9132.

**Example**

Query status of digital input 1 (address 0x10D5):

Data request (CRC16 = 0x3291)

| | | | | |
|---|---|---|---|---|
| 01 | 03 | 10 D5 | 00 01 | 91 32 |
| Slave | Function | Address | Read one word | CRC |

Response (CRC16 = 0x8479)

| | | | | |
|---|---|---|---|---|
| 01 | 03 | 02 | 00 01 | 79 84 |
| Slave | Function | Number of bytes | Value | CRC |

## 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 defined incompletely or excessively
- The number of words to read is zero

In these cases, the data request should be sent again after a timeout time of approx. 1 s has elapsed.

**Error codes**

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

- 01 = Invalid function
- 02 = Invalid address or too many words should be read or written
- 03 = Value is outside the admissible range
- 08 = Value is write-protected

**Response in case of error**

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

The function code is ORed with 0x80. As a result, the highest value bit (MSB) is set to 1.

**Example**

Data request:

| 01 | 06 | 1099 | 0001 | 9CE5 |
|---|---|---|---|---|
| Slave | Write word | Word address | Word value | CRC |

Response:

| 01 | 86 | 08 | 43A6 |
|---|---|---|---|
| Slave | OR function | Error | CRC |

Response with error code 08, because address 0x1099 is write-protected.

## 3.9.2 Error messages for invalid values

For measured values in the floating-point format, the error is displayed in the value itself, i.e. it contains the error code instead of the measured value.

| **Error code for floating-point values** | **Error** |
|---|---|
| $1.0 \times 10^{37}$ | Underrange |
| $2.0 \times 10^{37}$ | Overrange |
| $3.0 \times 10^{37}$ | Value invalid/integrator or statistics destroyed |
| $4.0 \times 10^{37}$ | Division by zero |
| $5.0 \times 10^{37}$ | Math error |
| $6.0 \times 10^{37}$ | Error: terminal temperature/compensation signal |
| $7.0 \times 10^{37}$ | Probe short circuit |
| $8.0 \times 10^{37}$ | Probe break |

**Example**

Reading in the analog input 1 at address 0x1006:

Data request:

| 01 | 03 | 10 06 | 00 02 | 20 CA |
|---|---|---|---|---|
| Slave | Function | Word address | Number of words | CRC |

Response:

| 01 | 03 | 04 | 8E 52 | 7D B4 | 51 ED |
|---|---|---|---|---|---|
| Slave | Function | Read bytes | Error code | | CRC |

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

### 3.9.3 Error codes as integer return values

**NOTE!**

The following table contains error codes which apply to the various devices. The device in question may only support part of the error codes specified here.

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

**Error codes**

| Error code | Description |
|---|---|
| **Error list: Program memory management** | |
| 1 | Program cannot be created |
| 2 | Program does not exist |
| 3 | Program cannot be deleted |
| 4 | Section cannot be deleted |
| 5 | Checksum cannot be stored |
| 6 | Checksum cannot be read |
| 7 | Program cannot be copied |
| 8 | Section cannot be copied |
| 9 | Program checksum error |
| 10 | Program pointer tab. checksum error |
| 11 | Program memory end |
| 12 | Section does not exist |
| 13 | Repeat jump marks |
| **Error list: General input and output** | |
| 14 | Please use the ENTER key to confirm |
| 15 | Invalid number of digits |
| 16 | The entry contains invalid characters |
| 17 | Value outside of limits |
| 18 | Section not programmed correctly |
| 19 | Password error |
| **Error list: 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 is locked |
| 27 | Programming is locked |
| 28 | Write error in the ser. EEPROM (calib) |
| 29 | Hardware error: MANUAL + AUTO locked |
| 30 | Editing during active program inadmissible |
| 31 | Copying during active program inadmissible |

| Error code | Description |
|---|---|
| 32 | MANUAL is inadmissible during AUTO lead time |
| 33 | Section change! Image update needed |
| 34 | No DB number, image update from PLC |
| 35 | No DB number for process values from PLC |
| 36 | Printer busy or not ready |
| 37 | Setpoint value 1 not programmed |
| 38 | Set up printer (config. / interface) |
| 39 | Possible only when device in MANUAL mode |
| 40 | Autotuning already running |
| 41 | Time axis elapsed or not programmed |
| 42 | Time axis cannot be copied |
| 43 | Time axis does not exist |
| 44 | Program change is locked |
| 45 | MANUAL operation is locked |
| 46 | Program start is locked |
| **Error list: Interface processing** | |
| 47 | Incorrect response length |
| 48 | Timeout error (no response) |
| 49 | Error reported in telegram protocol |
| 50 | Checksum error |
| 51 | Parity error |
| 52 | Framing error |
| 53 | Interface buffer full |
| 54 | Addressing error (for example, address does not exist) |
| 55 | Incorrect or unexpected command |
| **Error list: Event processing** | |
| 60 | Event could not be created |
| 61 | Event setting failed |
| 62 | Event clear failed |
| 63 | Event wait failed |
| 64 | Event close failed |
| 65 | Event open failed |
| 66 | Sync error between group and data manager |
| **Error list: Message processing** | |
| 70 | No queue memory available |
| 71 | Message queue cannot be opened |
| 72 | Message pool cannot be created |
| 73 | Memory cannot be requested from memory pool |
| 74 | Message cannot be sent |
| **Error list: Processing of MQX functions** | |
| 80 | Task creation failed |
| 81 | Hardware timer not created |
| **Error list: Flash processing** | |
| 90 | Flash memory write error |
| **Error list: Other errors** | |

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

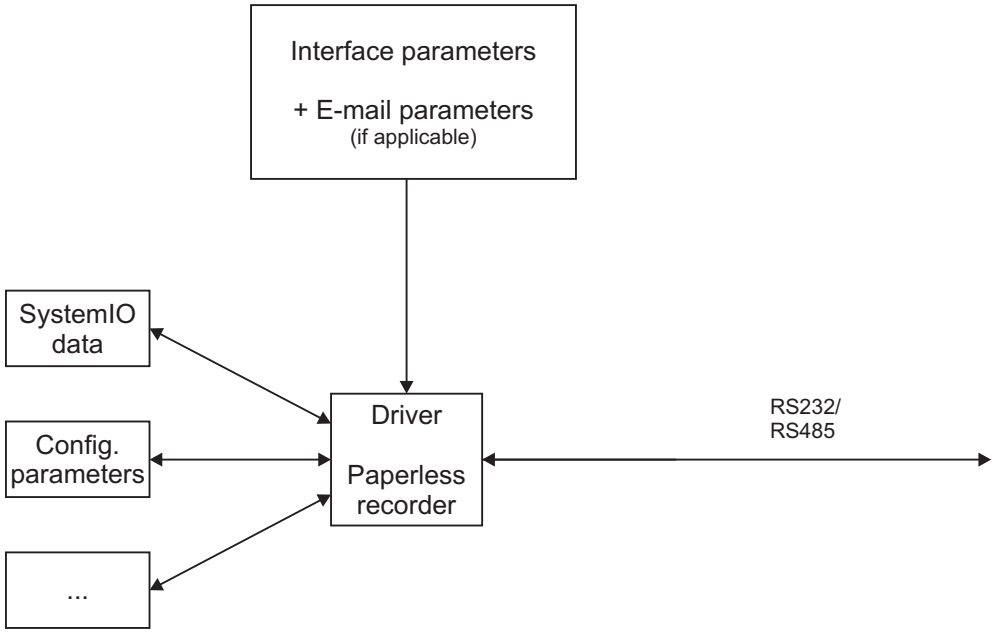
| Error code | Description |
|---|---|
| 152 | DNS response is split |
| 153 | No IP received via DNS |
| 154 | Unknown UDP port |
| 155 | TCP checksum incorrect |
| 156 | Incorrect TCP port |
| 157 | Unknown TCP-SYN option |
| 158 | TCP port not used |
| 159 | Unknown POP3 response |
| 160 | Unknown SMTP response |
| 161 | Unknown DNS name |
| 162 | No MD5 requested for CHAP |
| 163 | Authentication error |
| 164 | Canceled from opposite end |
| 165 | Error creating TCP socket |
| 166 | Error binding TCP socket |
| 167 | Error during TCP connect |
| 168 | Error sending TCP telegram |
| 169 | Error closing TCP socket |
| 170 | Error during TCP listen |
| 171 | Reset during TCP accept |
| 172 | Error during TCP accept |
| 173 | SMTP server reports syntax error |
| 174 | TCP socket is already closed |
| 175 | Incorrect frame configuration |
| **Error list: File system processing** | |
| 200 | Error installing partition manager |
| 201 | Error installing MFS file system |
| 202 | Error uninstalling partition manager |
| 203 | Error uninstalling MFS file system |

The serial interface is configurable as an RS232 or RS485 interface.

The interface may be used in Modbus slave operation, for example, to connect to a PLC, a PC with visualization software, or an SCADA system.

In Modbus master operation, external devices may be connected as a Modbus slave, for example, controllers, power controllers, or sensors. It is therefore possible to transmit external analog and digital signals, as well as texts, to a paperless recorder.

## 4.1 Modbus slave operation via serial interface




If the paperless recorder 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 their device address.
The master can query all device variables according to the Modbus address tables of this slave.

⇨ chapter 7 "Modbus address tables", page 43

Refer to the operating manual of the paperless recorder for the configuration parameters and their settings (Configuration > Serial interface > Modbus slave).

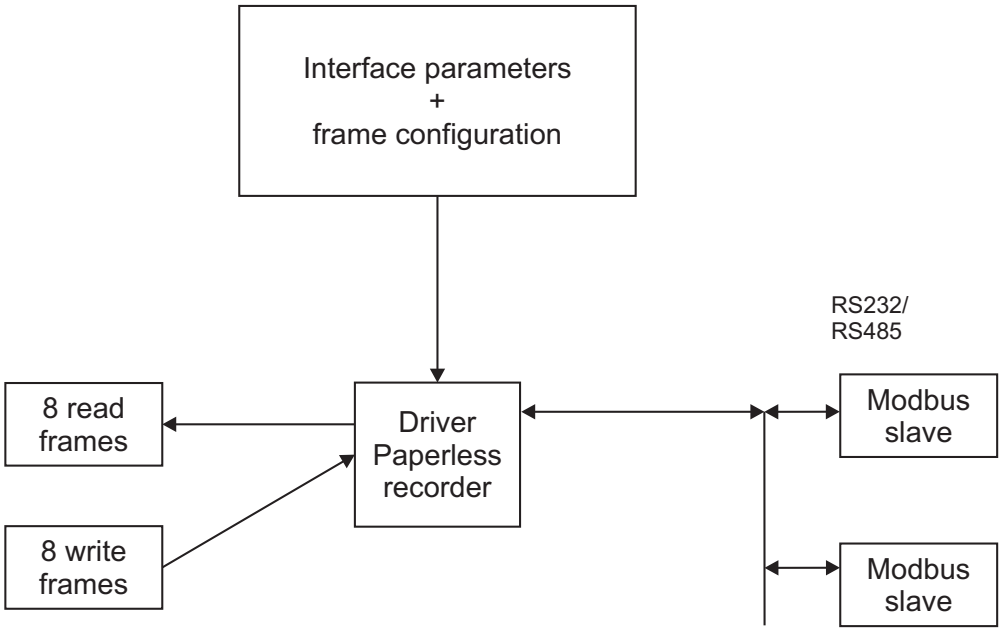
**Timeout monitoring**

This function monitors the communication between the Modbus master and Modbus slave from the perspective of the Modbus slave. After activated timeout monitoring, a timer starts to run with the first received request (timer time is the time set in the configuration under "Timeout"). The timer starts to run again after each new request. If the request does not appear, an internal digital signal is activated after the timer has elapsed and an entry is made in the alarm and event list. The signal is reset with the next request.

The signal (slave timeout Com1) is available in the digital selector (also as an inverted signal) and can also be queried via Modbus (addresses 0x113E and 0x113F). It is therefore possible to evaluate the connection status of the serial interface.

