WEST
Control Solutions

DataVU 7 - Interface Manual Modbus

59494/1

Contents

Contents

1.1 Preface

Please read this interface description before commissioning the interfaces of the device. Keep the interface description in a place accessible to all users at all times.

Please assist us in improving this Interface description.

Your comments are highly appreciated.

All information required to operate the interface is described in the supplied Interface description. Nevertheless, should problems be encountered during commissioning/start-up, please refrain from carrying out any inadmissible manipulations. This could jeopardize your warranty rights!

Please contact the nearest subsidiary or the head office.

When returning controller modules, assemblies or components, please observe the regulations according to DIN EN 61340-5-1 and DIN EN 61340-5-2 „Protection of electrostatic sensitive devices“. For transport, only use **ESD** packages.

Please note that we cannot accept any liability for damage caused by ESD.

ESD=**E**lectro **S**tatic **D**ischarge

1.2 Typographical conventions

1.2.1 Warning signs

The symbols for **Caution** and **Attention** are used in this operating manual under the following conditions:

Caution This symbol is used when there may be **danger to personnel** if the instructions are ignored or not followed correctly!

Attention This symbol is used when there may be **damage to equipment or data** if the instructions are ignored or not followed correctly!

Attention This symbol is used where special care is required when handling components liable to damage through electrostatic discharge.

1.2.2 Note symbols

Note This symbol is used when your **special attention** is drawn to a remark.

abc[1] **Footnote** Footnotes are remarks that refer to specific points in the text. Footnotes consist of two parts:

A marker in the text and the foot note text itself.

The markers in the text are arranged as continuous superscript numbers.

1.2.3 Number types

0x0010 **Hexadecimal number** A hexadecimal number is identified by „0x“ preceding the actual number (here: 16 decimal).

2.1 Target group

This operating manual is intended for user, who want to use the paperless recorder as a Modbus slave and read out data from a Modbus master (e.g. PLC).

2.2 Interfaces

ex-factory

The paperless recorder is equipped with several interfaces as a standard:

- Serial interface RS232 or RS485
- Serial interface RS232 (barcode scanner)
- Ethernet 10 / 100 MBit/s
- 2 USB host and 2 USB device interfaces

The serial and the ethernet interfaces serves for communication with a bus system or PC. They can be used, for example, to read out the measured values and/or instrument and process data from the paperless recorder. In connection with the ethernet interface and a PC webbrowser, recorders can be monitored via the internet. Two RS232 interfaces are available. Both interfaces are to be considered as equal.

The USB interfaces are fitted on the front and the back and intended for the operation with the setup program or the PCA communication software (PCC) or for reading out data via the USB memory stick. Only use one USB host or USB device interface at a time. Instruments with stainless steel fronts are not equipped with USB interfaces on the front.

Option

The paperless recorder is equipped with several interfaces as an option:

- PROFIBUS-DP interface

2.3 System prerequisites

The following items are required for interface operation:

- Connection line:
  for RS232
  Interface cable RS232 (9/9-pin)
  Interface cable RS232 (9/25-pin)
  for ethernet, e.g.
  RJ 45 patch cable, CAT 5 or higher (cross over)
- Setup or evaluation program, e.g.
  Setup program
  PC evaluation software PCA3000
  PCA communication software PCC
- PC or notebook

2 General information

3.1 Connection position

Front view of the paperless recorder

USB host
for the data exchange (measured data, configuration data, user lists) between paperless recorder and PC

USB device
for communication with the setup program or PCC

Connection diagram

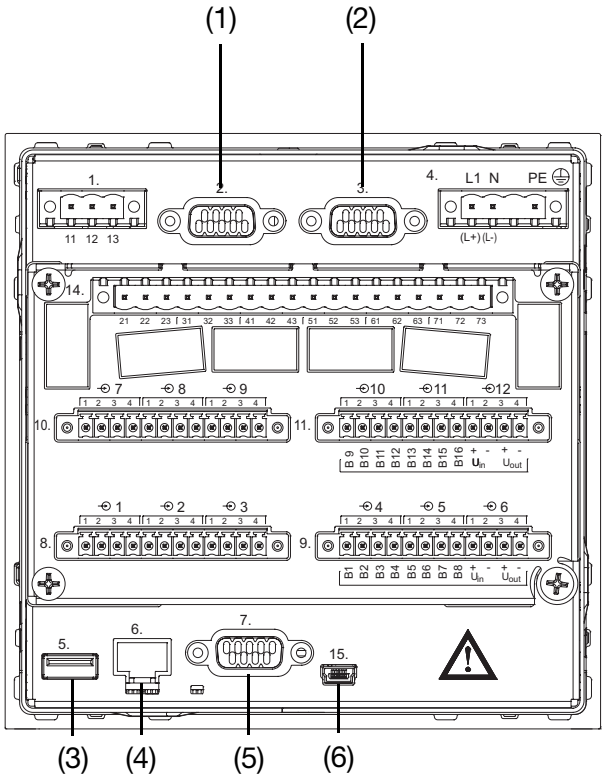
Use a USB interface either at the front or at the rear! Only use one host or device interface at a time.

USB host = Plug type A

USB device = Plug type B mini

3 Connecting the interface

Rear view of the paperless recorder



(1) RS232 interface for barcode scanner (serial)
(2) PROFIBUS-DP interface (optional)
(3) USB host interface (serial)
(4) Ethernet interface (serial)
(5) RS232/485 interface (serial)
(6) USB device interface (serial)

Connection diagram RS232/RS485

RS232 (plugs (1) + (5))		RS485 (plug (5))	
1		1	
2	RxD	2	
3	TxD	3	TxD+/RxD+
4		4	
5	GND	5	GND
6		6	
7		7	
8		8	TxD-/RxD-
9		9	

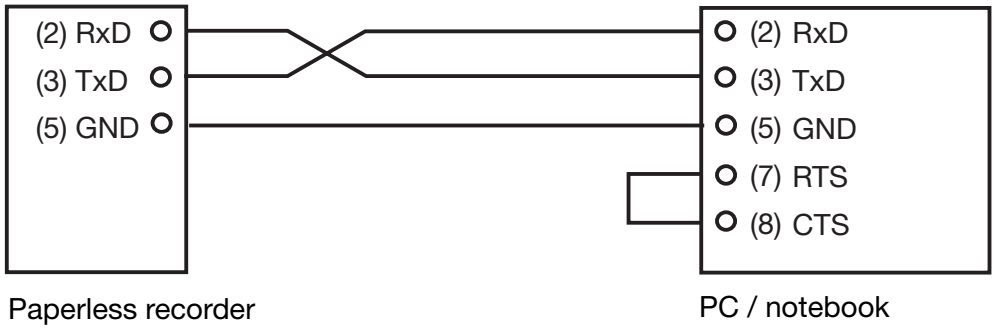
We recommend the use of a twisted connection cable with shielding!
Only connect the signals specified above, otherwise errors will occur!

3.2 RS232

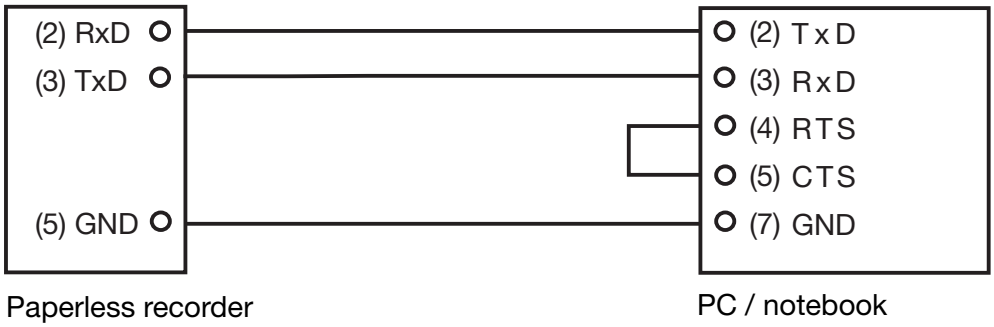
The RS232 interface does not use the handshake lines (RTS, CTS). The RTS line (CTS on the paperless recorder) coming from the master (PC or notebook) is not taken into consideration. The response is immediately transmitted from the paperless recorder. The CTS line of the master (RTS on the paperless recorder) remains open.

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

PC COM interface with 9-pin Sub-D socket



PC COM interface with 25-pin Sub-D socket



3.3 Toggling between RS232 and RS485

Toggling between RS232 and RS485 interface is carried out by means of the paperless recorder parameter
Configuration ➔ *Interface* ➔ *RS232/RS485* ➔ *General* ➔ *Type*
or via the setup program
Edit ➔ *Serial interface* ➔ *RS232/RS485* ➔ *Type*

3 Connecting the interface

3.4 Configuration of the serial interfaces

Configuration on the paperless recorder

✱ Select *Configuration* ➔ *Interface* ➔ *RS232/RS485* ➔ *General* on the paperless recorder.
Now the parameters required to configure the interface are available.

Configuration via the setup program

The *Edit* ➔ *Serial interface* ➔ *RS232/RS485* menu point is used for the configuration by means of the setup software.

	Parameters	Value/Selection	Description
Device address	➔ Device address	**1**…254	see Chapter 4.5 Device address, Page 18.
Interface type	➔ Type	**RS232**, RS485	Can only be edited for RS232/RS485. See Chapter 3.3 Toggling between RS232 and RS485, Page 11.
Protocol	➔ Protocol	**Modbus slave**, Modbus master, Barcode	see Chapter 5 Serial protocol types, Page 35.
Baud rate	➔ Baud rate	9**600 baud**, 19200 baud, 38400 baud	see Chapter 4.3 Timing of the communication, Page 16.
Transmission mode (RTU)	➔ Data format	**8 - 1 - no Parity**, 8 -1 - odd Parity, 8 -1 - even Parity	see Chapter 4.2 Transmission mode (RTU), Page 15.
Min. response time	➔ Min. response time	**0**…500ms	see Chapter 4.3 Timing of the communication, Page 16.

For communication via the RS232 interfaces, the device address has to be taken into account although it is not a bus interface.

The serial interfaces as well as the USB and Ethernet interface can be operated simultaneously. Only on the protocol level, some inhibits are present (e.g. only one setup may be written at a time).

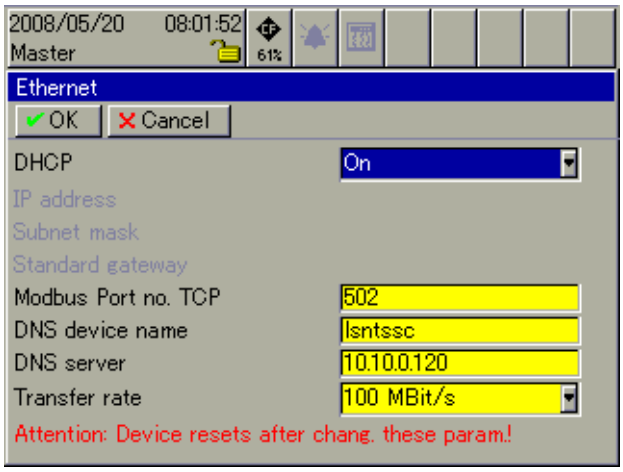
3.5 Configuration of the Ethernet interface

Configuration on the paperless recorder

is carried out using the *Configuration* ➔ *Interface* ➔ *Ethernet* menu point.

Configuration via the setup program

is carried out using the *Extras* ➔ *Ethernet interface* menu point.