# 4 Serial transmission modes

## 4.2 Modbus master operation via serial interface




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

A maximum of 32 frames can be activated for reading and a maximum of 32 frames for writing. The activated frames are cyclically processed in sequence. This also applies to writing frames regardless of whether a process value has changed or not. This rules out a transmission that depends on the condition of a change. The complete frame is always transmitted with the frame length configured in the setup program.

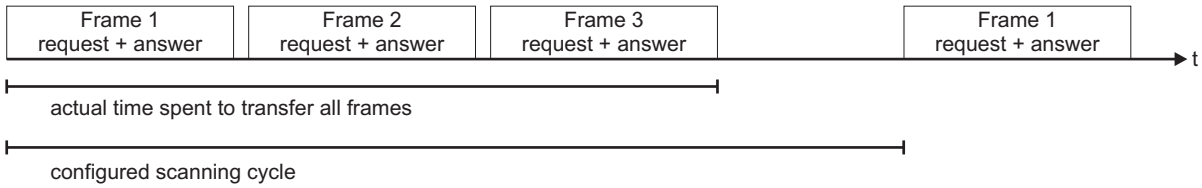
**NOTE!**

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

Refer to the operating manual of the paperless recorder for the configuration parameters and their settings (Configuration > Serial interface > Modbus master).

### Chronological sequence

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




If the configured scanning cycle is greater than the elapsed actual time required for scanning, the paperless recorder 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 break.

## 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 packed in a TCP frame (tunneled) and transmitted via Ethernet. The Modbus telegram (without CRC) is transmitted with an additional 6 or 7 byte "MBAP header" (Modbus Application Header). The seventh byte corresponds to the first serial byte, but has a different designation here.

**Structure of a Modbus/TCP telegram**

| **MBAP header** | | | | **Modbus telegram** |
|---|---|---|---|---|
| 2 bytes<br>Transaction ID | 2 bytes<br>Protocol ID | 2 bytes<br>Length | 1 byte<br>Unit ID | Additional bytes as below, however, without CRC |
| Identical in request and response | Must be 0 for Modbus | Length of request/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<br>Address<br>1 byte | Function code<br>1 byte | Data field<br>x bytes | CRC16<br>2 bytes |

With Modbus/TCP, a suitable process data visualization program can read and write device values via a company-wide Ethernet network, for example. All device variables from the Modbus address tables can be accessed.

⇨ chapter 7 "Modbus address tables", page 43

**Example: Reading n words**

Reading the IP address of the device. 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 in.

See also the Modbus example in chapter 3.6.1 "Reading n words", page 15

**Request:**

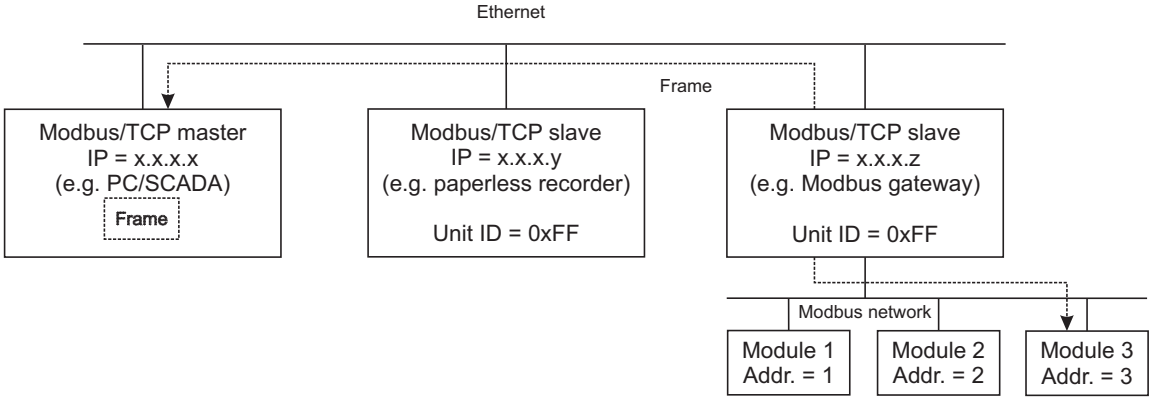
| **MBAP header** | | | | **Modbus telegram (without slave address and CRC)** | | |
|---|---|---|---|---|---|---|
| 00 01 | 00 00 | 00 06 | FF | 03 | 00 1B | 00 04 |
| 2 bytes Transaction ID | 2 bytes Protocol ID | 2 bytes Length | 1 byte Unit ID | 1 byte Function code | 2 bytes Address First word | 2 bytes Number of words |
| Assignment of response to request (consecutive numbering) | For Modbus always 0x00 | 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 IP address to be read | 4 words should 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 bytes Transaction ID | 2 bytes Protocol ID | 2 bytes Length | 1 byte Unit ID | 1 byte Function code | 1 byte Number of bytes read | 8 bytes Data read | | | |
| Assignment of response to request (consecutive numbering) | For Modbus always 0x00 | Length of the request 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 of the networking options when using the Modbus/TCP protocol:




Each Modbus/TCP node is accessible via a unique IP address. The "Unit ID" also contained in the protocol (= device address) must be 0xFF for the node directly attached to the network. However, a node can present itself virtually as several subordinate devices. For example, this may be the case with a Modbus gateway with the connected modules. These modules can then be accessed with device addresses 1 to 254.

## 5.3 Modbus master with Modbus/TCP

Up to 4 connections are possible to different Modbus/TCP devices which allow the exchange of a maximum of 32 reading and 32 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.

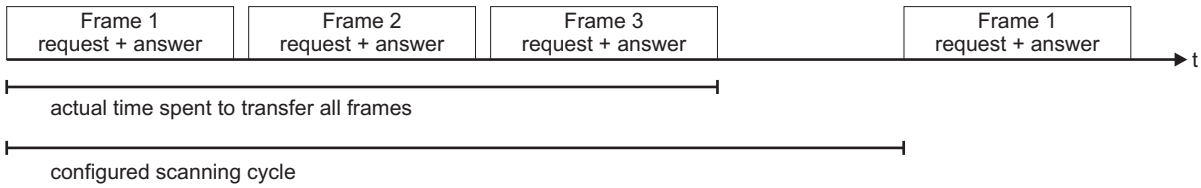
⇨ chapter 4 "Serial transmission modes", page 25

Frames with 4 opposing ends (devices 1 to 4) can be exchanged using 4 sockets. The Modbus device address that can be edited inside each frame mask and that is transmitted in the Modbus/TCP protocol can also be used to address modules behind Modbus gateways. The frame from Modbus gateway can thus be forwarded to the module.

Refer to the operating manual of the paperless recorder for the configuration parameters and their settings (Configuration > Modbus/TCP > Modbus master).

### Chronological sequence

The relevant interface (setup program: ***Only setup*** > ***Modbus frames for reading/writing*** > Parameter ***Interface***) searches for all frames configured for it and cyclically transmits them in sequence, to frame 1, frame 2, and frame 3, such as in the example below.




If the configured scanning cycle is greater than the elapsed actual time required for scanning, the paperless recorder 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 break.

## 5.4 Modbus slave with Modbus/TCP

Here, the paperless recorder functions as a slave and is available for requests on the bus from the Modbus master. A master can query all device variables according to the Modbus address tables for this slave.

⇨ chapter 7 "Modbus address tables", page 43

**NOTE!**

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

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

Refer to the operating manual of the paperless recorder for the configuration parameters and their settings (Configuration > Modbus/TCP > Modbus slave).

**Timeout monitoring**

This function monitors the communication between the Modbus master and Modbus slave from the perspective of the Modbus slave. After activated timeout monitoring, a timer starts to run with the first received request (timer time is the time set in the configuration under "Timeout"). The timer starts to run again after each new request. If the request does not appear, an internal digital signal is activated after the timer has elapsed and an entry is made in the alarm and event list. The signal is reset with the next request.

The signal of the relevant connection (Master 1: slave timeout TCP1; Master 2: slave timeout TCP2) is available in the digital selector (also as an inverted signal) and can also be queried via Modbus (addresses from 0x1140). It is therefore possible to evaluate the connection status of the Modbus/TCP interface.

## 5.5 HTTP

In this case, the paperless recorder 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 (PCA3000), or PCA Communication Software (PCC).

## 5.6 Web server

The paperless recorder can also be accessed by a browser using the HTTP protocol. The URL required for this purpose is the IP address of the paperless recorder. The HTML start page "index.htm" is accessed and can be used to branch to further HTML pages.

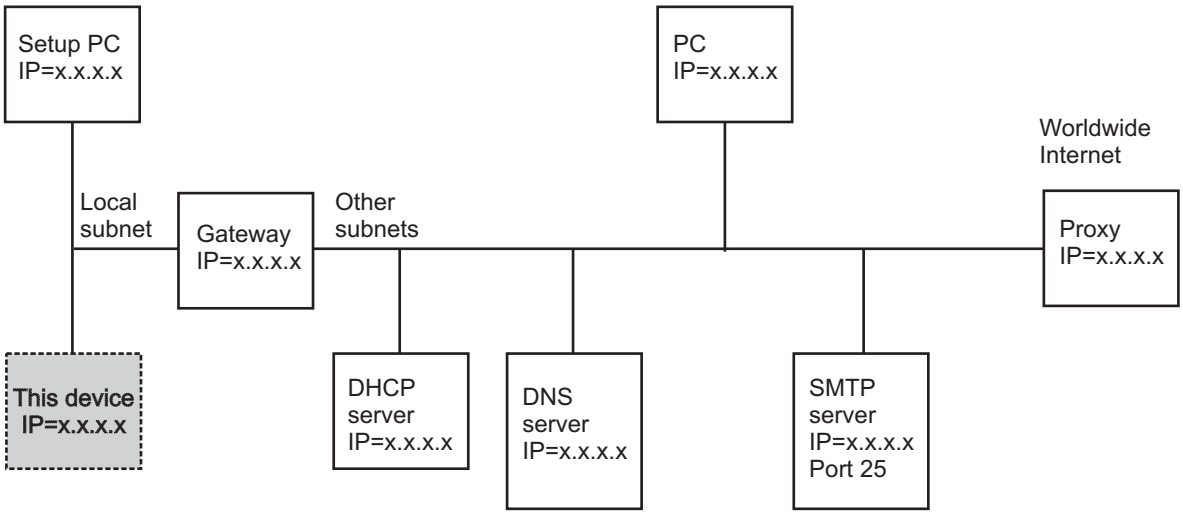
The start page "index.htm" and other HTML pages can be loaded into the paperless recorder using the setup program. The Microsoft® plugin "Silverlight®" is required for the browser in order to use the index.htm supplied per default.

Refer to the operating manual of the paperless recorder for the configuration parameters and their settings (Configuration – only in the setup program > Web server).

## 5.7 Email (SMTP)

The paperless recorder can send emails (for example, alarms). In this case, it is the master (client) and can access SMTP servers at the standard port (25).

**Typical networking in the company network**




**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; a DHCP server is then no longer needed.

**DNS server:**

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

**SMTP server:**

Used to send emails.

**Proxy:**

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

# 5 Ethernet transmission modes

### Parameters for email and mail server

Refer to the operating manual of the paperless recorder for the configuration parameters and their settings (Configuration – only in the setup program > Email).

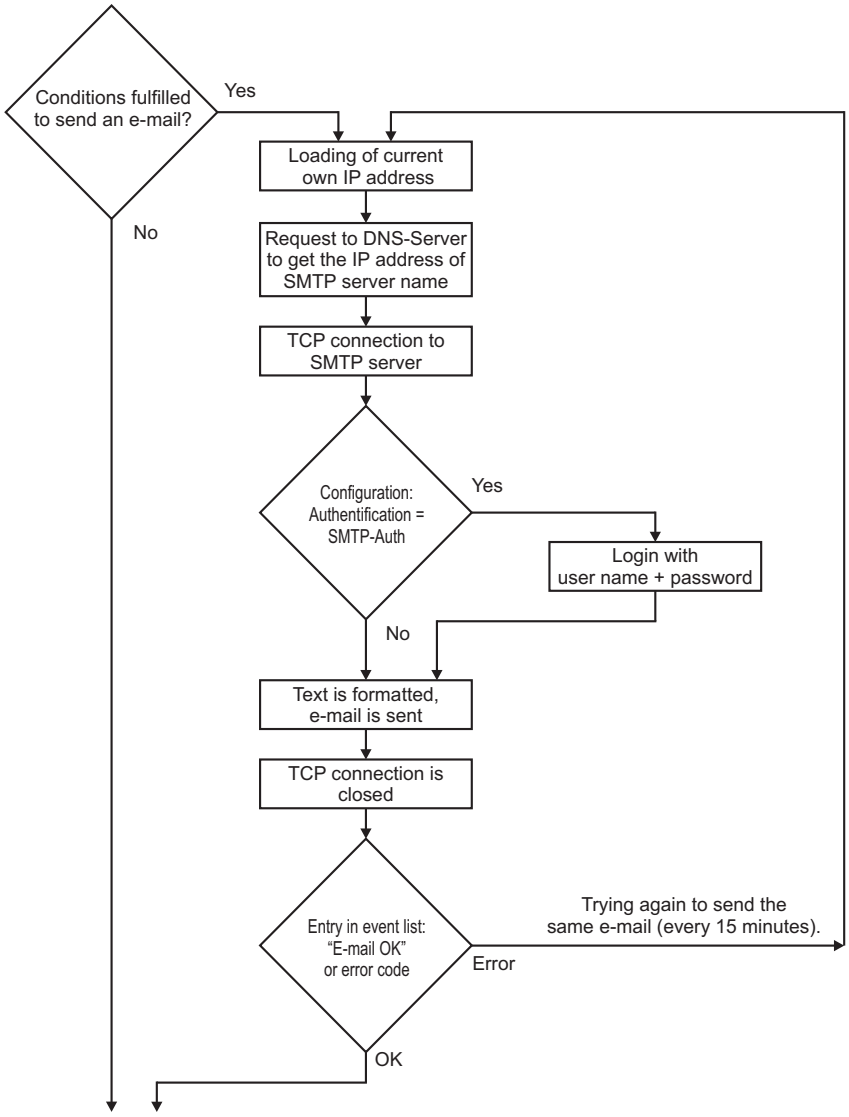
**NOTE!**

The mail server must be on the company network (not accessible on the Internet or without proxy addressing). This mail server should also be able to transmit emails via the Internet.

### Sending an e-mail via the Internet

Here, several steps depend on configured device parameters. An error code of the event entry 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 21



## 6.1 General information

The paperless recorder allows the user to individually compile Modbus frames for their application. As a result, the highest level of flexibility is reached and the data exchange on the bus is reduced.

This offers a considerable advantage with regard to the transfer 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.

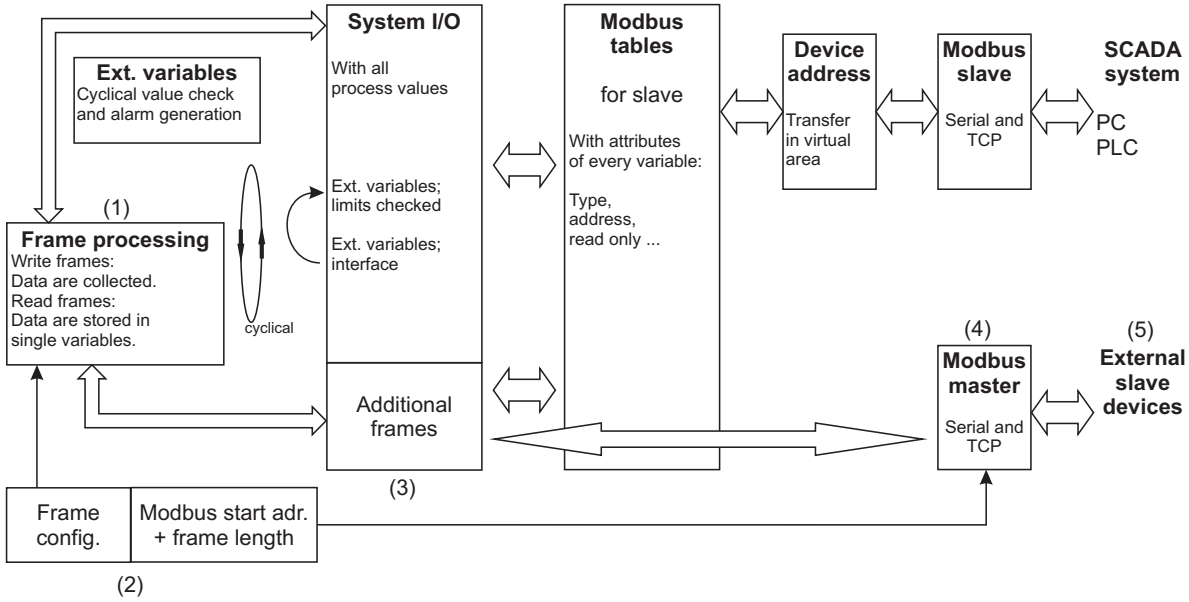
The maximum size of a frame is set to 254 bytes. This corresponds to 127 words for each command. The activated reading or writing frames are cyclically processed in sequence. The frame length specified in the setup program is always used. This length is automatically calculated using the number of entries (24 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 opposite end. |
| (5) | The opposite end receives these frames and responds. |

# 6 Modbus frames

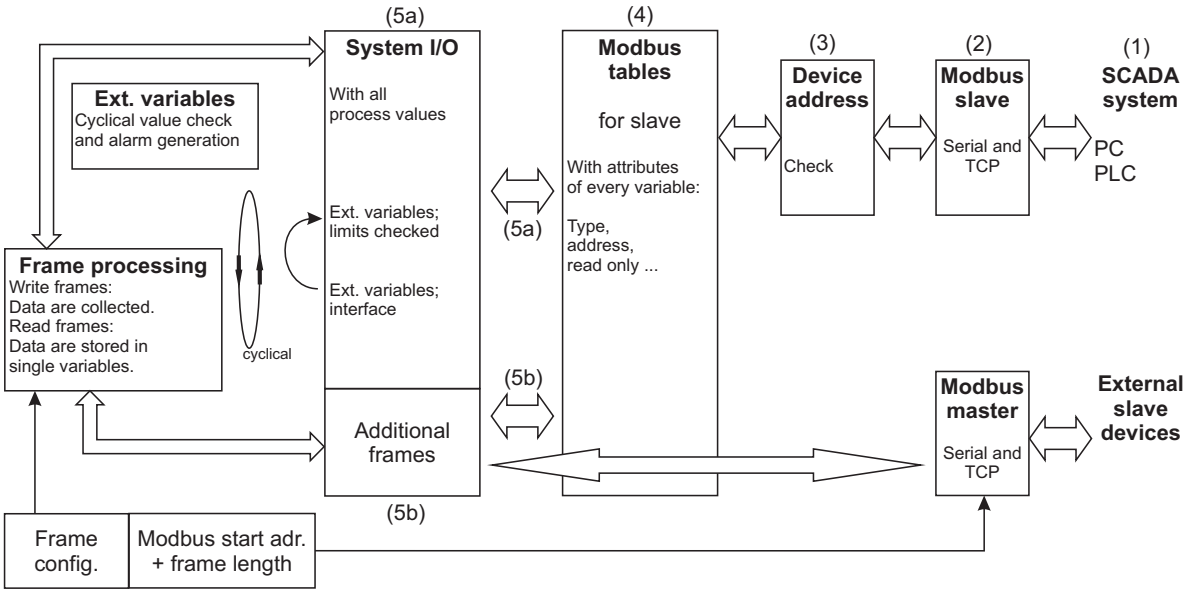
**NOTE!**

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", page 21.

### Error monitoring

Each frame is monitored for data transmission. If an error occurs, the corresponding error flag is set, an error code is also stored for each frame, and an entry is written in the event list. This entry is not cyclically generated and set; instead, it is generated and set once when the error occurs and then again when the transmission functions again. The following applies for all read frames:
If there is no valid response after 3 recurrences, all values for this frame are set to "No input value".

### Block diagram of the inputs and outputs for the slave function




### Reading/writing Modbus (slave)

| | |
|---|---|
| (1) | An external master issues a 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

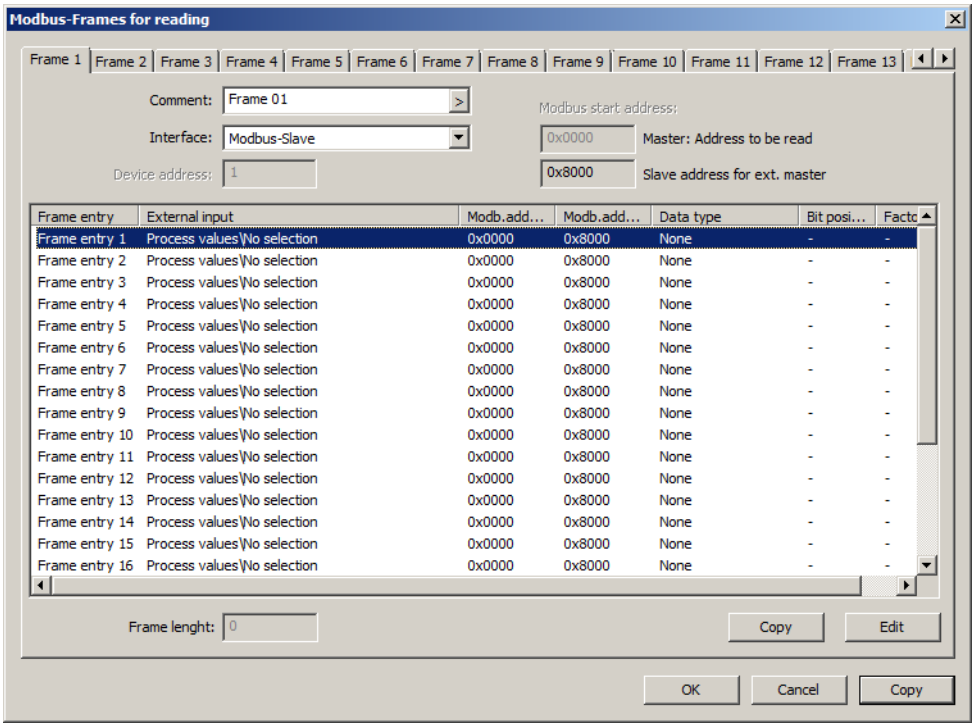
The functions "Modbus frames for reading" and "Modbus frames for writing" are in the setup program in the navigation tree under ***Only setup***.

### 6.3.1 Modbus frames for reading

This function is used to compile up to 32 Modbus frames for reading process values from external devices (via interface) individually for each opposite end. The process values (analog and digital values; texts) are written to the selected variables (external inputs and texts) from the received Modbus telegram and are available for use in the paperless recorder.

Each frame can be used to configure up to 24 frame entries with a process value each, which are then grouped and transmitted in a Modbus telegram.

**Setup dialog**




**CAUTION!**

**A variable can be used in multiple frames.**

This means that different process values are written to the same variable.

▶ You must ensure that no variables are overwritten unintentionally.