3.5.1 Parameters

DHCP

DHCP (Dynamic Host Configuration Protocol) is used to ensure that the paperless recorder is automatically assigned an IP address and further communication parameters by a DHCP server.

On	DHCP is switched on, the paperless recorder obtains its IP address by the DHCP server

The further communication parameters usually assigned to the paperless recorder by the DHCP server include, among others, the subnet mask, the standard gateway address and the so-called lease time.

After the lease time (period of use) has elapsed, the validity of the IP address expires. To ensure that the paperless recorder always has a valid IP address, after 50% of the lease time it transmits a query to the known DHCP server asking whether or not the address is still valid. If the DHCP server cannot be accessed, the paperless recorder repeats its query until 87.5% of the lease time has elapsed. Then the paperless recorder transmits its query not only to the DHCP server but also to the entire network. When the lease time has elapsed and the IP address is still not acknowledged, the paperless recorder declares the address as invalid and is no longer accessible via the net.

The DHCP server can change an assigned address. If, for example, automatic data download with the PCA communication software is used, the address must be changed within the software.

The automatically assigned IP address query can be inquired in the *Device manager* ➔ *Device info* ➔ *Ethernet info* menu.

IP address

If automatic IP address assignment is not used ("DHCP = OFF"), the IP address of the paperless recorder is set here.

Subnet mask

If automatic IP address assignment is not used ("DHCP = OFF"), the subnet mask is set here.

The subnet mask is used to group devices (PC, paperless recorder, etc.) into subnets. All devices, the IP address of which is linked with AND identical to the subnet mask, belong to one subnet and can communicate with each other.

If devices are to be accessed outside the subnet, communication must be carried out via a gateway (standard gateway).

Standard Gateway

If automatic IP address assignment is not used ("DHCP = OFF"), the address of the standard gateway is set here.

The standard gateway is used for communication by devices which are not part in a subnet.

Port Modbus TCP

The port address must be set when the paperless recorder is accessed by a visualization software and the Modbus TCP protocol (Modbus tunneling: external frame Ethernet, internal frame Modbus) is used.

Changes to this parameter will become effective after the paperless recorder has been restarted.

DNS Device name

The DNS device name is entered here. This ensures that the device can be accessed not only via its IP address but also via its name.

DNS server

Here the IP address of a DNS server installed in the network is set. The DNS server is required to convert the name for Email transmittion via Ethernet.

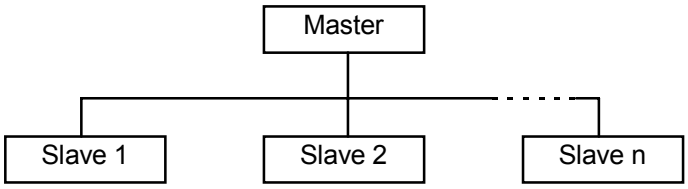
Changes to this parameter will become effective after the paperless recorder has been restarted.

Transfer rate

The transfer rate used by the paperless recorder to communicate with the DHCP server and other PCs is configured here.

4.1 Master-Slave principle

Communication between a master (PC or notebook) and a slave (paperless recorder) using the Modbus / J-bus 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.

The paperless recorder can be operated either as a Modbus slave, see Chapter 5.1 Modbus slave, Page 35, or as a Modbus master, see Chapter 5.2 Modbus master, Page 38.

In a Modbus network, only one device can be assigned with the master function.

4.2 Transmission mode (RTU)

The transmission mode used is the RTU mode (Remote Terminal Unit). The data is transmitted in the binary format (hexadecimal) with 8 or 16 bits for integer values and 32 bits for float values.

Data format

The data format describes the structure of a byte transmitted.

Data word	Parity bit	Stop bit	Number of bits
8 bits	no	1	9
8 bits	even	1	10
8 bits	odd	1	10

The data format to be used can be set, see Chapter 3.4 Configuration of the serial interfaces, Page 12.

4 Modbus protocol description

4.3 Timing of the communication

Character transmission time

Start and end of a data block are marked by transmission pauses. The character transmission time (time required to transmit one single character) depends on the baud rate and the data format used.

For a data format of 8 data bits, no parity bit and one stop bit, this is:

Character transmission time [ms] = 1000 * 9 bit / baud rate

For the other data formats, this is:

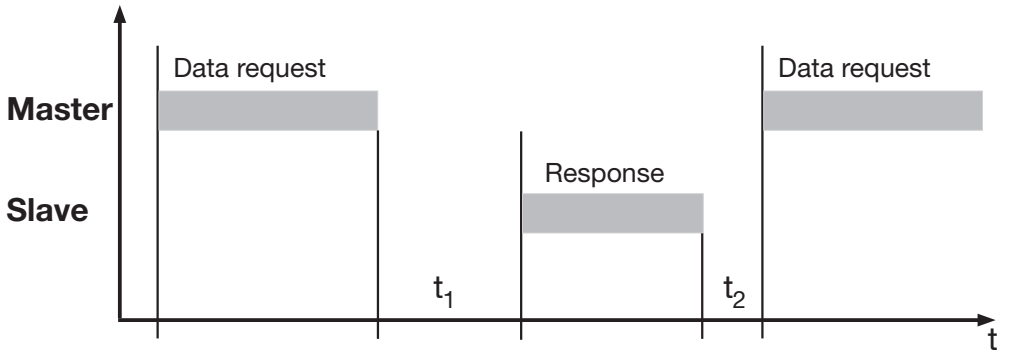
Character transmission time [ms] = 1000 * 10 bit / baud rate

Example

Baud rate [baud]	**Data format** [bit]	**character transmission time** [ms]
38400	10	0,260
	9	0,234
19200	10	0,521
	9	0,469
9600	10	1,042
	9	0,938

Timing scheme

A data request runs according to the following timing scheme:



t_1 Internal waiting time of the paperless recorder prior to checking the data request and the internal processing time.

min.:	5 ms
typical:	5 … 45 ms
max.:	60 ms or set "minimum response time"

A minimum response time can be set in the controller under the menu point *Configuration* ➔ *Interface*. This preset time is the minimum waiting time before an answer is transmitted (0…500 ms). If a smaller value is set, then the response time may be longer than the preset value (internal processing takes longer), the controller answers as soon as internal processing is completed. The preset time of 0 msec means that the controller responds at the maximum possible speed.

The minimum response time, which can be set is required by the RS485 interface in the master, to be able to switch over the interface drivers from transmit to receive. This parameter is not required for the RS232 interface.

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

for RS232	min. 3.5 times the transmission time required for 1 character (the time depends on the baud rate)
for RS485	60 ms

No data requests from the master are permitted during t_1 and t_2 , because the paperless recorder will either ignore the request or declare it to be invalid.

4.4 Structure of a Modbus telegram

Data structure All telegrams have the same structure:

Slave address	Function code	Data field	Checksum CRC16
1 byte	1 byte	x byte	2 byte

Each telegram contains four fields:

Slave address	device address of a specific paperless recorder
Function code	function selection (read, write words)
Data field	Contains information (according to the function code): - Word address / bit address - Number of words / bits - word / bit values
Checksum	detection of transmission errors

4 Modbus protocol description

4.5 Device address

The device address of the paperless recorder can be set between 1 and 254 (decimal), see Chapter 3.4 Configuration of the serial interfaces, Page 12.

A maximum of 31 paperless recorders can be addressed via the RS 485 interface.

The device address 0 is reserved as the Modbus broadcast address.
An instruction of the master to address 0 is carried out by all slaves, but no response is transmitted by them (because this would result in a data collision).

If only **one** paperless recorder is connected to the PC or notebook, it can also be addressed via device address 255 (even if a different device address is configured). The paperless recorder always responds to instructions using device address 255.

In the transmission protocol, the address is specified in the binary format (hexadecimal).

4.6 Function codes

Function overview

The functions described as follows can be used to read out the measured values and further device and process data from the paperless recorder.

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

Please refer to Chapter 4.9 Error messages, Page 28, if the paperless recorder does not react to these functions or emits an error code.

4.6.1 Read n bit

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

Data request

Slave address	Function 0x01 or 0x02	Address first bit	Number of bits	Checksum CRC16
1 byte	1 byte	2 byte	2 byte	2 byte

Response

Slave address	Function 0x01 or 0x02	Number of bit read	Bit value(s)	Checksum CRC16
1 byte	1 byte	1 byte	x byte	2 byte

The response always comes in full bytes of 8 bits each.
Non-requested bit values will be complemented with the 0 value.

Example

Reading a bit starting from bit address 0x0340 (this is word address 0x0034).

For the addresses please refer to Chapter 7.2 Modbus addresses of important device and process data, Page 47.

Data request:

01	02	0340	0001	B85A

Response:

01	02	01	01	6048
			Bit value	

4.6.2 Read n words

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

Data request

Slave address	Function 0x03 or 0x04	Address first word	Number of words	Checksum CRC16
1 byte	1 byte	2 byte	2 byte	2 byte

Response

Slave address	Function 0x03 or 0x04	Number of byte read	Word value(s)	Checksum CRC16
1 byte	1 byte	1 byte	x byte	2 byte

Example

Reading the first 3 analog inputs (these are the first 6 words starting from Modbus address 0x1257).

For the analog input address please refer to Chapter 7.2 Modbus addresses of important device and process data, Page 47.

Data request:

01	03	1257	0006	7160

Response:

01	03	0C	1999	4348	4CCC	4348	2666	4396	8548
			Measured value 1 200,1		Measured value 2 200,3		Measured value 3 300,3		

4.6.3 Write one bit

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

Instruction

Slave address	Function 0x05	Bit address	Bit value xx00	ChecksumCR C16
1 byte	1 byte	2 byte	2 byte xx = 00 ➔ Bit is set to 0 xx = FF ➔ Bit is set to 1	2 byte

Response

Slave address	Function 0x05	Bit address	Bit value	ChecksumCR C16
1 byte	1 byte	2 byte	2 byte	2 byte

Example

Writing a bit starting from bit address 0x0340 (this is bit 0 of the word address 0x0034).

For the addresses please refer to Chapter 7.2 Modbus addresses of important device and process data, Page 47.

Instruction:

01	05	0340	FF00	8DAA

Response (as instruction):

01	05	0340	FF00	8DAA

4.6.4 Write one word

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

Instruction

Slave address	Function 0x06	Word address	Word value	ChecksumCR C16
1 byte	1 byte	2 byte	2 byte	2 byte

Response

Slave address	Function 0x06	Word address	Word value	ChecksumCR C16
1 byte	1 byte	2 byte	2 byte	2 byte

Example

Set the variable "External Binary In 1" (Modbus address 0x1638) to 1.

For the addresses please refer to Chapter 7.2 Modbus addresses of important device and process data, Page 47.

Instruction:

01	06	1638	0001	CD8F

Response (as instruction):

01	06	1638	0001	CD8F

4.6.5 Write n words

instruction

Slave Address	Fun-ction 0x10	Address first word	Number of words	Number of words	Number of value(s)	Check sum CRC16
1 byte	1 byte	2 byte	2 byte	1 byte	x byte	2 byte

Response

Slave address	Function 0x10	Address first word	Number of words	ChecksumCRC16
1 byte	1 byte	2 byte	2 byte	2 byte

Example

Writing the word "Test" (ASCII coding: 0x54 0x65 0x73 0x74 0x00) to address 0x148A ff to ensure that this text is entered in the event list of group 1:

Instruction:

01	10	148A	0003	06	54 65 73 74 00 00	9BFA

Response:

01	10	148A	0003	A412

4 Modbus protocol description

4.7 Transmission format (integer, float, double and text values)

Integer values

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

Example

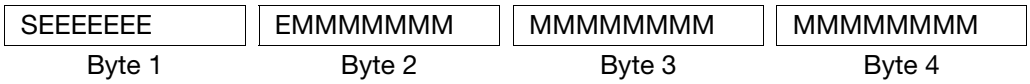
Request of the integer value of address 0x1017, if value "4" (word value 0x0004) is written under this address.