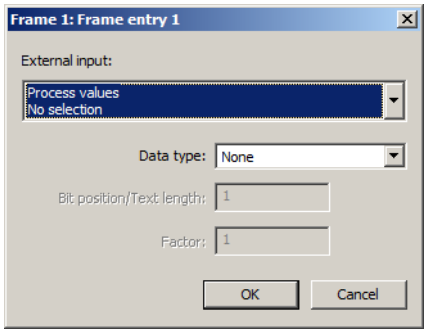
# 6 Modbus frames

**Parameters**

| Parameters | Selection/settings | Description |
|---|---|---|
| Comment | **Frame 01**<br>Use default text or edit text. | Comment on the more detailed description of the frame |
| Interface | The selection determines whether the frame is actively transmitted as a master or only available for queries as a slave. In the case of a Modbus master, the interface on which the relevant frame is used is also specified. If it is an Ethernet interface, the external device to be addressed must also be selected. | |
| | Modbus slave | Compiled frame only available for queries as a slave |
| | Modbus master TCP 1 | Modbus master; Modbus/TCP via Ethernet, device 1 addressed |
| | Modbus master TCP 2 | Modbus master; Modbus/TCP via Ethernet, device 2 addressed |
| | Modbus master TCP 3 | Modbus master; Modbus/TCP via Ethernet, device 3 addressed |
| | Modbus master TCP 4 | Modbus master; Modbus/TCP via Ethernet, device 4 addressed |
| | Modbus master serial | Modbus master; Modbus RTU via RS232/485 |
| Device address | **1** to 255 | Device address of the external device (Modbus slave)<br>Address 255 must be set for Modbus/TCP. |
| Modbus start address, master<br>(paperless recorder is the Modbus master) | **0x0000** to 0xFFFF | Modbus start address (offset) of the external slave device<br>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<br>(paperless recorder is the Modbus slave) | **0x8000** | Display of the Modbus start address (offset) of the paperless recorder<br>The start address is used for the setting in the external Modbus master. |
| Entry 1 to Entry 24 | Select the desired entry (double-click the line with the entry or mark and then click the "Edit" button). | |
| Frame length | **0** to 254 | Displays the frame length (byte(s)) |

### Editing

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




### Parameters

| Parameters | Selection/settings | Description |
|---|---|---|
| External input | External process values (analog inputs, digital inputs, texts) | |
| | **No selection** | No process value selected |
| | Select process value | Selector for selecting a process value |
| Data type | The data type is preset depending on the type of external input. It can subsequently be changed. | |
| | None | No data type selected |
| | Floating point (LSB) | Floating-point number; lowest value bit (LSB) is transmitted first |
| | Floating point (MSB) | Floating-point number; highest value bit (MSB) is transmitted first<br>(This is the standard format for floating point 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 |
| | Unsigned int. (1 byte) | Integer without prefix sign, with length of 1 byte |
| | Unsigned int. (2 bytes) | Integer without prefix sign, with length of 2 bytes |
| | Unsigned int. (4 bytes) | Integer without prefix 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<br>Integer (2 bytes): **0** to 15<br>Integer (4 bytes): **0** to 31<br>Text (1 char./word): **1** to 127<br>Text (2 char./word): **1** to 254 | The setting ranges of the bit position or text length depend on the data type selected.<br>(This parameter is inactive with an external analog input.) |

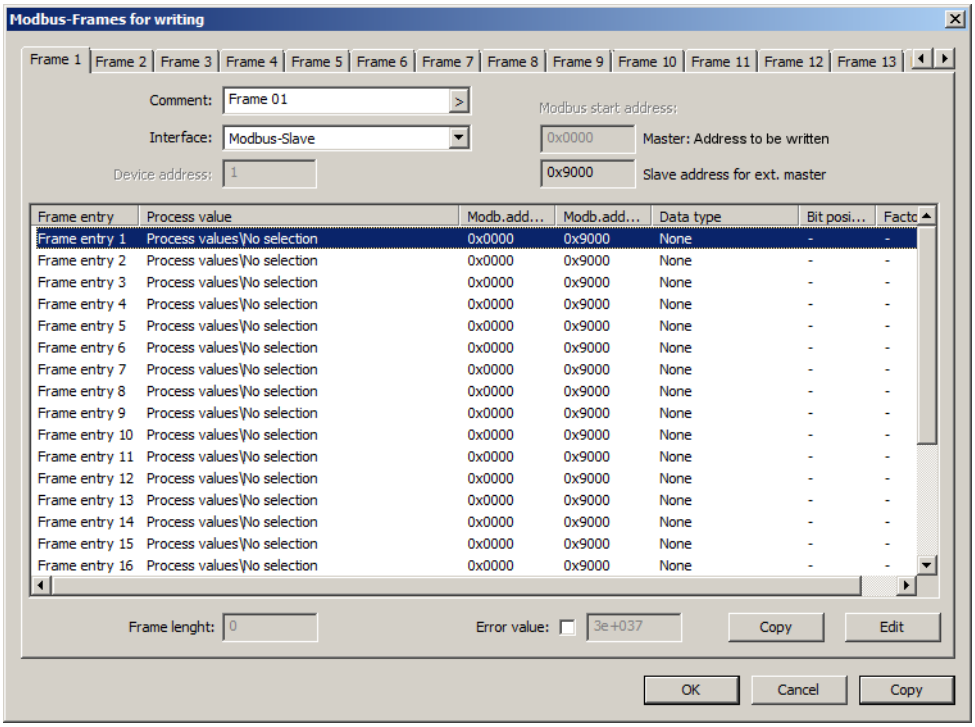
| Parameters | Selection/settings | Description |
|---|---|---|
| Factor | Using the factor makes it possible to transmit floating-point values in the integer format, for example. The transmitter must multiply the data with the corresponding factor before transmission. The data must be divided by the same value in the receiver. | |
| | Complete floating-point range allowed; default value = 1.0 | This factor is used to rescale values during the transmission, in particular for simultaneous type conversions. (This parameter is inactive with an external digital input or text.) |

## 6.3.2 Modbus frames for writing

This function can be used to compile up to 32 Modbus frames for writing the paperless recorder's process values to external devices (via interface) individually for each opposite end. The paperless recorder writes the process values (analog and digital values; texts) to the frames, where they are available for external devices.

Each frame can be used to configure up to 24 frame entries with a process value each, which are then grouped and transmitted in a Modbus telegram.

### Setup dialog

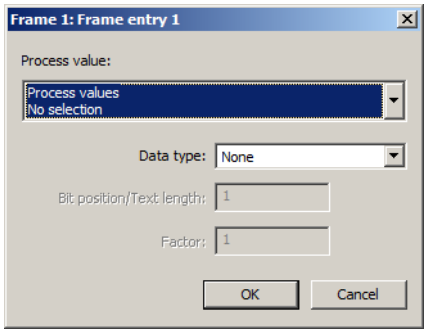


**Parameters**

| Parameters | Selection/settings | Description |
|---|---|---|
| Comment | **Frame 1**<br>Use default text or edit text. | Comment on the more detailed description of the frame |
| Interface | The selection determines whether the frame is actively transmitted as a master or only available for queries as a slave. In the case of a Modbus master, the interface on which the relevant frame is used is also specified. If it is an Ethernet interface, the external device to be addressed must also be selected. | |
| | Modbus slave | Compiled frame only available for queries as a slave |
| | Modbus master TCP 1 | Modbus master; Modbus/TCP via Ethernet, device 1 addressed |
| | Modbus master TCP 2 | Modbus master; Modbus/TCP via Ethernet, device 2 addressed |
| | Modbus master TCP 3 | Modbus master; Modbus/TCP via Ethernet, device 3 addressed |
| | Modbus master TCP 4 | Modbus master; Modbus/TCP via Ethernet, device 4 addressed |
| | Modbus master serial | Modbus master; Modbus RTU via RS232/485 |
| 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.<br>Address 255 must be set for Modbus/TCP. |
| Modbus start address, master<br>(paperless recorder is the Modbus master) | **0x0000** to 0xFFFF | Modbus start address (offset) of the external slave device<br>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<br>(paperless recorder is the Modbus slave) | **0x9000** | Display of the Modbus start address (offset) of the paperless recorder<br>The start address is used for the setting in the external Modbus master. |
| Entry 1 to entry 24 | Select the desired entry (double-click the line with the entry or mark and then click the "Edit" button). | |
| Frame length | **0** to 254 | Displays the frame length (byte(s)) |
| Error value | -3.37E+38 to **+3.0E+37** to +3.37E+38 | Value (code) that is transmitted instead of the measured value if an error occurs (for measured values in the floating-point format). |

### Editing

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




### Parameters

| Parameters | Selection/settings | Description |
|---|---|---|
| Process value | Process values of the paperless recorder (analog signals, digital signals, texts) | |
| | **No selection** | No process value selected |
| | Select process value | Selector for selecting a process value |
| Data type | The data type is preset depending on the type of process value. It can subsequently be changed. | |
| | None | No data type selected |
| | Floating point (LSB) | Floating-point number; lowest value bit (LSB) is transmitted first |
| | Floating point (MSB) | Floating-point number; most significant bit (MSB) is transmitted first. (This is the standard format for floating-point values with Modbus.) |
| | Integer (1 byte) | Integer with length of 1 byte |
| | Integer (2 bytes) | Integer with length of 2 bytes |
| | Integer (4 bytes) | Integer with length of 4 bytes |
| | Unsigned int. (1 byte) | Integer without prefix sign, with length of 1 byte |
| | Unsigned int. (2 bytes) | Integer without prefix sign, with length of 2 bytes |
| | Unsigned int. (4 bytes) | Integer without prefix 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<br>Integer (2 bytes): **0** to 15<br>Integer (4 bytes): **0** to 31<br>Text (1 char./word): **1** to 127<br>Text (2 char./word): **1** to 254 | The setting ranges of the bit position or text length depend on the data type selected.<br>(This parameter is inactive with an analog signal.) |

| Parameters | Selection/settings | Description |
|---|---|---|
| Factor | Using the factor makes it possible to transmit floating-point values in the integer format, for example. The transmitter must multiply the data with the corresponding factor before transmission. The data must be divided by the same value in the receiver. | |
| | Complete floating-point range allowed; default value = 1.0 | This factor is used to rescale values during the transmission, in particular for simultaneous type conversions. (This parameter is inactive with a digital signal or text.) |

## 6.4 Examples for the data transmission options with frames

There are flexible customization options between the content of the variables in the paperless recorder and the data format on the opposite end. The process value selected from the selector determines the data format within the paperless recorder; the selection under "Data type" determines the data format in the Modbus frame or on the opposite end. The two data formats do not have to match. As a result, type conversion can be carried out. The logical direction is displayed in the following diagram:

| | | |
|---|---|---|
| Data format of the process value | Inclusion of bit pos./text length and factor depending on the selection | "Data type" in the Modbus frame or on the opposite end |

**NOTE!**

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

The admissible setting options are:

| | | | | |
|---|---|---|---|---|
| Floating-point value | ⟷ | x factor | ⟷ | Floating-point value |
| Floating-point value | ⟷ | x factor | ⟷ | Integer value |
| Integer value | ⟷ | x factor | ⟷ | Floating-point value |
| Integer value | ⟷ | x factor | ⟷ | Integer value |
| Binary/Boolean value | ⟷ | Bit position | ⟷ | Integer value[a] |
| Binary/Boolean value | ⟷ | Bit position | ⟷ | "None"[b] |
| Text | ⟷ | Text length in bytes | ⟷ | Text (1 character/word)<br>Text (2 characters/word) |

[a] Only the configured bit (bit position within the integer value) is used.

[b] Other bits from the previous frame entry are used with the data type "Integer...". The bit position must be specified in every frame entry with the data type "None". This determines at which point the bit for the relevant frame entry is transmitted within the previous integer value.

# 6 Modbus frames

**NOTE!**

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

**NOTE!**

Device address 255 must be used if the paperless recorder is accessed via its IP address (Modbus slave).