Request: 01031017000130CE (CRC16 = CE30)
Response: 010302**0004**B987 (CRC16 = 87B9)

Float values

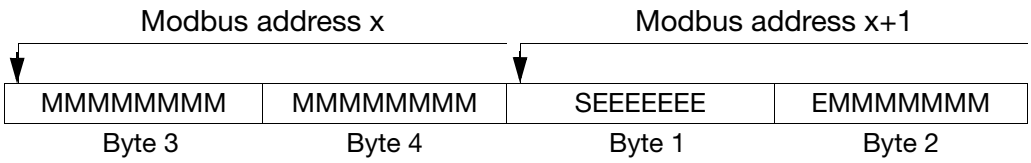
In the case of float values, the Modbus operates with the IEEE-754 standard format (32 bits), the only difference being that byte 1 and 2 are changed over with byte 3 and 4.

Single-float format (32 bit) as per IEEE 754standard



S - sign bit
E - exponent (two's complement)
M - 23 bits normalized mantissa

Modbus float format



Example

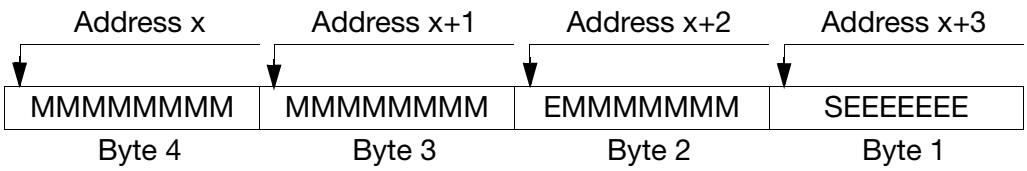
Request of the float value of address 0x0035, if value "550.0" (0x44098000 in IEEE-754 format) is written under this address.

Request: 140300350002D6C0 (CRC16 = C0D6)
Response: 140304**80004409**6434 (CRC16 = 3464)

Once transmission from the device is completed, the bytes of the float value need to be changed over accordingly.

A large number of compilers (e.g. Microsoft Visual C++) file the float values in the following order:

Float value



Please find out the way float values are saved in your application. After the request from the paperless recorder, it might be necessary to change the bytes over in the interface program you are using.

Double values

The double values are also processed in the Modbus in the IEEE-754 standard format (32bit). Unlike float values, bytes are not changed over for double values.

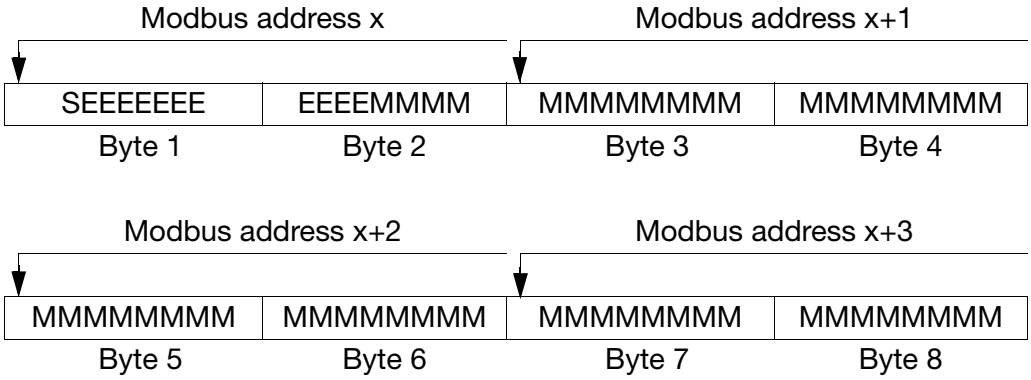
Doupble-float format (32bit) as per IEEE 754standard

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

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

S - sign bit
E - exponent (two's complement)
M - 52bits normalized mantissa

Modbus double format



Example

Request of the double value of address 0x0066, if value 1234567.89 (0x4132D687E3D70A3D in the IEEE-754 format) is written under this address.

Request: 140300660004A6D3 (CRC16 = D3A6)
Response: 140308**4132D687E3D70A3D**E1C1 (CRC16 = C1E1)

Please find out the way double values are saved in your application. After the request from the paperless recorder, it might be necessary to change the bytes over in the program you are using.

Strings (texts) Character strings are transmitted in the ASCII format.

To mark the end, the last character to be transmitted can be a "\0" (ASCII code 0x00). Characters after this mark are without significance.

Knowing that the transmission of texts takes place word by word (16 bits), 0x00 is additionally appended where an odd number of characters is used (incl. "\0").

The maximum length specified in the address tables (see “Address tables” on page 47. ff) for strings also includes the terminating "/0". This means, in the case of "char 11", the text can consist of max. 10 readable characters.

Example Text inquiry from address 0x1000, if the
character string "**LS NT**"
(ASCII code: **0x4C**, **0x53**, **0x20**, **0x4E**, **0x54**, 0x00) is entered under this address.

Request: 01031000000440C9
Response: 010308**4C53204E54**0000AA0D96

Instead of "AA" in front of the CRC sum, there could be any value, because it is behind "/0" and will be ignored.

4.8 Checksum (CRC16)

Calculation scheme

The checksum (CRC16) serves to recognize transmission errors. If an error is identified during evaluation, the device concerned does not respond.

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

The low byte of the check sum is the first to be transmitted!

Example 1

Requesting the status of relay output 1.

Instruction: Read a word from the address 0x1631

01	03	1631	0001	D18D

Response (CRC = 0x8479):

01	03	02	0001	7984
			Word 1	

Word 1 = 1 indicates that relay 1 is active.

4 Modbus protocol description

4.9 Error messages

4.9.1 Modbus error codes

No response by the paperless recorder

In the event of the following errors, the slave will not respond:

- Baud rate and/or data format of Master (PC or notebook) and Slave (paperless recorder) are not compatible.
- The device address of the paperless recorder does not comply with that contained in the protocol.
- The checksum (CRC16) is not correct.
- The instruction from the Master is incomplete or over-defined.
- The number of words to be read is zero.

In these cases the data request should be transmitted again once the timeout time (2 s)
has elapsed.

Error codes

If the data request of the master has been received by the paperless recorder without transmission errors but could not be processed, the paperless recorder will respond with an error code.

The following error codes can be occur:

- 01 invalid error code
- 02 invalid address or
  number of words or bits to be read or written is too large
- 03 Value not within the admissible range
- 08 Value write-protected

Response in the event of an error

Slave address	Function XX OR 80h	Error code	Checksum CRC16
1 byte	1 byte	1 byte	2 byte

0x80 is used to set the function code to its OR status, i.e. the MSB (most significant bit) is set to 1.

Example

Data request:

01	06	1257	0001	FCA2

Response:

01	86	08	43A6

Respons is error code 08 because address 0x1257 is write-protected.

4.9.2 Error messages for invalid values

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

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

Example

Data request:

01	03	1259	0002	1160

Response:

01	03	04	8E52	7DB4	51ED

The measured value 0x7DB48E52 (=3.0×10^{37}) supplied by analog input 2 (Modbus address 0x1259) indicates that the input value is invalid.

4.9.3 Error codes as integer return values

For some lengthy sequences (e.g. email transfer) an error code is entered at the end into an event field or the event list.

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	Segment cannot be deleted
5	Checksum cannot be saved
6	Checksum cannot be read
7	Program cannot be copied
8	Segment cannot be copied
9	Program checksum error
10	Program pointer tab. checksum error
11	Program memory end
12	Segment does not exist
13	Repeat jump marks cannot be corrected
Error list: General inputs and outputs	
14	Please acknowledge with ENTER
15	Invalid number of places
16	The entry contains invalid characters
17	Value not within the limits
18	Segment incorrectly programmed
19	Password error
Error list: Profibus job 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 code	Description
Error list: Keyboard and program lock	
26	Keyboard locked
27	Programming locked
28	Write error in the ser. EEprom (Calib)
29	Hardware error: MANUAL + AUTO locked
30	Edit is inadmissible when the program is active
31	Copy is inadmissible when the program is active
32	MANUAL is inadmissible during AUTO lead time
33	Segment change Image update required
34	No DB number, image update by PLC
35	No DB number for process values of PLC
36	Printer loaded or not operational
37	Set point value 1 was not programmed
38	Configure printer (config. / interface)
39	Only possible, when the device is in MANUAL mode
40	Self-optimization already running
41	Tme axis elapsed or not programmed
42	Time axis cannot be copied
43	Time axis not existing
44	Program change is locked
45	MANUAL mode locked
46	Program start 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	Address error (e.g. address does not exist)

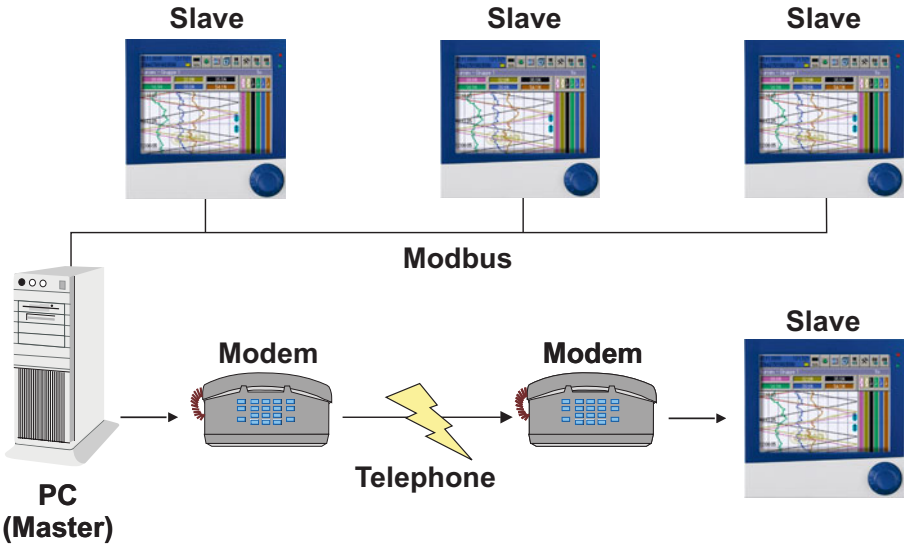
Error code	Description
55	Incorrect or unexpected command
Error list: Event processing	
60	event could not be created
61	event setting failed
62	event clear failed
63	event wait failed
64	event close failed
65	event open failed
66	Sync error between group and data manager
Error list: message processing	
70	Queue memory does not exist
71	Message queue cannot be opened
72	Message pool cannot be generated
73	Memory from message pool cannot be inquired
74	Message cannot be transmitted
Error list: Processing of MQX functions	
80	Task creation failed
81	Hardware-Timer not created
Error list: Flash processing	
90	Data flash write error
Error list: Other errors	
100	Undefined error
101	Division by zero
102	RAM cannot be detected
103	RTC run-time overrun
104	ID does not exist
105	Index too large (overflow)
106	Invalid data
107	Invalid pointer
109	String without 0 characters
110	Timeout during initialization

Error code	Description
111	Value must not be written to
112	Log entry with error bits initiating debug mode
Error list: Email transfer via modem and Ethernet	
120	Step error in the status automatic
121	Invalid response length
122	No CONNECT from the modem
123	FCS checksum incorrect
124	Unexpected value or response
125	Conf-Request not accepted
126	No Conf-Request from the other end
127	No Chap-Request from the other end
128	Response timeout
129	Unknown modem response
130	Unexpected OK by the modem
131	Unexpected CONNECT from the modem
132	Unknown frame received
133	Unexpected PROTOCOL by the modem
134	Unexpected COMPRESS from the modem
135	Invalid PPP package received
136	Unexpected BUSY from the modem
137	Unknown authentification protocol
138	Ignored LCP option
139	Unexpected DELAYED from the modem
140	Unexpected NODIALTONE
141	Unknown PPP protocol
142	Unknown PAP code
143	Ignored IPCP option
144	Ignored IPCP code
145	Unknown CHAP code
146	IP checksum incorrect
147	Unknown IP protocol

Error code	Description
148	Unknown ICMP type
149	Unknown LCP type
150	As a client DNS request received
151	Unknown DNS error
152	DNS response is divided
153	No IP received via DNS
154	Unknown Udp port
155	TCP checksum incorrect
156	TCP port incorrect
157	Unknown TCP-SYN option
158	Unused TCP port
159	Unknown POP3 response
160	Unknown SMTP response
161	Unknown DNS name
162	No MD5 requested from CHAP
163	Authentification error
164	Cancel from other end
165	Error when creating TCP socket
166	Error when binding TCP socket
167	Error on TCP connect
168	Error when transmitting TCP telegram
169	Error when closing TCP socket
170	Error on TCP listing
171	Reset on TCP accept
172	Error on TCP accept
173	SMTP server indicates syntax error
Error list: File system processing	
200	Error when installing the partition manager
201	Error when installing the MFS file system
202	Error when deinstalling the partition manager
203	Error when deinstalling the MFS file system

5.1 Modbus slave

If the paperless recorder is configured as a slave,
see Chapter 3.4 Configuration of the serial interfaces, Page 12,
in the network it responds to Modbus requests of the master.
The master controls the data exchange, the slaves only have a response function. They are identified by their device address.
The master usually is a PC with a setup or visualization program installed. The master can inquire all instrument variables of the paperless recorder (see Chapter 7 Address tables, Page 47).



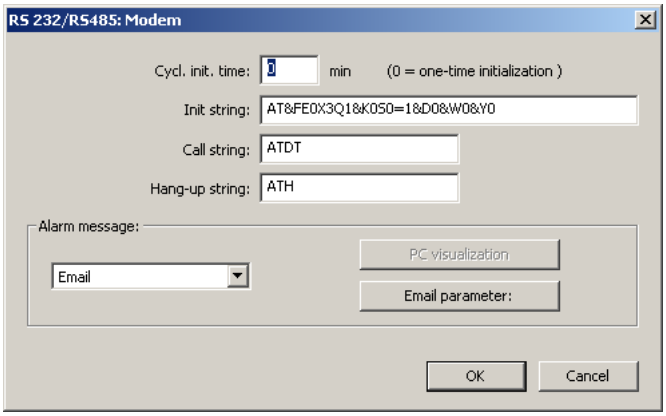
In a Modbus network, only one instrument may be assigned with the master function.

Modem operation

- A paperless recorder in the slave mode can be controlled by the master via a telephone connection using a modem, see above illustration.
- The paperless recorder can initialize a modem (also cyclically, if the modem is switched on after the instrument has been activated).
- The paperless recorder can use the Init string (entry per setup program in the "RS232/RS485: Modem" mask) to configure the modem so that an incoming call is automatically answered. Then a remote inquiry for the paperless recorder can be carried out by the master using Modbus commands or the recorder can send a signal (e.g. an alarm) or an email after the active dial-in.
- The paperless recorder (slave) can use a dial-in/hang-up string to alarm a PC (master) with the suitable Modbus master software (which recognizes incoming modem calls).
- The paperless recorder can use a dial-in/hang-up string to call an internet provider and send an email.

When the modem operation is active, the following parameters (by setup program only) can be edited:

Modem parameters



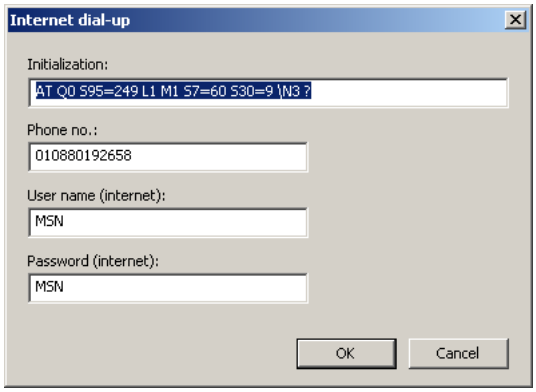
The following Init string is required for the operation as a Modbus slave via modem:

AT&FE0X3Q1&K0S0=1&D0&W0&Y0

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

The dial-in and hang-up string are only required for one of the active modem alarms.

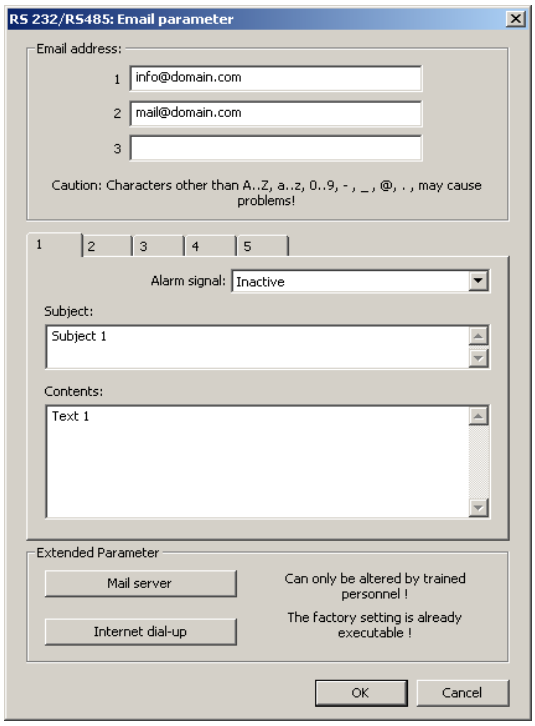
Internet dial-in



For the active internet dial-in for alarms by email, the instrument modem must be switched to a different mode using another INIT string.

Telephone number, user name and password must be entered according to the specifications of the selected internet provider.
Once the internet dial-in has been completed, the modem is automatically reset to its initial state using the Init string entered under modem parameters.

Email parameters

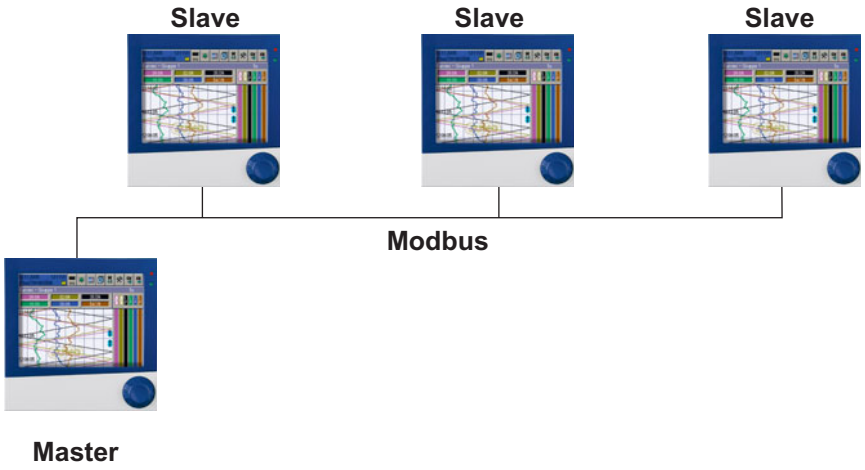


5 Serial protocol types

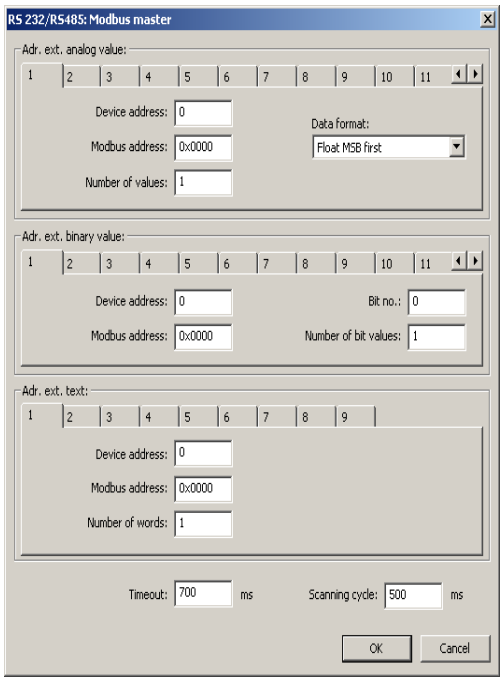
5.2 Modbus master

If the paperless recorder is configured as a master,
(see Chapter 3.4 Configuration of the serial interfaces, Page 12),
it can send Modbus inquiries in the network to slaves (e.g. other paperless recorders). The requested values are written into external instrument variables of the paperless recorder:

- external analog values 1 to 24 in the analog selector
- external binary values 1 to 24 in the binary selector
- external texts 1 to 9 (e.g. to link batch texts in the instrument)



Modbus master



These parameters can be edited in the setpu program and on the paperless recorder.

For each target variable, it is possible to enter the instrument address and Modbus address used to request the value.

Each programmed inquiry can be deactivated by entering the instrument address 0 (if, for example, the "external analog value 2" is no longer to be written to by the Modbus master but by the Profibus).

Double writing of a target variable will lead to undefined states and must be avoided.

For analog and binary values, the entry of a "Number of measured values" or "Number of bits" larger than 1 allows reading consecutively several variables using one command. The following target variables are automatically assigned during saving.

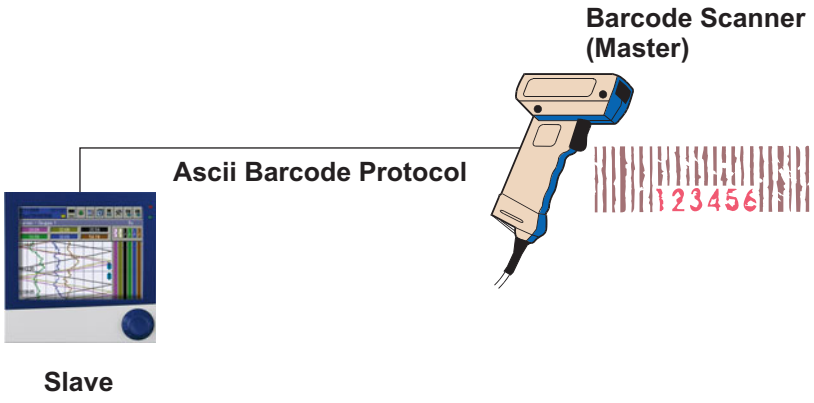
Timeout defines the maximum time-out time used for each sent command to wait for response before the next command is carried out.

Inquiry cycle defines the time interval used to read in variables.