## 7.1 Data types and access types

### Data types

| | |
|---|---|
| Bool | The least significant bit of a word (16-bit) as a Boolean value (1 = TRUE; 0 = FALSE); the remaining bits are not used. |
| Byte | Low byte of a word as an integer value (value range 0 to 255); the high byte is not used. |
| Byte[n] | Data field with n bytes (n/2 words); max. size of a reading or writing frame; the data type depends on the use in the frame. |
| Word | Word (16-bit) as an integer value (value range 0 to 65535) |
| Uint32 | Double word (32-bit) as an unsigned integer value (unsigned integer, value range: 0 to 4,294,967,295) |
| Uint16 | Word (16-bit) as an unsigned integer value (unsigned integer, value range: 0 to 65535) |
| Uint8[n] | Data field from n unsigned integer values (n/2 words; unsigned integer, each value range 0 to 255); is used for a MAC address, for example. |
| Floating point | Double word (32-bit) as floating-point value according to IEEE 754 |
| Char[n] | Text with n characters (2 characters in one word) |

### Access types

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

**CAUTION!**

**Write operations in some R/W parameters result in them being saved in 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 Addresses of the paperless recorder

The following tables contain important device data, process values, and configuration parameters for the paperless recorder including their address, data type, and type of access.

**Device data: versions, addresses, date/time**

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x0001 | 1 | Uint32 | R/O | Software version **basic device** (main circuit board) |
| 0x0003 | 3 | Uint16 | R/O | --- |
| 0x0004 | 4 | Uint16 | R/O | Internal SW version |
| 0x0005 | 5 | Uint32 | R/O | VdN number |
| 0x0007 | 7 | Uint32 | R/O | HW compatibility index |
| 0x0009 | 9 | Uint32 | R/O | HW version |
| 0x000B | 11 | Uint32 | R/O | HW VdN number |
| 0x000D | 13 | Char[22] | R/O | Fabrication number |
| 0x0018 | 24 | Uint8[6] | R/O | MAC address |
| | | | | |
| 0x001B | 27 | Byte | R/O | Ethernet: IP address, 1st byte |
| 0x001C | 28 | Byte | R/O | Ethernet: IP address, 2nd byte |
| 0x001D | 29 | Byte | R/O | Ethernet: IP address, 3rd byte |
| 0x001E | 30 | Byte | R/O | Ethernet: IP address, 4th byte |
| 0x001F | 31 | Char[190] | R/O | DNS device name |
| | | | | |
| 0x007E | 126 | Uint32 | R/O | Software version **analog/digital I/O 1** (Option 1) |
| 0x0080 | 128 | Uint16 | R/O | --- |
| 0x0081 | 129 | Uint16 | R/O | Internal SW version |
| 0x0082 | 130 | Uint32 | R/O | VdN number |
| 0x0084 | 132 | Uint32 | R/O | HW version |
| 0x0086 | 134 | Char[22] | R/O | Fabrication number |
| 0x0091 | 145 | Uint16 | R/O | --- |
| | | | | |
| 0x0092 | 146 | Uint32 | R/O | Software version **analog/digital I/O 2** (Option 2) |
| 0x0094 | 148 | Uint16 | R/O | --- |
| 0x0095 | 149 | Uint16 | R/O | Internal SW version |
| 0x0096 | 150 | Uint32 | R/O | VdN number |
| 0x0098 | 152 | Uint32 | R/O | HW version |
| 0x009A | 154 | Char[22] | R/O | Fabrication number |
| 0x00A5 | 165 | Uint16 | R/O | --- |
| | | | | |
| 0x00A6 | 166 | Uint32 | R/O | Software version **digital I/O** (Option 3) |
| 0x00A8 | 168 | Uint16 | R/O | --- |
| 0x00A9 | 169 | Uint16 | R/O | Internal SW version |
| 0x00AA | 170 | Uint32 | R/O | VdN number |
| 0x00AC | 172 | Uint32 | R/O | HW version |
| 0x00AE | 174 | Char[22] | R/O | Fabrication number |
| | | | | |

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x1000 | 4096 | Word | R/O | Time: year |
| 0x1001 | 4097 | Word | R/O | Time: month |
| 0x1002 | 4098 | Word | R/O | Time: day |
| 0x1003 | 4099 | Word | R/O | Time: hour |
| 0x1004 | 4100 | Word | R/O | Time: minute |
| 0x1005 | 4101 | Word | R/O | Time: second |

**Process values and texts**

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x1006 | 4102 | Floating point | R/O | Actual value of analog input 1 |
| 0x1008 | 4104 | Floating point | R/O | Actual value of analog input 2 |
| 0x100A | 4106 | Floating point | R/O | Actual value of analog input 3 |
| 0x100C | 4108 | Floating point | R/O | Actual value of analog input 4 |
| 0x100E | 4110 | Floating point | R/O | Actual value of analog input 5 |
| 0x1010 | 4112 | Floating point | R/O | Actual value of analog input 6 |
| | | | | |
| 0x1012 | 4114 | Floating point | R/O | Internal Pt100 cold junction (in degrees) of analog input 1 |
| 0x1014 | 4116 | Floating point | R/O | Internal Pt100 cold junction (in degrees) of analog input 2 |
| 0x1016 | 4118 | Floating point | R/O | Internal Pt100 cold junction (in degrees) of analog input 3 |
| 0x1018 | 4120 | Floating point | R/O | Internal Pt100 cold junction (in degrees) of analog input 4 |
| 0x101A | 4122 | Floating point | R/O | Internal Pt100 cold junction (in degrees) of analog input 5 |
| 0x101C | 4124 | Floating point | R/O | Internal Pt100 cold junction (in degrees) of analog input 6 |
| | | | | |
| 0x101E | 4126 | Floating point | R/O | Actual value of analog output 1 |
| 0x1020 | 4128 | Floating point | R/O | Actual value of analog output 2 |
| | | | | |
| 0x1022 | 4130 | Floating point | R/O | Flow |
| 0x1024 | 4132 | Floating point | R/O | Flow frequency (revolutions per minute) |
| | | | | |
| 0x1026 | 4134 | Floating point | R/O | Counter status of counter 1 |
| 0x1028 | 4136 | Floating point | R/O | Counter status of counter 2 |
| 0x102A | 4138 | Floating point | R/O | Counter status of counter 3 |
| 0x102C | 4140 | Floating point | R/O | Counter status of counter 4 |
| 0x102E | 4142 | Floating point | R/O | Counter status of counter 5 |
| 0x1030 | 4144 | Floating point | R/O | Counter status of counter 6 |
| | | | | |
| 0x1032 | 4146 | Floating point | R/O | Result of math 1 |
| 0x1034 | 4148 | Floating point | R/O | Result of math 2 |

# 7 Modbus address tables

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x1036 | 4150 | Floating point | R/O | Result of math 3 |
| 0x1038 | 4152 | Floating point | R/O | Result of math 4 |
| 0x103A | 4154 | Floating point | R/O | Result of math 5 |
| 0x103C | 4156 | Floating point | R/O | Result of math 6 |
| | | | | |
| 0x103E | 4158 | Floating point | R/W | Actual value of external analog input 1 |
| 0x1040 | 4160 | Floating point | R/W | Actual value of external analog input 2 |
| 0x1042 | 4162 | Floating point | R/W | Actual value of external analog input 3 |
| 0x1044 | 4164 | Floating point | R/W | Actual value of external analog input 4 |
| 0x1046 | 4166 | Floating point | R/W | Actual value of external analog input 5 |
| 0x1048 | 4168 | Floating point | R/W | Actual value of external analog input 6 |
| 0x104A | 4170 | Floating point | R/W | Actual value of external analog input 7 |
| 0x104C | 4172 | Floating point | R/W | Actual value of external analog input 8 |
| 0x104E | 4174 | Floating point | R/W | Actual value of external analog input 9 |
| 0x1050 | 4176 | Floating point | R/W | Actual value of external analog input 10 |
| 0x1052 | 4178 | Floating point | R/W | Actual value of external analog input 11 |
| 0x1054 | 4180 | Floating point | R/W | Actual value of external analog input 12 |
| 0x1056 | 4182 | Floating point | R/W | Actual value of external analog input 13 |
| 0x1058 | 4184 | Floating point | R/W | Actual value of external analog input 14 |
| 0x105A | 4186 | Floating point | R/W | Actual value of external analog input 15 |
| 0x105C | 4188 | Floating point | R/W | Actual value of external analog input 16 |
| 0x105E | 4190 | Floating point | R/W | Actual value of external analog input 17 |
| 0x1060 | 4192 | Floating point | R/W | Actual value of external analog input 18 |
| 0x1062 | 4194 | Floating point | R/W | Actual value of external analog input 19 |
| 0x1064 | 4196 | Floating point | R/W | Actual value of external analog input 20 |
| 0x1066 | 4198 | Floating point | R/W | Actual value of external analog input 21 |
| 0x1068 | 4200 | Floating point | R/W | Actual value of external analog input 22 |
| 0x106A | 4202 | Floating point | R/W | Actual value of external analog input 23 |
| 0x106C | 4204 | Floating point | R/W | Actual value of external analog input 24 |
| | | | | |
| 0x106E | 4206 | Word | R/O | High-speed counter |
| | | | | |
| 0x106F | 4207 | Bool | R/O | Alarm 1 of analog input 1 |
| 0x1070 | 4208 | Bool | R/O | Alarm 2 of analog input 1 |
| 0x1071 | 4209 | Bool | R/O | Alarm 1 of analog input 2 |
| 0x1072 | 4210 | Bool | R/O | Alarm 2 of analog input 2 |
| 0x1073 | 4211 | Bool | R/O | Alarm 1 of analog input 3 |
| 0x1074 | 4212 | Bool | R/O | Alarm 2 of analog input 3 |
| 0x1075 | 4213 | Bool | R/O | Alarm 1 of analog input 4 |
| 0x1076 | 4214 | Bool | R/O | Alarm 2 of analog input 4 |
| 0x1077 | 4215 | Bool | R/O | Alarm 1 of analog input 5 |
| 0x1078 | 4216 | Bool | R/O | Alarm 2 of analog input 5 |
| 0x1079 | 4217 | Bool | R/O | Alarm 1 of analog input 6 |
| 0x107A | 4218 | Bool | R/O | Alarm 2 of analog input 6 |

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| | | | | |
| 0x107B | 4219 | Bool | R/O | Alarm 1 of counter 1 |
| 0x107C | 4220 | Bool | R/O | Alarm 2 of counter 1 |
| 0x107D | 4221 | Bool | R/O | Alarm 1 of counter 2 |
| 0x107E | 4222 | Bool | R/O | Alarm 2 of counter 2 |
| 0x107F | 4223 | Bool | R/O | Alarm 1 of counter 3 |
| 0x1080 | 4224 | Bool | R/O | Alarm 2 of counter 3 |
| 0x1081 | 4225 | Bool | R/O | Alarm 1 of counter 4 |
| 0x1082 | 4226 | Bool | R/O | Alarm 2 of counter 4 |
| 0x1083 | 4227 | Bool | R/O | Alarm 1 of counter 5 |
| 0x1084 | 4228 | Bool | R/O | Alarm 2 of counter 5 |
| 0x1085 | 4229 | Bool | R/O | Alarm 1 of counter 6 |
| 0x1086 | 4230 | Bool | R/O | Alarm 2 of counter 6 |
| | | | | |
| 0x1087 | 4231 | Bool | R/O | Status of math 1 |
| 0x1088 | 4232 | Bool | R/O | Status of math 2 |
| 0x1089 | 4233 | Bool | R/O | Status of math 3 |
| 0x108A | 4234 | Bool | R/O | Status of math 4 |
| 0x108B | 4235 | Bool | R/O | Status of math 5 |
| 0x108C | 4236 | Bool | R/O | Status of math 6 |
| | | | | |
| 0x108D | 4237 | Bool | R/O | Alarm 1 of math 1 |
| 0x108E | 4238 | Bool | R/O | Alarm 2 of math 1 |
| 0x108F | 4239 | Bool | R/O | Alarm 1 of math 2 |
| 0x1090 | 4240 | Bool | R/O | Alarm 2 of math 2 |
| 0x1091 | 4241 | Bool | R/O | Alarm 1 of math 3 |
| 0x1092 | 4242 | Bool | R/O | Alarm 2 of math 3 |
| 0x1093 | 4243 | Bool | R/O | Alarm 1 of math 4 |
| 0x1094 | 4244 | Bool | R/O | Alarm 2 of math 4 |
| 0x1095 | 4245 | Bool | R/O | Alarm 1 of math 5 |
| 0x1096 | 4246 | Bool | R/O | Alarm 2 of math 5 |
| 0x1097 | 4247 | Bool | R/O | Alarm 1 of math 6 |
| 0x1098 | 4248 | Bool | R/O | Alarm 2 of math 6 |
| | | | | |
| 0x1099 | 4249 | Bool | R/O | Result of logic 1 |
| 0x109A | 4250 | Bool | R/O | Result of logic 2 |
| 0x109B | 4251 | Bool | R/O | Result of logic 3 |
| 0x109C | 4252 | Bool | R/O | Result of logic 4 |
| 0x109D | 4253 | Bool | R/O | Result of logic 5 |
| 0x109E | 4254 | Bool | R/O | Result of logic 6 |
| | | | | |
| 0x109F | 4255 | Bool | R/O | Alarm of logic 1 |
| 0x10A0 | 4256 | Bool | R/O | Alarm of logic 2 |
| 0x10A1 | 4257 | Bool | R/O | Alarm of logic 3 |