5.3 Barcode

In the "Barcode mode" (see Chapter 3.4 Configuration of the serial interfaces, Page 12), the paperless recorder appears as a slave. It waits for the ASCII strings sent by the barcode scanner (master).
This interface mode only requires the setting of the "Baud rate" and "Data format" configuration parameters, see Chapter 3.4 Configuration of the serial interfaces, Page 12, further parameters are not required here.

The paperless recorder can use these strings for the incremental control of a batch sequence (see 59484).



5 Serial protocol types

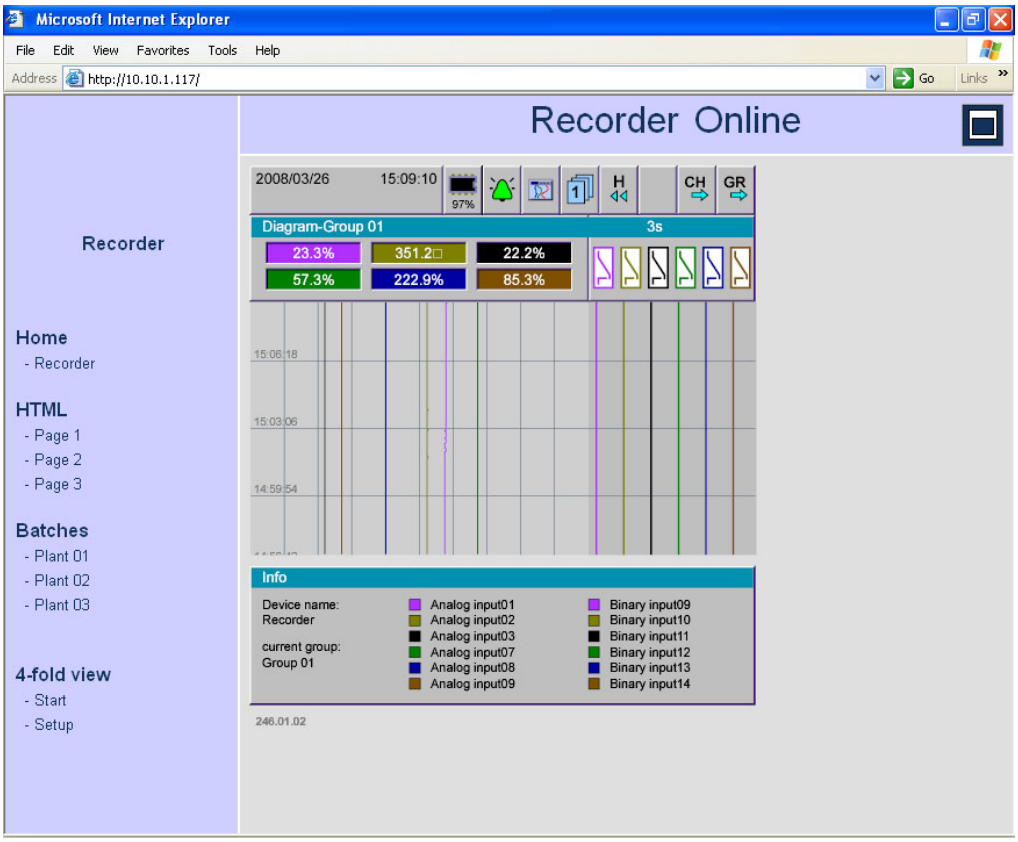
The advantage of the Ethernet connection compared with a serial connection lies in the higher transmission rate the company-wide accessibility.

6.1 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 software, PC evaluation software (PCA) or PCA communication software (PCC).

6.2 Browser connection

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 *Edit* ➔ *Web server* ➔ *Web import*. For this purpose, a memory space of 512 kB is available.

The online visualization as well as three HTML pages and three HTML batch pages are factory-saved as templates.
In the HTML pages, special tags can be used to access device variables. Support is provided in a help window which can be used to select the device variable in the setup program and copy the corresponding HTML tag into the clipboard.

6 Ethernet protocols

6.3 Modbus TCP

In this case, the paperless recorder is configured as a slave and handles incoming requests as a server via port 502. The port can also be changed, see Chapter 3.5 Configuration of the Ethernet interface, Page 13.

Modbus TCP is a standardized process, in which a Modbus telegram is packed into a TCP frame (tunnelled) and transmitted via Ethernet.
The Modbus telegram (without CRC) is transmitted with an additional "MBAP header" of 6 or 7 byte. The seventh byte is identical to the first serial byte, but has a different designation.

Structure of a Modbus TCP telegram

MBAP header				Modbus telegram
2 byte Trans-action ID	2 byte Protocol ID	2 byte Length	1 byte Unit ID	Further bytes as specified below, however, without CRC
Identical in request and response	Must be 0 for Modbus	Length of question and response in byte from (incl.) "Unit ID"	Corresponds to the controller address. For TCP must be 0xFF or 0 (0=broadcast)	

For comparison: the "normal" Modbus telegram, see Chapter 4.4 Structure of a Modbus telegram, Page 17:

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

This protocol can be used, e.g. by a suitable process data visualization program to read and write values of the paperless recorder via a company-wide Ethernet network. All device variables from the Modbus address tables (see Chapter 7 Address tables, Page 47) can be accessed.

Only one Modbus master (client) can access a paperless recorder via Modbus TCP at a time.

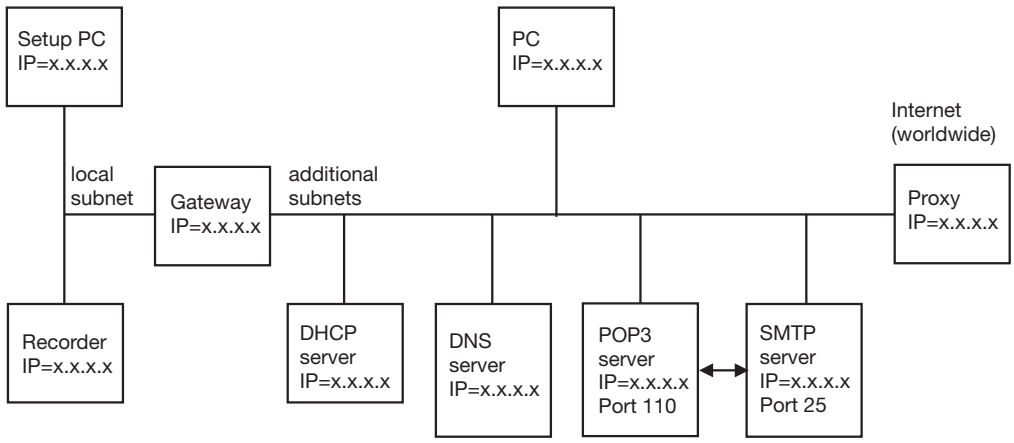
A connection opened by a client is cancelled by the paperless recorder after 30 seconds of inactivity.

If a Modbus TCP port is closed (by the paperless recorder or the other end), it can be reopened after 10 seconds.

6.4 Email (SMTP and POP3)

The paperless recorder can transmit Emails (e.g. alarms). In this case, it is the master (client) and can access SMTP servers at the standard port (25) as well as POP3 servers at the standard port (110)

Typical networking in the company network



Function of the individual stations

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

DHCP server:
can automatically assign IP address, sub-net mask and gateway address to other stations when switching on. These parameters can also be entered manually, then a DHCP server is not required.

DNS server:
converts symbolic names to IP addresses, e.g. question: "www.name.de" will generate the "www.name.de has IP=10.12.32.45" response.

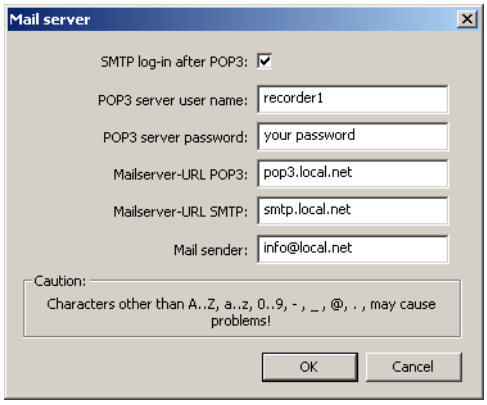
POP3 server:
serves to read out received Emails of a mail account. The POP3 mail account can be accessed after log-in entering user name and password. A successful log-in process often releases the transmission authorization of a connected SMTP server.

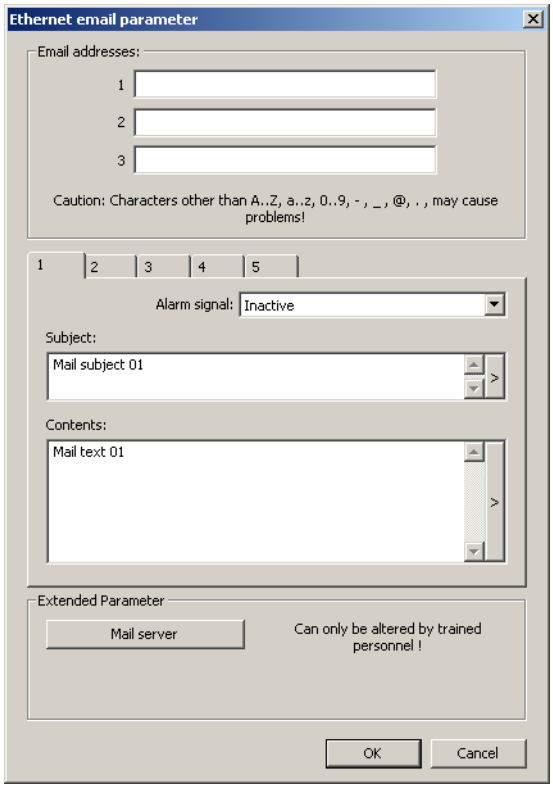
SMTP server:
serves to transmit Emails. The authorization to transmit Emails via a mail account must be released in several networks by previously logging in at the corresponding POP3 server.

Proxy:
serves as a gateway between the local company network and the worldwide internet. It is also used for the conversion of "local" IP addresses (used in the company network) to "once-only" IP addresses (used in the internet).
The device software cannot address a proxy!