# 7 Modbus address tables

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x10A2 | 4258 | Bool | R/O | Alarm of logic 4 |
| 0x10A3 | 4259 | Bool | R/O | Alarm of logic 5 |
| 0x10A4 | 4260 | Bool | R/O | Alarm of logic 6 |
| | | | | |
| 0x10A5 | 4261 | Bool | R/O | Alarm 1 of external analog input 1 |
| 0x10A6 | 4262 | Bool | R/O | Alarm 2 of external analog input 1 |
| 0x10A7 | 4263 | Bool | R/O | Alarm 1 of external analog input 2 |
| 0x10A8 | 4264 | Bool | R/O | Alarm 2 of external analog input 2 |
| 0x10A9 | 4265 | Bool | R/O | Alarm 1 of external analog input 3 |
| 0x10AA | 4266 | Bool | R/O | Alarm 2 of external analog input 3 |
| 0x10AB | 4267 | Bool | R/O | Alarm 1 of external analog input 4 |
| 0x10AC | 4268 | Bool | R/O | Alarm 2 of external analog input 4 |
| 0x10AD | 4269 | Bool | R/O | Alarm 1 of external analog input 5 |
| 0x10AE | 4270 | Bool | R/O | Alarm 2 of external analog input 5 |
| 0x10AF | 4271 | Bool | R/O | Alarm 1 of external analog input 6 |
| 0x10B0 | 4272 | Bool | R/O | Alarm 2 of external analog input 6 |
| 0x10B1 | 4273 | Bool | R/O | Alarm 1 of external analog input 7 |
| 0x10B2 | 4274 | Bool | R/O | Alarm 2 of external analog input 7 |
| 0x10B3 | 4275 | Bool | R/O | Alarm 1 of external analog input 8 |
| 0x10B4 | 4276 | Bool | R/O | Alarm 2 of external analog input 8 |
| 0x10B5 | 4277 | Bool | R/O | Alarm 1 of external analog input 9 |
| 0x10B6 | 4278 | Bool | R/O | Alarm 2 of external analog input 9 |
| 0x10B7 | 4279 | Bool | R/O | Alarm 1 of external analog input 10 |
| 0x10B8 | 4280 | Bool | R/O | Alarm 2 of external analog input 10 |
| 0x10B9 | 4281 | Bool | R/O | Alarm 1 of external analog input 11 |
| 0x10BA | 4282 | Bool | R/O | Alarm 2 of external analog input 11 |
| 0x10BB | 4283 | Bool | R/O | Alarm 1 of external analog input 12 |
| 0x10BC | 4284 | Bool | R/O | Alarm 2 of external analog input 12 |
| 0x10BD | 4285 | Bool | R/O | Alarm 1 of external analog input 13 |
| 0x10BE | 4286 | Bool | R/O | Alarm 2 of external analog input 13 |
| 0x10BF | 4287 | Bool | R/O | Alarm 1 of external analog input 14 |
| 0x10C0 | 4288 | Bool | R/O | Alarm 2 of external analog input 14 |
| 0x10C1 | 4289 | Bool | R/O | Alarm 1 of external analog input 15 |
| 0x10C2 | 4290 | Bool | R/O | Alarm 2 of external analog input 15 |
| 0x10C3 | 4291 | Bool | R/O | Alarm 1 of external analog input 16 |
| 0x10C4 | 4292 | Bool | R/O | Alarm 2 of external analog input 16 |
| 0x10C5 | 4293 | Bool | R/O | Alarm 1 of external analog input 17 |
| 0x10C6 | 4294 | Bool | R/O | Alarm 2 of external analog input 17 |
| 0x10C7 | 4295 | Bool | R/O | Alarm 1 of external analog input 18 |
| 0x10C8 | 4296 | Bool | R/O | Alarm 2 of external analog input 18 |
| 0x10C9 | 4297 | Bool | R/O | Alarm 1 of external analog input 19 |
| 0x10CA | 4298 | Bool | R/O | Alarm 2 of external analog input 19 |
| 0x10CB | 4299 | Bool | R/O | Alarm 1 of external analog input 20 |
| 0x10CC | 4300 | Bool | R/O | Alarm 2 of external analog input 20 |

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x10CD | 4301 | Bool | R/O | Alarm 1 of external analog input 21 |
| 0x10CE | 4302 | Bool | R/O | Alarm 2 of external analog input 21 |
| 0x10CF | 4303 | Bool | R/O | Alarm 1 of external analog input 22 |
| 0x10D0 | 4304 | Bool | R/O | Alarm 2 of external analog input 22 |
| 0x10D1 | 4305 | Bool | R/O | Alarm 1 of external analog input 23 |
| 0x10D2 | 4306 | Bool | R/O | Alarm 2 of external analog input 23 |
| 0x10D3 | 4307 | Bool | R/O | Alarm 1 of external analog input 24 |
| 0x10D4 | 4308 | Bool | R/O | Alarm 2 of external analog input 24 |
| | | | | |
| 0x10D5 | 4309 | Bool | R/O | Value of digital input 1 |
| 0x10D6 | 4310 | Bool | R/O | Value of digital input 2 |
| 0x10D7 | 4311 | Bool | R/O | Value of digital input 3 |
| 0x10D8 | 4312 | Bool | R/O | Value of digital input 4 |
| 0x10D9 | 4313 | Bool | R/O | Value of digital input 5 |
| 0x10DA | 4314 | Bool | R/O | Value of digital input 6 |
| 0x10DB | 4315 | Bool | R/O | Value of digital input 7 |
| 0x10DC | 4316 | Bool | R/O | Value of digital input 8 |
| 0x10DD | 4317 | Bool | R/O | Value of digital input 9 |
| 0x10DE | 4318 | Bool | R/O | Value of digital input 10 |
| 0x10DF | 4319 | Bool | R/O | Value of digital input 11 |
| 0x10E0 | 4320 | Bool | R/O | Value of digital input 12 |
| | | | | |
| 0x10E1 | 4321 | Bool | R/O | Value of digital input/output 1 |
| 0x10E2 | 4322 | Bool | R/O | Value of digital input/output 2 |
| 0x10E3 | 4323 | Bool | R/O | Value of digital input/output 3 |
| 0x10E4 | 4324 | Bool | R/O | Value of digital input/output 4 |
| 0x10E5 | 4325 | Bool | R/O | Value of digital input/output 5 |
| 0x10E6 | 4326 | Bool | R/O | Value of digital input/output 6 |
| 0x10E7 | 4327 | Bool | R/O | Value of digital input/output 7 |
| 0x10E8 | 4328 | Bool | R/O | Value of digital input/output 8 |
| 0x10E9 | 4329 | Bool | R/O | Value of digital input/output 9 |
| 0x10EA | 4330 | Bool | R/O | Value of digital input/output 10 |
| 0x10EB | 4331 | Bool | R/O | Value of digital input/output 11 |
| 0x10EC | 4332 | Bool | R/O | Value of digital input/output 12 |
| | | | | |
| 0x10ED | 4333 | Bool | R/O | Value of relay output |
| | | | | |
| 0x10EE | 4334 | Bool | R/O | Alarm of digital input 1 |
| 0x10EF | 4335 | Bool | R/O | Alarm of digital input 2 |
| 0x10F0 | 4336 | Bool | R/O | Alarm of digital input 3 |
| 0x10F1 | 4337 | Bool | R/O | Alarm of digital input 4 |
| 0x10F2 | 4338 | Bool | R/O | Alarm of digital input 5 |
| 0x10F3 | 4339 | Bool | R/O | Alarm of digital input 6 |
| 0x10F4 | 4340 | Bool | R/O | Alarm of digital input 7 |

# 7 Modbus address tables

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x10F5 | 4341 | Bool | R/O | Alarm of digital input 8 |
| 0x10F6 | 4342 | Bool | R/O | Alarm of digital input 9 |
| 0x10F7 | 4343 | Bool | R/O | Alarm of digital input 10 |
| 0x10F8 | 4344 | Bool | R/O | Alarm of digital input 11 |
| 0x10F9 | 4345 | Bool | R/O | Alarm of digital input 12 |
| | | | | |
| 0x10FA | 4346 | Bool | R/O | Alarm of digital input/output 1 |
| 0x10FB | 4347 | Bool | R/O | Alarm of digital input/output 2 |
| 0x10FC | 4348 | Bool | R/O | Alarm of digital input/output 3 |
| 0x10FD | 4349 | Bool | R/O | Alarm of digital input/output 4 |
| 0x10FE | 4350 | Bool | R/O | Alarm of digital input/output 5 |
| 0x10FF | 4351 | Bool | R/O | Alarm of digital input/output 6 |
| 0x1100 | 4352 | Bool | R/O | Alarm of digital input/output 7 |
| 0x1101 | 4353 | Bool | R/O | Alarm of digital input/output 8 |
| 0x1102 | 4354 | Bool | R/O | Alarm of digital input/output 9 |
| 0x1103 | 4355 | Bool | R/O | Alarm of digital input/output 10 |
| 0x1104 | 4356 | Bool | R/O | Alarm of digital input/output 11 |
| 0x1105 | 4357 | Bool | R/O | Alarm of digital input/output 12 |
| | | | | |
| 0x1106 | 4358 | Bool | R/O | Alarm of external digital input 1 |
| 0x1107 | 4359 | Bool | R/O | Alarm of external digital input 2 |
| 0x1108 | 4360 | Bool | R/O | Alarm of external digital input 3 |
| 0x1109 | 4361 | Bool | R/O | Alarm of external digital input 4 |
| 0x110A | 4362 | Bool | R/O | Alarm of external digital input 5 |
| 0x110B | 4363 | Bool | R/O | Alarm of external digital input 6 |
| 0x110C | 4364 | Bool | R/O | Alarm of external digital input 7 |
| 0x110D | 4365 | Bool | R/O | Alarm of external digital input 8 |
| 0x110E | 4366 | Bool | R/O | Alarm of external digital input 9 |
| 0x110F | 4367 | Bool | R/O | Alarm of external digital input 10 |
| 0x1110 | 4368 | Bool | R/O | Alarm of external digital input 11 |
| 0x1111 | 4369 | Bool | R/O | Alarm of external digital input 12 |
| 0x1112 | 4370 | Bool | R/O | Alarm of external digital input 13 |
| 0x1113 | 4371 | Bool | R/O | Alarm of external digital input 14 |
| 0x1114 | 4372 | Bool | R/O | Alarm of external digital input 15 |
| 0x1115 | 4373 | Bool | R/O | Alarm of external digital input 16 |
| 0x1116 | 4374 | Bool | R/O | Alarm of external digital input 17 |
| 0x1117 | 4375 | Bool | R/O | Alarm of external digital input 18 |
| 0x1118 | 4376 | Bool | R/O | Alarm of external digital input 19 |
| 0x1119 | 4377 | Bool | R/O | Alarm of external digital input 20 |
| 0x111A | 4378 | Bool | R/O | Alarm of external digital input 21 |
| 0x111B | 4379 | Bool | R/O | Alarm of external digital input 22 |
| 0x111C | 4380 | Bool | R/O | Alarm of external digital input 23 |
| 0x111D | 4381 | Bool | R/O | Alarm of external digital input 24 |
| | | | | |