Parameters for mail server and Email parameters

These parameters can only be edited via the setup program.
Edit ➔ *Ethernet Email parameters*

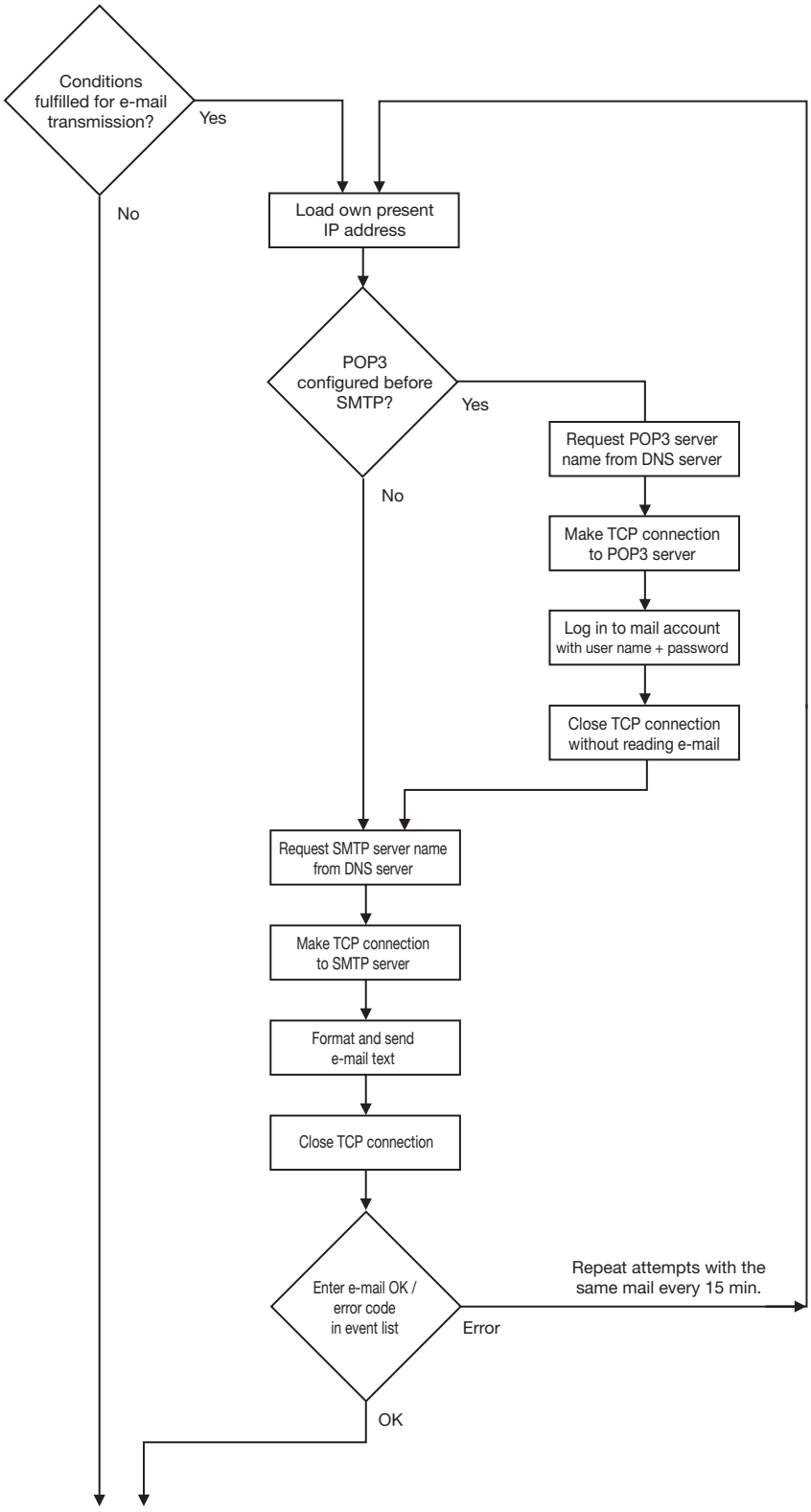




☞ A mail server within the company network (not the internet) must be entered here. This mail server should be able to also transmit Emails to the internet.

Transmission of an Email via internet

Here, serveral steps depend on configured device parameters. An error code of the event entry (see Chapter 4.9.3 Error codes as integer return values, Page 30 (particularly the error codes 120 to 173)) can suggest an incorrectly set parameter. An incorrectly entered DNS server IP, for example, generates the error code 153 = "no IP received via DNS".



6 Ethernet protocols

7.1 Data types and type of access

The Chapter 7.2 contain descriptions of all process values (variables) including their addresses, data type and type of access.

Meaning:

R	Read only access
W	Write access
R/W	Read and write access
char xx	Character string consisting of ASCII characters (8 bits) with the length xx; xx = Length including the "\0" character string end character 2 characters are read and written in a word.
word	Unsigned Integer 16 bit (1 word)
int	Signed Integer 16 bit (1 word)
long	Signed Integer 32 bit (2 words)
float	Float value (4 byte = 2 words) as per IEEE 754
bool	Bool can be read and written as word, its value range is 0 … 1
byte	Byte (8 bit) can be read and written as word, its value range is 0 … 255
Bit x	Bit No. x (bit 0 is always the bit with the lowest value)

7.2 Modbus addresses of important device and process data

Address	Access	Data type	Signal designation
0x0009	R	char 12	Software version
0x0103	R	word	Status binary inputs / outputs 1...16
0x0104	R	word	Status binary inputs / outputs 17...24
0x0105	R	word	Status ext. binary inputs / outputs 1...16 (via PROFIBUS-DP also R/W)
0x0106	R	word	Status ext. binary inputs / outputs 17...24 (via PROFIBUS-DP also R/W)
0x0107	R	word	Status Relay 1…7
0x0108	R	word	Math bool 1…16
0x0109	R	word	Math bool 17…18
0x010A	R	word	Logic result 1…16
0x010B	R	word	Logic result 17…18
0x010C	R	word	Limit value monitoring Status 1...9
0x010D	R	word	Analog alarms 1 channel 1...16
0x010E	R	word	Analog alarms 1 channel 17...18
0x010F	R	word	Analog alarms 2 channel 1...16

0x0110	R	word	Analog alarms 2 channel 17...18
0x0111	R	word	External analog alarms 1 channel 1...16
0x0112	R	word	External analog alarms 1 channel 17...18
0x0113	R	word	External analog alarms 2 channel 1...16
0x0114	R	word	External analog alarms 2 channel 17...18
0x0115	R	word	Binary inputs / outputs alarm 1...16
0x0116	R	word	Binary inputs / outputs alarm 17...24
0x0117	R	word	External binary inputs alarm 1...16
0x0118	R	word	External binary inputs alarm 17...24
0x0119	R	word	Math alarm 1 1…16
0x011A	R	word	Math alarm 1 17…18
0x011B	R	word	Math alarm 2 1…16
0x011C	R	word	Math alarm 2 17…18
0x011D	R	word	Logic alarm 1…16
0x011E	R	word	Logic alarm 17…18
0x011F	R	word	Limit value monitoring alarm 1...9
0x0120	R	word	Counter Alarm 1 1…16
0x0121	R	word	Counter Alarm 1 17…27
0x0122	R	word	Counter Alarm 2 1…16
0x0123	R	word	Counter Alarm 2 17…27
0x0124	R	word	Group alarms 1...9
0x0125	R	word	Alarm pos. tolerance band group 1...9
0x0126	R	word	Alarm neg. tolerance band group 1...9
0x0127	R	word	Multi-input alarms
0x1000	R	char 46	Device name
0x1017	R/W	byte	Brightness of the display
0x120F	R	long	Highspeed counter B1 (HW counter B1) The normal counters 1 … 27 cannot be read out via Modbus but only via the highspeed counters.
0x1211	R	long	Highspeed counter B2 (HW counter B2)
0x1213	R	long	Highspeed counter B9 (HW counter B9)
0x1215	R	long	Highspeed counter B10 (HW counter B10)
0x1217	R	long	Highspeed counter B17 (HW counter B17)
0x1219	R	long	Highspeed counter B18 (HW counter B18)
0x121B	R	char 11	Software version card 1
0x1221	R	char 11	Software version card 2
0x1227	R	char 11	Software version card 3
0x122D	R	char 11	Software version Profibus

Address	Access	Data type	Signal designation
0x1257	R	float	Analog process value filtered 1
0x1259	R	float	Analog process value filtered 2
0x125B	R	float	Analog process value filtered 3
0x125D	R	float	Analog process value filtered 4
0x125F	R	float	Analog process value filtered 5
0x1261	R	float	Analog process value filtered 6

Address	Access	Data type	Signal designation
0x1263	R	float	Analog process value filtered 7
0x1265	R	float	Analog process value filtered 8
0x1267	R	float	Analog process value filtered 9
0x1269	R	float	Analog process value filtered 10
0x126B	R	float	Analog process value filtered 11
0x126D	R	float	Analog process value filtered 12
0x126F	R	float	Analog process value filtered 13
0x1271	R	float	Analog process value filtered 14
0x1273	R	float	Analog process value filtered 15
0x1275	R	float	Analog process value filtered 16
0x1277	R	float	Analog process value filtered 17
0x1279	R	float	Analog process value filtered 18
0x127B	R	bool	Analog alarms 1, channel 1
0x127C	R	bool	Analog alarms 1, channel 2
0x127D	R	bool	Analog alarms 1, channel 3
0x127E	R	bool	Analog alarms 1, channel 4
0x127F	R	bool	Analog alarms 1, channel 5
0x1280	R	bool	Analog alarms 1, channel 6
0x1281	R	bool	Analog alarms 1, channel 7
0x1282	R	bool	Analog alarms 1, channel 8
0x1283	R	bool	Analog alarms 1, channel 9
0x1284	R	bool	Analog alarms 1, channel 10
0x1285	R	bool	Analog alarms 1, channel 11
0x1286	R	bool	Analog alarms 1, channel 12
0x1287	R	bool	Analog alarms 1, channel 13
0x1288	R	bool	Analog alarms 1, channel 14
0x1289	R	bool	Analog alarms 1, channel 15
0x128A	R	bool	Analog alarms 1, channel 16
0x128B	R	bool	Analog alarms 1, channel 17
0x128C	R	bool	Analog alarms 1, channel 18
0x128D	R	bool	Analog alarms 2, channel 1
0x128E	R	bool	Analog alarms 2, channel 2
0x128F	R	bool	Analog alarms 2, channel 3
0x1290	R	bool	Analog alarms 2, channel 4
0x1291	R	bool	Analog alarms 2, channel 5
0x1292	R	bool	Analog alarms 2, channel 6
0x1293	R	bool	Analog alarms 2, channel 7
0x1294	R	bool	Analog alarms 2, channel 8
0x1295	R	bool	Analog alarms 2, channel 9
0x1296	R	bool	Analog alarms 2, channel 10
0x1297	R	bool	Analog alarms 2, channel 11
0x1298	R	bool	Analog alarms 2, channel 12
0x1299	R	bool	Analog alarms 2, channel 13
0x129A	R	bool	Analog alarms 2, channel 14
0x129B	R	bool	Analog alarms 2, channel 15
0x129C	R	bool	Analog alarms 2, channel 16

Address	Access	Data type	Signal designation
0x129D	R	bool	Analog alarms 2, channel 17
0x129E	R	bool	Analog alarms 2, channel 18
0x12AA	R	float	External analog process value, limit-checked 1
0x12AC	R	float	External analog process value, limit-checked 2
0x12AE	R	float	External analog process value, limit-checked 3
0x12B0	R	float	External analog process value, limit-checked 4
0x12B2	R	float	External analog process value, limit-checked 5
0x12B4	R	float	External analog process value, limit-checked 6
0x12B6	R	float	External analog process value, limit-checked 7
0x12B8	R	float	External analog process value, limit-checked 8
0x12BA	R	float	External analog process value, limit-checked 9
0x12BC	R	float	External analog process value, limit-checked 10
0x12BE	R	float	External analog process value, limit-checked 11
0x12C0	R	float	External analog process value, limit-checked 12
0x12C2	R	float	External analog process value, limit-checked 13
0x12C4	R	float	External analog process value, limit-checked 14
0x12C6	R	float	External analog process value, limit-checked 15
0x12C8	R	float	External analog process value, limit-checked 16
0x12CA	R	float	External analog process value, limit-checked 17
0x12CC	R	float	External analog process value, limit-checked 18
0x12CE	R	float	External analog process value, limit-checked 19
0x12D0	R	float	External analog process value, limit-checked 20
0x12D2	R	float	External analog process value, limit-checked 21
0x12D4	R	float	External analog process value, limit-checked 22
0x12D6	R	float	External analog process value, limit-checked 23

Address	Access	Data type	Signal designation
0x12D8	R	float	External analog process value, limit-checked 24
0x12DA	R/W	float	External analog process value, interface 1
0x12DC	R/W	float	External analog process value, interface 2
0x12DE	R/W	float	External analog process value, interface 3
0x12E0	R/W	float	External analog process value, interface 4
0x12E2	R/W	float	External analog process value, interface 5
0x12E4	R/W	float	External analog process value, interface 6
0x12E6	R/W	float	External analog process value, interface 7
0x12E8	R/W	float	External analog process value, interface 8
0x12EA	R/W	float	External analog process value, interface 9
0x12EC	R/W	float	External analog process value, interface 10
0x12EE	R/W	float	External analog process value, interface 11
0x12F0	R/W	float	External analog process value, interface 12
0x12F2	R/W	float	External analog process value, interface 13
0x12F4	R/W	float	External analog process value, interface 14
0x12F6	R/W	float	External analog process value, interface 15
0x12F8	R/W	float	External analog process value, interface 16
0x12FA	R/W	float	External analog process value, interface 17
0x12FC	R/W	float	External analog process value, interface 18
0x12FE	R/W	float	External analog process value, interface 19
0x1300	R/W	float	External analog process value, interface 20
0x1302	R/W	float	External analog process value, interface 21
0x1304	R/W	float	External analog process value, interface 22
0x1306	R/W	float	External analog process value, interface 23
0x1308	R/W	float	External analog process value, interface 24
0x130A	R	bool	External analog alarms 1, channel 1
0x130B	R	bool	External analog alarms 1, channel 2
0x130C	R	bool	External analog alarms 1, channel 3
0x130D	R	bool	External analog alarms 1, channel 4
0x130E	R	bool	External analog alarms 1, channel 5
0x130F	R	bool	External analog alarms 1, channel 6
0x1310	R	bool	External analog alarms 1, channel 7
0x1311	R	bool	External analog alarms 1, channel 8
0x1312	R	bool	External analog alarms 1, channel 9
0x1313	R	bool	External analog alarms 1, channel 10
0x1314	R	bool	External analog alarms 1, channel 11
0x1315	R	bool	External analog alarms 1, channel 12
0x1316	R	bool	External analog alarms 1, channel 13
0x1317	R	bool	External analog alarms 1, channel 14
0x1318	R	bool	External analog alarms 1, channel 15
0x1319	R	bool	External analog alarms 1, channel 16
0x131A	R	bool	External analog alarms 1, channel 17
0x131B	R	bool	External analog alarms 1, channel 18
0x131C	R	bool	External analog alarms 1, channel 19
0x131D	R	bool	External analog alarms 1, channel 20

Address	Access	Data type	Signal designation
0x131E	R	bool	External analog alarms 1, channel 21
0x131F	R	bool	External analog alarms 1, channel 22
0x1320	R	bool	External analog alarms 1, channel 23
0x1321	R	bool	External analog alarms 1, channel 24
0x1322	R	bool	External analog alarms 2, channel 1
0x1323	R	bool	External analog alarms 2, channel 2
0x1324	R	bool	External analog alarms 2, channel 3
0x1325	R	bool	External analog alarms 2, channel 4
0x1326	R	bool	External analog alarms 2, channel 5
0x1327	R	bool	External analog alarms 2, channel 6
0x1328	R	bool	External analog alarms 2, channel 7
0x1329	R	bool	External analog alarms 2, channel 8
0x132A	R	bool	External analog alarms 2, channel 9
0x132B	R	bool	External analog alarms 2, channel 10
0x132C	R	bool	External analog alarms 2, channel 11
0x132D	R	bool	External analog alarms 2, channel 12
0x132E	R	bool	External analog alarms 2, channel 13
0x132F	R	bool	External analog alarms 2, channel 14
0x1330	R	bool	External analog alarms 2, channel 15
0x1331	R	bool	External analog alarms 2, channel 16
0x1332	R	bool	External analog alarms 2, channel 17
0x1333	R	bool	External analog alarms 2, channel 18
0x1334	R	bool	External analog alarms 2, channel 19
0x1335	R	bool	External analog alarms 2, channel 20
0x1336	R	bool	External analog alarms 2, channel 21
0x1337	R	bool	External analog alarms 2, channel 22
0x1338	R	bool	External analog alarms 2, channel 23
0x1339	R	bool	External analog alarms 2, channel 24
0x133A	R	bool	Binary input / output status 1
0x133B	R	bool	Binary input / output status 2
0x133C	R	bool	Binary input / output status 3
0x133D	R	bool	Binary input / output status 4
0x133E	R	bool	Binary input / output status 5
0x133F	R	bool	Binary input / output status 6
0x1340	R	bool	Binary input / output status 7
0x1341	R	bool	Binary input / output status 8
0x1342	R	bool	Binary input / output status 9
0x1343	R	bool	Binary input / output status 10
0x1344	R	bool	Binary input / output status 11
0x1345	R	bool	Binary input / output status 12
0x1346	R	bool	Binary input / output status 13
0x1347	R	bool	Binary input / output status 14
0x1348	R	bool	Binary input / output status 15
0x1349	R	bool	Binary input / output status 16
0x134A	R	bool	Binary input / output status 17
0x134B	R	bool	Binary input / output status 18

Address	Access	Data type	Signal designation
0x134C	R	bool	Binary input / output status 19
0x134D	R	bool	Binary input / output status 20
0x134E	R	bool	Binary input / output status 21
0x134F	R	bool	Binary input / output status 22
0x1350	R	bool	Binary input / output status 23
0x1351	R	bool	Binary input / output status 24
0x1352	R	bool	Binary input / output alarm 1
0x1353	R	bool	Binary input / output alarm 2
0x1354	R	bool	Binary input / output alarm 3
0x1355	R	bool	Binary input / output alarm 4
0x1356	R	bool	Binary input / output alarm 5
0x1357	R	bool	Binary input / output alarm 6
0x1358	R	bool	Binary input / output alarm 7
0x1359	R	bool	Binary input / output alarm 8
0x135A	R	bool	Binary input / output alarm 9
0x135B	R	bool	Binary input / output alarm 10
0x135C	R	bool	Binary input / output alarm 11
0x135D	R	bool	Binary input / output alarm 12
0x135E	R	bool	Binary input / output alarm 13
0x135F	R	bool	Binary input / output alarm 14
0x1360	R	bool	Binary input / output alarm 15
0x1361	R	bool	Binary input / output alarm 16
0x1362	R	bool	Binary input / output alarm 17
0x1363	R	bool	Binary input / output alarm 18
0x1364	R	bool	Binary input / output alarm 19
0x1365	R	bool	Binary input / output alarm 20
0x1366	R	bool	Binary input / output alarm 21
0x1367	R	bool	Binary input / output alarm 22
0x1368	R	bool	Binary input / output alarm 23
0x1369	R	bool	Binary input / output alarm 24
0x136A	R/W	char 64	External batch text 1
0x138A	R/W	char 64	External batch text 2
0x13AA	R/W	char 64	External batch text 3
0x13CA	R/W	char 64	External batch text 4
0x13EA	R/W	char 64	External batch text 5
0x140A	R/W	char 64	External batch text 6
0x142A	R/W	char 64	External batch text 7
0x144A	R/W	char 64	External batch text 8
0x146A	R/W	char 64	External batch text 9
0x148A	R/W	char 94	External event text group 1
0x14B9	R/W	char 94	External event text group 2
0x14E8	R/W	char 94	External event text group 3
0x1517	R/W	char 94	External event text group 4
0x1546	R/W	char 94	External event text group 5
0x1575	R/W	char 94	External event text group 6
0x15A4	R/W	char 94	External event text group 7

Address	Access	Data type	Signal designation
0x15D3	R/W	char 94	External event text group 8
0x1602	R/W	char 94	External event text group 9
0x1631	R	bool	Relay output 1
0x1632	R	bool	Relay output 2
0x1633	R	bool	Relay output 3
0x1634	R	bool	Relay output 4
0x1635	R	bool	Relay output 5
0x1636	R	bool	Relay output 6
0x1637	R	bool	Relay output 7
0x1638	R/W	bool	External binary input, status 1
0x1639	R/W	bool	External binary input, status 2
0x163A	R/W	bool	External binary input, status 3
0x163B	R/W	bool	External binary input, status 4
0x163C	R/W	bool	External binary input, status 5
0x163D	R/W	bool	External binary input, status 6
0x163E	R/W	bool	External binary input, status 7
0x163F	R/W	bool	External binary input, status 8
0x1640	R/W	bool	External binary input, status 9
0x1641	R/W	bool	External binary input, status 10
0x1642	R/W	bool	External binary input, status 11
0x1643	R/W	bool	External binary input, status 12
0x1644	R/W	bool	External binary input, status 13
0x1645	R/W	bool	External binary input, status 14
0x1646	R/W	bool	External binary input, status 15
0x1647	R/W	bool	External binary input, status 16
0x1648	R/W	bool	External binary input, status 17
0x1649	R/W	bool	External binary input, status 18
0x164A	R/W	bool	External binary input, status 19
0x164B	R/W	bool	External binary input, status 20
0x164C	R/W	bool	External binary input, status 21
0x164D	R/W	bool	External binary input, status 22
0x164E	R/W	bool	External binary input, status 23
0x164F	R/W	bool	External binary input, status 24
0x1650	R	bool	External binary input, alarm 1
0x1651	R	bool	External binary input, alarm 2
0x1652	R	bool	External binary input, alarm 3
0x1653	R	bool	External binary input, alarm 4
0x1654	R	bool	External binary input, alarm 5
0x1655	R	bool	External binary input, alarm 6
0x1656	R	bool	External binary input, alarm 7
0x1657	R	bool	External binary input, alarm 8
0x1658	R	bool	External binary input, alarm 9
0x1659	R	bool	External binary input, alarm 10
0x165A	R	bool	External binary input, alarm 11
0x165B	R	bool	External binary input, alarm 12
0x165C	R	bool	External binary input, alarm 13

Address	Access	Data type	Signal designation
0x165D	R	bool	External binary input, alarm 14
0x165E	R	bool	External binary input, alarm 15
0x165F	R	bool	External binary input, alarm 16
0x1660	R	bool	External binary input, alarm 17
0x1661	R	bool	External binary input, alarm 18
0x1662	R	bool	External binary input, alarm 19
0x1663	R	bool	External binary input, alarm 20
0x1664	R	bool	External binary input, alarm 21
0x1665	R	bool	External binary input, alarm 22
0x1666	R	bool	External binary input, alarm 23
0x1667	R	bool	External binary input, alarm 24
0x1668	R	float	Math result 1
0x166A	R	float	Math result 2
0x166C	R	float	Math result 3
0x166E	R	float	Math result 4
0x1670	R	float	Math result 5
0x1672	R	float	Math result 6
0x1674	R	float	Math result 7
0x1676	R	float	Math result 8
0x1678	R	float	Math result 9
0x167A	R	bool	Math alarm 1, 1
0x167B	R	bool	Math alarm 1, 2
0x167C	R	bool	Math alarm 1, 3
0x167D	R	bool	Math alarm 1, 4
0x167E	R	bool	Math alarm 1, 5
0x167F	R	bool	Math alarm 1, 6
0x1680	R	bool	Math alarm 1, 7
0x1681	R	bool	Math alarm 1, 8
0x1682	R	bool	Math alarm 1, 9
0x1683	R	bool	Math alarm 2, 1
0x1684	R	bool	Math alarm 2, 2
0x1685	R	bool	Math alarm 2, 3
0x1686	R	bool	Math alarm 2, 4
0x1687	R	bool	Math alarm 2, 5
0x1688	R	bool	Math alarm 2, 6
0x1689	R	bool	Math alarm 2, 7
0x168A	R	bool	Math alarm 2, 8
0x168B	R	bool	Math alarm 2, 9
0x168C	R	bool	Logic result 1
0x168D	R	bool	Logic result 2
0x168E	R	bool	Logic result 3
0x168F	R	bool	Logic result 4
0x1690	R	bool	Logic result 5
0x1691	R	bool	Logic result 6
0x1692	R	bool	Logic result 7
0x1693	R	bool	Logic result 8