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x111E | 4382 | Bool | R/O | Status of limit value monitoring 1 |
| 0x111F | 4383 | Bool | R/O | Status of limit value monitoring 2 |
| 0x1120 | 4384 | Bool | R/O | Status of limit value monitoring 3 |
| 0x1121 | 4385 | Bool | R/O | Status of limit value monitoring 4 |
| 0x1122 | 4386 | Bool | R/O | Status of limit value monitoring 5 |
| 0x1123 | 4387 | Bool | R/O | Status of limit value monitoring 6 |
| | | | | |
| 0x1124 | 4388 | Bool | R/O | Alarm of limit value monitoring 1 |
| 0x1125 | 4389 | Bool | R/O | Alarm of limit value monitoring 2 |
| 0x1126 | 4390 | Bool | R/O | Alarm of limit value monitoring 3 |
| 0x1127 | 4391 | Bool | R/O | Alarm of limit value monitoring 4 |
| 0x1128 | 4392 | Bool | R/O | Alarm of limit value monitoring 5 |
| 0x1129 | 4393 | Bool | R/O | Alarm of limit value monitoring 6 |
| | | | | |
| 0x112A | 4394 | Bool | R/O | Alarm of upper tolerance band, group 1 |
| 0x112B | 4395 | Bool | R/O | Alarm of upper tolerance band, group 2 |
| 0x112C | 4396 | Bool | R/O | Alarm of upper tolerance band, group 3 |
| 0x112D | 4397 | Bool | R/O | Alarm of upper tolerance band, group 4 |
| 0x112E | 4398 | Bool | R/O | Alarm of lower tolerance band, group 1 |
| 0x112F | 4399 | Bool | R/O | Alarm of lower tolerance band, group 2 |
| 0x1130 | 4400 | Bool | R/O | Alarm of lower tolerance band, group 3 |
| 0x1131 | 4401 | Bool | R/O | Alarm of lower tolerance band, group 4 |
| | | | | |
| 0x1132 | 4402 | Bool | R/O | Alarm 1 of flow |
| 0x1133 | 4403 | Bool | R/O | Alarm 2 of flow |
| | | | | |
| 0x1134 | 4404 | Bool | R/O | Collective alarm |
| 0x1135 | 4405 | Bool | R/O | Collective alarm with acknowledgement |
| 0x1136 | 4406 | Bool | R/O | Memory alarm (internal flash memory) |
| 0x1137 | 4407 | Bool | R/O | User logged on |
| 0x1138 | 4408 | Bool | R/O | Fault |
| 0x1139 | 4409 | Bool | R/O | USB flash drive active |
| 0x113A | 4410 | Bool | R/O | Battery empty |
| 0x113B | 4411 | Bool | R/O | Battery weak |
| 0x113C | 4412 | Bool | R/O | Temperature unit of interface: °F |
| 0x113D | 4413 | Bool | R/O | Temperature unit of device (display): °F |
| 0x113E | 4414 | Bool | R/O | Modbus slave timeout Com1 |
| 0x113F | 4415 | Bool | R/O | Modbus slave timeout Com1, inverted |
| 0x1140 | 4416 | Bool | R/O | Modbus slave timeout TCP1 |
| 0x1141 | 4417 | Bool | R/O | Modbus slave timeout TCP1, inverted |
| 0x1142 | 4418 | Bool | R/O | Modbus slave timeout TCP2 |
| 0x1143 | 4419 | Bool | R/O | Modbus slave timeout TCP3, inverted |
| | | | | |
| 0x1144 | 4420 | Bool | R/W | Value of/for external digital input 1 |

# 7 Modbus address tables

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x1145 | 4421 | Bool | R/W | Value of/for external digital input 2 |
| 0x1146 | 4422 | Bool | R/W | Value of/for external digital input 3 |
| 0x1147 | 4423 | Bool | R/W | Value of/for external digital input 4 |
| 0x1148 | 4424 | Bool | R/W | Value of/for external digital input 5 |
| 0x1149 | 4425 | Bool | R/W | Value of/for external digital input 6 |
| 0x114A | 4426 | Bool | R/W | Value of/for external digital input 7 |
| 0x114B | 4427 | Bool | R/W | Value of/for external digital input 8 |
| 0x114C | 4428 | Bool | R/W | Value of/for external digital input 9 |
| 0x114D | 4429 | Bool | R/W | Value of/for external digital input 10 |
| 0x114E | 4430 | Bool | R/W | Value of/for external digital input 11 |
| 0x114F | 4431 | Bool | R/W | Value of/for external digital input 12 |
| 0x1150 | 4432 | Bool | R/W | Value of/for external digital input 13 |
| 0x1151 | 4433 | Bool | R/W | Value of/for external digital input 14 |
| 0x1152 | 4434 | Bool | R/W | Value of/for external digital input 15 |
| 0x1153 | 4435 | Bool | R/W | Value of/for external digital input 16 |
| 0x1154 | 4436 | Bool | R/W | Value of/for external digital input 17 |
| 0x1155 | 4437 | Bool | R/W | Value of/for external digital input 18 |
| 0x1156 | 4438 | Bool | R/W | Value of/for external digital input 19 |
| 0x1157 | 4439 | Bool | R/W | Value of/for external digital input 20 |
| 0x1158 | 4440 | Bool | R/W | Value of/for external digital input 21 |
| 0x1159 | 4441 | Bool | R/W | Value of/for external digital input 22 |
| 0x115A | 4442 | Bool | R/W | Value of/for external digital input 23 |
| 0x115B | 4443 | Bool | R/W | Value of/for external digital input 24 |
| | | | | |
| 0x115C | 4444 | Char[94] | R/W | External event text, group 1 |
| 0x118B | 4491 | Char[94] | R/W | External event text, group 2 |
| 0x11BA | 4538 | Char[94] | R/W | External event text, group 3 |
| 0x11E9 | 4585 | Char[94] | R/W | External event text, group 4 |
| | | | | |
| 0x1218 | 4632 | Char[64] | R/W | External text variable 1 |
| 0x1238 | 4664 | Char[64] | R/W | External text variable 2 |
| 0x1258 | 4696 | Char[64] | R/W | External text variable 3 |
| 0x1278 | 4728 | Char[64] | R/W | External text variable 4 |
| 0x1298 | 4760 | Char[64] | R/W | External text variable 5 |
| 0x12B8 | 4792 | Char[64] | R/W | External text variable 6 |
| 0x12D8 | 4824 | Char[64] | R/W | External text variable 7 |
| 0x12F8 | 4856 | Char[64] | R/W | External text variable 8 |
| 0x1318 | 4888 | Char[64] | R/W | External text variable 9 |
| 0x1338 | 4920 | Char[64] | R/W | External text variable 10 |
| | | | | |
| | | | | As of device version 02: |
| 0x1358 | 4952 | Char[244] | R/W | External event text (long), group 1 |
| 0x13D2 | 5074 | Char[244] | R/W | External event text (long), group 2 |
| 0x144C | 5196 | Char[244] | R/W | External event text (long), group 3 |

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x14C6 | 5318 | Char[244] | R/W | External event text (long), group 4 |
| | | | | |
| | | | | As of device version 02: |
| 0x1540 | 5440 | Char[244] | R/W | External text variable (long) 1 |
| 0x15BA | 5562 | Char[244] | R/W | External text variable (long) 2 |
| 0x1634 | 5684 | Char[244] | R/W | External text variable (long) 3 |
| 0x16AE | 5806 | Char[244] | R/W | External text variable (long) 4 |
| 0x1728 | 5928 | Char[244] | R/W | External text variable (long) 5 |
| 0x17A2 | 6050 | Char[244] | R/W | External text variable (long) 6 |
| 0x181C | 6172 | Char[244] | R/W | External text variable (long) 7 |
| 0x1896 | 6294 | Char[244] | R/W | External text variable (long) 8 |
| 0x1910 | 6416 | Char[244] | R/W | External text variable (long) 9 |
| 0x198A | 6538 | Char[244] | R/W | External text variable (long) 10 |
| | | | | |
| | | | | As of device version 02: |
| 0x1A04 | 6660 | Bool | R/O | Limit value monitoring 1: status |
| 0x1A05 | 6661 | Bool | R/O | Limit value monitoring 2: status |
| 0x1A06 | 6662 | Bool | R/O | Limit value monitoring 3: status |
| 0x1A07 | 6663 | Bool | R/O | Limit value monitoring 4: status |
| 0x1A08 | 6664 | Bool | R/O | Limit value monitoring 5: status |
| 0x1A09 | 6665 | Bool | R/O | Limit value monitoring 6: status |
| 0x1A0A | 6666 | Bool | R/O | Limit value monitoring 7: status |
| 0x1A0B | 6667 | Bool | R/O | Limit value monitoring 8: status |
| 0x1A0C | 6668 | Bool | R/O | Limit value monitoring 9: status |
| 0x1A0D | 6669 | Bool | R/O | Limit value monitoring 10: status |
| 0x1A0E | 6670 | Bool | R/O | Limit value monitoring 11: status |
| 0x1A0F | 6671 | Bool | R/O | Limit value monitoring 12: status |
| 0x1A10 | 6672 | Bool | R/O | Limit value monitoring 13: status |
| 0x1A11 | 6673 | Bool | R/O | Limit value monitoring 14: status |
| 0x1A12 | 6674 | Bool | R/O | Limit value monitoring 15: status |
| 0x1A13 | 6675 | Bool | R/O | Limit value monitoring 16: status |
| 0x1A14 | 6676 | Bool | R/O | Limit value monitoring 17: status |
| 0x1A15 | 6677 | Bool | R/O | Limit value monitoring 18: status |
| 0x1A16 | 6678 | Bool | R/O | Limit value monitoring 19: status |
| 0x1A17 | 6679 | Bool | R/O | Limit value monitoring 20: status |
| 0x1A18 | 6680 | Bool | R/O | Limit value monitoring 21: status |
| 0x1A19 | 6681 | Bool | R/O | Limit value monitoring 22: status |
| 0x1A1A | 6682 | Bool | R/O | Limit value monitoring 23: status |
| 0x1A1B | 6683 | Bool | R/O | Limit value monitoring 24: status |
| 0x1A1C | 6684 | Bool | R/O | Limit value monitoring 1: alarm |
| 0x1A1D | 6685 | Bool | R/O | Limit value monitoring 2: alarm |
| 0x1A1E | 6686 | Bool | R/O | Limit value monitoring 3: alarm |
| 0x1A1F | 6687 | Bool | R/O | Limit value monitoring 4: alarm |
| 0x1A20 | 6688 | Bool | R/O | Limit value monitoring 5: alarm |

# 7 Modbus address tables

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x1A21 | 6689 | Bool | R/O | Limit value monitoring 6: alarm |
| 0x1A22 | 6690 | Bool | R/O | Limit value monitoring 7: alarm |
| 0x1A23 | 6691 | Bool | R/O | Limit value monitoring 8: alarm |
| 0x1A24 | 6692 | Bool | R/O | Limit value monitoring 9: alarm |
| 0x1A25 | 6693 | Bool | R/O | Limit value monitoring 10: alarm |
| 0x1A26 | 6694 | Bool | R/O | Limit value monitoring 11: alarm |
| 0x1A27 | 6695 | Bool | R/O | Limit value monitoring 12: alarm |
| 0x1A28 | 6696 | Bool | R/O | Limit value monitoring 13: alarm |
| 0x1A29 | 6697 | Bool | R/O | Limit value monitoring 14: alarm |
| 0x1A2A | 6698 | Bool | R/O | Limit value monitoring 15: alarm |
| 0x1A2B | 6699 | Bool | R/O | Limit value monitoring 16: alarm |
| 0x1A2C | 6700 | Bool | R/O | Limit value monitoring 17: alarm |
| 0x1A2D | 6701 | Bool | R/O | Limit value monitoring 18: alarm |
| 0x1A2E | 6702 | Bool | R/O | Limit value monitoring 19: alarm |
| 0x1A2F | 6703 | Bool | R/O | Limit value monitoring 20: alarm |
| 0x1A30 | 6704 | Bool | R/O | Limit value monitoring 21: alarm |
| 0x1A31 | 6705 | Bool | R/O | Limit value monitoring 22: alarm |
| 0x1A32 | 6706 | Bool | R/O | Limit value monitoring 23: alarm |
| 0x1A33 | 6707 | Bool | R/O | Limit value monitoring 24: alarm |
| | | | | |
| 0xA000 | 40960 | Char[1204] | R/W | Batch recipe |
| | | | | |
| 0xA600 | 42496 | Char[480] | R/O | Batch text, line 1 |
| 0xA800 | 43008 | Char[480] | R/O | Batch text, line 2 |
| 0xAA00 | 43520 | Char[480] | R/O | Batch text, line 3 |
| 0xAC00 | 44032 | Char[480] | R/O | Batch text, line 4 |
| 0xAE00 | 44544 | Char[480] | R/O | Batch text, line 5 |
| 0xB000 | 45056 | Char[480] | R/O | Batch text, line 6 |
| 0xB200 | 45568 | Char[480] | R/O | Batch text, line 7 |
| 0xB400 | 46080 | Char[480] | R/O | Batch text, line 8 |
| 0xB600 | 46592 | Char[480] | R/O | Batch text, line 9 |
| 0xB800 | 47104 | Char[480] | R/O | Batch text, line 10 |

**NOTE!**

As of device version 02, the external text variables and the external event texts can also be transmitted as long text. For this, separate Modbus addresses are available. It must be ensured in the application that only one of the two possible addresses of a variable is used.

**Configuration parameters**

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| | | | | As of device version 02: |
| 1A34 | 6708 | Float | R/W | Limit value monitoring 1: limit value |
| 1A36 | 6710 | Float | R/W | Limit value monitoring 2: limit value |
| 1A38 | 6712 | Float | R/W | Limit value monitoring 3: limit value |

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 1A3A | 6714 | Float | R/W | Limit value monitoring 4: limit value |
| 1A3C | 6716 | Float | R/W | Limit value monitoring 5: limit value |
| 1A3E | 6718 | Float | R/W | Limit value monitoring 6: limit value |
| 1A40 | 6720 | Float | R/W | Limit value monitoring 7: limit value |
| 1A42 | 6722 | Float | R/W | Limit value monitoring 8: limit value |
| 1A44 | 6724 | Float | R/W | Limit value monitoring 9: limit value |
| 1A46 | 6726 | Float | R/W | Limit value monitoring 10: limit value |
| 1A48 | 6728 | Float | R/W | Limit value monitoring 11: limit value |
| 1A4A | 6730 | Float | R/W | Limit value monitoring 12: limit value |
| 1A4C | 6732 | Float | R/W | Limit value monitoring 13: limit value |
| 1A4E | 6734 | Float | R/W | Limit value monitoring 14: limit value |
| 1A50 | 6736 | Float | R/W | Limit value monitoring 15: limit value |
| 1A52 | 6738 | Float | R/W | Limit value monitoring 16: limit value |
| 1A54 | 6740 | Float | R/W | Limit value monitoring 17: limit value |
| 1A56 | 6742 | Float | R/W | Limit value monitoring 18: limit value |
| 1A58 | 6744 | Float | R/W | Limit value monitoring 19: limit value |
| 1A5A | 6746 | Float | R/W | Limit value monitoring 20: limit value |
| 1A5C | 6748 | Float | R/W | Limit value monitoring 21: limit value |
| 1A5E | 6750 | Float | R/W | Limit value monitoring 22: limit value |
| 1A60 | 6752 | Float | R/W | Limit value monitoring 23: limit value |
| 1A62 | 6754 | Float | R/W | Limit value monitoring 24: limit value |
| 1A64 | 6756 | Float | R/W | Limit value monitoring 1: switching differential |
| 1A66 | 6758 | Float | R/W | Limit value monitoring 2: switching differential |
| 1A68 | 6760 | Float | R/W | Limit value monitoring 3: switching differential |
| 1A6A | 6762 | Float | R/W | Limit value monitoring 4: switching differential |
| 1A6C | 6764 | Float | R/W | Limit value monitoring 5: switching differential |
| 1A6E | 6766 | Float | R/W | Limit value monitoring 6: switching differential |
| 1A70 | 6768 | Float | R/W | Limit value monitoring 7: switching differential |
| 1A72 | 6770 | Float | R/W | Limit value monitoring 8: switching differential |
| 1A74 | 6772 | Float | R/W | Limit value monitoring 9: switching differential |
| 1A76 | 6774 | Float | R/W | Limit value monitoring 10: switching differential |
| 1A78 | 6776 | Float | R/W | Limit value monitoring 11: switching differential |
| 1A7A | 6778 | Float | R/W | Limit value monitoring 12: switching differential |
| 1A7C | 6780 | Float | R/W | Limit value monitoring 13: switching differential |
| 1A7E | 6782 | Float | R/W | Limit value monitoring 14: switching differential |
| 1A80 | 6784 | Float | R/W | Limit value monitoring 15: switching differential |
| 1A82 | 6786 | Float | R/W | Limit value monitoring 16: switching differential |
| 1A84 | 6788 | Float | R/W | Limit value monitoring 17: switching differential |
| 1A86 | 6790 | Float | R/W | Limit value monitoring 18: switching differential |
| 1A88 | 6792 | Float | R/W | Limit value monitoring 19: switching differential |
| 1A8A | 6794 | Float | R/W | Limit value monitoring 20: switching differential |
| 1A8C | 6796 | Float | R/W | Limit value monitoring 21: switching differential |
| 1A8E | 6798 | Float | R/W | Limit value monitoring 22: switching differential |
| 1A90 | 6800 | Float | R/W | Limit value monitoring 23: switching differential |

# 7 Modbus address tables

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 1A92 | 6802 | Float | R/W | Limit value monitoring 24: switching differential |

**Reading frames**

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x8000 | 32768 | Byte[254] | R/W | Frame 1 reading |
| 0x8080 | 32896 | Byte[254] | R/W | Frame 2 reading |
| 0x8100 | 33024 | Byte[254] | R/W | Frame 3 reading |
| 0x8180 | 33152 | Byte[254] | R/W | Frame 4 reading |
| 0x8200 | 33280 | Byte[254] | R/W | Frame 5 reading |
| 0x8280 | 33408 | Byte[254] | R/W | Frame 6 reading |
| 0x8300 | 33536 | Byte[254] | R/W | Frame 7 reading |
| 0x8380 | 33664 | Byte[254] | R/W | Frame 8 reading |
| 0x8400 | 33792 | Byte[254] | R/W | Frame 9 reading |
| 0x8480 | 33920 | Byte[254] | R/W | Frame 10 reading |
| 0x8500 | 34048 | Byte[254] | R/W | Frame 11 reading |
| 0x8580 | 34176 | Byte[254] | R/W | Frame 12 reading |
| 0x8600 | 34304 | Byte[254] | R/W | Frame 13 reading |
| 0x8680 | 34432 | Byte[254] | R/W | Frame 14 reading |
| 0x8700 | 34560 | Byte[254] | R/W | Frame 15 reading |
| 0x8780 | 34688 | Byte[254] | R/W | Frame 16 reading |
| 0x8800 | 34816 | Byte[254] | R/W | Frame 17 reading |
| 0x8880 | 34944 | Byte[254] | R/W | Frame 18 reading |
| 0x8900 | 35072 | Byte[254] | R/W | Frame 19 reading |
| 0x8980 | 35200 | Byte[254] | R/W | Frame 20 reading |
| 0x8A00 | 35328 | Byte[254] | R/W | Frame 21 reading |
| 0x8A80 | 35456 | Byte[254] | R/W | Frame 22 reading |
| 0x8B00 | 35584 | Byte[254] | R/W | Frame 23 reading |
| 0x8B80 | 35712 | Byte[254] | R/W | Frame 24 reading |
| 0x8C00 | 35840 | Byte[254] | R/W | Frame 25 reading |
| 0x8C80 | 35968 | Byte[254] | R/W | Frame 26 reading |
| 0x8D00 | 36096 | Byte[254] | R/W | Frame 27 reading |
| 0x8D80 | 36224 | Byte[254] | R/W | Frame 28 reading |
| 0x8E00 | 36352 | Byte[254] | R/W | Frame 29 reading |
| 0x8E80 | 36480 | Byte[254] | R/W | Frame 30 reading |
| 0x8F00 | 36608 | Byte[254] | R/W | Frame 31 reading |
| 0x8F80 | 36736 | Byte[254] | R/W | Frame 32 reading |

**Writing frames**

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x9000 | 36864 | Byte[254] | R/O | Frame 1 writing |
| 0x9080 | 36992 | Byte[254] | R/O | Frame 2 writing |
| 0x9100 | 37120 | Byte[254] | R/O | Frame 3 writing |

| Address | | Data type | Access | Description |
|---|---|---|---|---|
| Hex. | Dec. | | | |
| 0x9180 | 37248 | Byte[254] | R/O | Frame 4 writing |
| 0x9200 | 37376 | Byte[254] | R/O | Frame 5 writing |
| 0x9280 | 37504 | Byte[254] | R/O | Frame 6 writing |
| 0x9300 | 37632 | Byte[254] | R/O | Frame 7 writing |
| 0x9380 | 37760 | Byte[254] | R/O | Frame 8 writing |
| 0x9400 | 37888 | Byte[254] | R/O | Frame 9 writing |
| 0x9480 | 38016 | Byte[254] | R/O | Frame 10 writing |
| 0x9500 | 38144 | Byte[254] | R/O | Frame 11 writing |
| 0x9580 | 38272 | Byte[254] | R/O | Frame 12 writing |
| 0x9600 | 38400 | Byte[254] | R/O | Frame 13 writing |
| 0x9680 | 38528 | Byte[254] | R/O | Frame 14 writing |
| 0x9700 | 38656 | Byte[254] | R/O | Frame 15 writing |
| 0x9780 | 38784 | Byte[254] | R/O | Frame 16 writing |
| 0x9800 | 38912 | Byte[254] | R/O | Frame 17 writing |
| 0x9880 | 39040 | Byte[254] | R/O | Frame 18 writing |
| 0x9900 | 39168 | Byte[254] | R/O | Frame 19 writing |
| 0x9980 | 39296 | Byte[254] | R/O | Frame 20 writing |
| 0x9A00 | 39424 | Byte[254] | R/O | Frame 21 writing |
| 0x9A80 | 39552 | Byte[254] | R/O | Frame 22 writing |
| 0x9B00 | 39680 | Byte[254] | R/O | Frame 23 writing |
| 0x9B80 | 39808 | Byte[254] | R/O | Frame 24 writing |
| 0x9C00 | 39936 | Byte[254] | R/O | Frame 25 writing |
| 0x9C80 | 40064 | Byte[254] | R/O | Frame 26 writing |
| 0x9D00 | 40192 | Byte[254] | R/O | Frame 27 writing |
| 0x9D80 | 40320 | Byte[254] | R/O | Frame 28 writing |
| 0x9E00 | 40448 | Byte[254] | R/O | Frame 29 writing |
| 0x9E80 | 40576 | Byte[254] | R/O | Frame 30 writing |
| 0x9F00 | 40704 | Byte[254] | R/O | Frame 31 writing |
| 0x9F80 | 40832 | Byte[254] | R/O | Frame 32 writing |

# 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 CM 20 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