Address	Access	Data type	Signal designation
0x1694	R	bool	Logic result 9
0x1695	R	bool	Logic alarm 1
0x1696	R	bool	Logic alarm 2
0x1697	R	bool	Logic alarm 3
0x1698	R	bool	Logic alarm 4
0x1699	R	bool	Logic alarm 5
0x169A	R	bool	Logic alarm 6
0x169B	R	bool	Logic alarm 7
0x169C	R	bool	Logic alarm 8
0x169D	R	bool	Logic alarm 9
0x169E	R	bool	Limit value monitoring status 1
0x169F	R	bool	Limit value monitoring status 2
0x16A0	R	bool	Limit value monitoring status 3
0x16A1	R	bool	Limit value monitoring status 4
0x16A2	R	bool	Limit value monitoring status 5
0x16A3	R	bool	Limit value monitoring status 6
0x16A4	R	bool	Limit value monitoring status 7
0x16A5	R	bool	Limit value monitoring status 8
0x16A6	R	bool	Limit value monitoring status 9
0x16A7	R	bool	Limit value monitoring alarm 1
0x16A8	R	bool	Limit value monitoring alarm 2
0x16A9	R	bool	Limit value monitoring alarm 3
0x16AA	R	bool	Limit value monitoring alarm 4
0x16AB	R	bool	Limit value monitoring alarm 5
0x16AC	R	bool	Limit value monitoring alarm 6
0x16AD	R	bool	Limit value monitoring alarm 7
0x16AE	R	bool	Limit value monitoring alarm 8
0x16AF	R	bool	Limit value monitoring alarm 9
0x16B0	R	bool	Counter alarm 1, 1
0x16B1	R	bool	Counter alarm 1, 2
0x16B2	R	bool	Counter alarm 1, 3
0x16B3	R	bool	Counter alarm 1, 4
0x16B4	R	bool	Counter alarm 1, 5
0x16B5	R	bool	Counter alarm 1, 6
0x16B6	R	bool	Counter alarm 1, 7
0x16B7	R	bool	Counter alarm 1, 8
0x16B8	R	bool	Counter alarm 1, 9
0x16B9	R	bool	Counter alarm 1, 10
0x16BA	R	bool	Counter alarm 1, 11
0x16BB	R	bool	Counter alarm 1, 12
0x16BC	R	bool	Counter alarm 1, 13
0x16BD	R	bool	Counter alarm 1, 14
0x16BE	R	bool	Counter alarm 1, 15
0x16BF	R	bool	Counter alarm 1, 16
0x16C0	R	bool	Counter alarm 1, 17
0x16C1	R	bool	Counter alarm 1, 18

Address	Access	Data type	Signal designation
0x16C2	R	bool	Counter alarm 1, 19
0x16C3	R	bool	Counter alarm 1, 20
0x16C4	R	bool	Counter alarm 1, 21
0x16C5	R	bool	Counter alarm 1, 22
0x16C6	R	bool	Counter alarm 1, 23
0x16C7	R	bool	Counter alarm 1, 24
0x16C8	R	bool	Counter alarm 1, 25
0x16C9	R	bool	Counter alarm 1, 26
0x16CA	R	bool	Counter alarm 1, 27
0x16CB	R	bool	Counter alarm 2, 1
0x16CC	R	bool	Counter alarm 2, 2
0x16CD	R	bool	Counter alarm 2, 3
0x16CE	R	bool	Counter alarm 2, 4
0x16CF	R	bool	Counter alarm 2, 5
0x16D0	R	bool	Counter alarm 2, 6
0x16D1	R	bool	Counter alarm 2, 7
0x16D2	R	bool	Counter alarm 2, 8
0x16D3	R	bool	Counter alarm 2, 9
0x16D4	R	bool	Counter alarm 2, 10
0x16D5	R	bool	Counter alarm 2, 11
0x16D6	R	bool	Counter alarm 2, 12
0x16D7	R	bool	Counter alarm 2, 13
0x16D8	R	bool	Counter alarm 2, 14
0x16D9	R	bool	Counter alarm 2, 15
0x16DA	R	bool	Counter alarm 2, 16
0x16DB	R	bool	Counter alarm 2, 17
0x16DC	R	bool	Counter alarm 2, 18
0x16DD	R	bool	Counter alarm 2, 19
0x16DE	R	bool	Counter alarm 2, 20
0x16DF	R	bool	Counter alarm 2, 21
0x16E0	R	bool	Counter alarm 2, 22
0x16E1	R	bool	Counter alarm 2, 23
0x16E2	R	bool	Counter alarm 2, 24
0x16E3	R	bool	Counter alarm 2, 25
0x16E4	R	bool	Counter alarm 2, 26
0x16E5	R	bool	Counter alarm 2, 27
0x17B0	R	bool	Device alarm
0x17B1	R	byte	Ethernet IP address 1st byte
0x17B2	R	byte	Ethernet IP address 2nd byte
0x17B3	R	byte	Ethernet IP address 3rd byte
0x17B4	R	byte	Ethernet IP address 4th byte
0x17B5	R/W	bool	Profibus commissioning/start-up
0x17B6	R	char 64	current batch text 1 - machine 1
0x17D6	R	char 64	current batch text 2 - machine 1
0x17F6	R	char 64	current batch text 3 - machine 1

Address	Access	Data type	Signal designation
0x1816	R	char 64	current batch text 4 - machine 1
0x1836	R	char 64	current batch text 5 - machine 1
0x1856	R	char 64	current batch text 6 - machine 1
0x1876	R	char 64	current batch text 7 - machine 1
0x1896	R	char 64	current batch text 8 - machine 1
0x18B6	R	char 64	current batch text 9 - machine 1
0x18D6	R	char 64	current batch text 10 - machine 1
0x18F6	R	char 64	current batch text 1 - machine 2
0x1916	R	char 64	current batch text 2 - machine 2
0x1936	R	char 64	current batch text 3 - machine 2
0x1956	R	char 64	current batch text 4 - machine 2
0x1976	R	char 64	current batch text 5 - machine 2
0x1996	R	char 64	current batch text 6 - machine 2
0x19B6	R	char 64	current batch text 7 - machine 2
0x19D6	R	char 64	current batch text 8 - machine 2
0x19F6	R	char 64	current batch text 9 - machine 2
0x1A16	R	char 64	current batch text 10 - machine 2
0x1A36	R	char 64	current batch text 1 - machine 3
0x1A56	R	char 64	current batch text 2 - machine 3
0x1A76	R	char 64	current batch text 3 - machine 3
0x1A96	R	char 64	current batch text 4 - machine 3
0x1AB6	R	char 64	current batch text 5 - machine 3
0x1AD6	R	char 64	current batch text 6 - machine 3
0x1AF6	R	char 64	current batch text 7 - machine 3
0x1B16	R	char 64	current batch text 8 - machine 3
0x1B36	R	char 64	current batch text 9 - machine 3
0x1B56	R	char 64	current batch text 10 - machine 3
0x1B76	R	float	Math result 10
0x1B78	R	float	Math result 11
0x1B7A	R	float	Math result 12
0x1B7C	R	float	Math result 13
0x1B7E	R	float	Math result 14
0x1B80	R	float	Math result 15
0x1B82	R	float	Math result 16
0x1B84	R	float	Math result 17
0x1B86	R	float	Math result 18
0x1B88	R	bool	Math alarm 1, 10
0x1B89	R	bool	Math alarm 1, 11
0x1B8A	R	bool	Math alarm 1, 12
0x1B8B	R	bool	Math alarm 1, 13
0x1B8C	R	bool	Math alarm 1, 14
0x1B8D	R	bool	Math alarm 1, 15
0x1B8E	R	bool	Math alarm 1, 16
0x1B8F	R	bool	Math alarm 1, 17
0x1B90	R	bool	Math alarm 1, 18
0x1B91	R	bool	Math alarm 2, 10

Address	Access	Data type	Signal designation
0x1B92	R	bool	Math alarm 2, 11
0x1B93	R	bool	Math alarm 2, 12
0x1B94	R	bool	Math alarm 2, 13
0x1B95	R	bool	Math alarm 2, 14
0x1B96	R	bool	Math alarm 2, 15
0x1B97	R	bool	Math alarm 2, 16
0x1B98	R	bool	Math alarm 2, 17
0x1B99	R	bool	Math alarm 2, 18
0x1B9A	R	bool	Logic result 10
0x1B9B	R	bool	Logic result 11
0x1B9C	R	bool	Logic result 12
0x1B9D	R	bool	Logic result 13
0x1B9E	R	bool	Logic result 14
0x1B9F	R	bool	Logic result 15
0x1BA0	R	bool	Logic result 16
0x1BA1	R	bool	Logic result 17
0x1BA2	R	bool	Logic result 18
0x1BA3	R	bool	Logic alarm 10
0x1BA4	R	bool	Logic alarm 11
0x1BA5	R	bool	Logic alarm 12
0x1BA6	R	bool	Logic alarm 13
0x1BA7	R	bool	Logic alarm 14
0x1BA8	R	bool	Logic alarm 15
0x1BA9	R	bool	Logic alarm 16
0x1BAA	R	bool	Logic alarm 17
0x1BAB	R	bool	Logic alarm 18
0x1BAC	R	bool	Math bool result 1
0x1BAD	R	bool	Math bool result 2
0x1BAE	R	bool	Math bool result 3
0x1BAF	R	bool	Math bool result 4
0x1BB0	R	bool	Math bool result 5
0x1BB1	R	bool	Math bool result 6
0x1BB2	R	bool	Math bool result 7
0x1BB3	R	bool	Math bool result 8
0x1BB4	R	bool	Math bool result 9
0x1BB5	R	bool	Math bool result 10
0x1BB6	R	bool	Math bool result 11
0x1BB7	R	bool	Math bool result 12
0x1BB8	R	bool	Math bool result 13
0x1BB9	R	bool	Math bool result 14
0x1BBA	R	bool	Math bool result 15
0x1BBB	R	bool	Math bool result 16
0x1BBC	R	bool	Math bool result 17
0x1BBD	R	bool	Math bool result 18
0x1BBE	R	bool	Group alarm, group 1
0x1BBF	R	bool	Group alarm, group 2

Address	Access	Data type	Signal designation
0x1BC0	R	bool	Group alarm, group 3
0x1BC1	R	bool	Group alarm, group 4
0x1BC2	R	bool	Group alarm, group 5
0x1BC3	R	bool	Group alarm, group 6
0x1BC4	R	bool	Group alarm, group 7
0x1BC5	R	bool	Group alarm, group 8
0x1BC6	R	bool	Group alarm, group 9
0x1BC7	R	bool	Alarm positive tolerance band, group 1
0x1BC8	R	bool	Alarm positive tolerance band, group 2
0x1BC9	R	bool	Alarm positive tolerance band, group 3
0x1BCA	R	bool	Alarm positive tolerance band, group 4
0x1BCB	R	bool	Alarm positive tolerance band, group 5
0x1BCC	R	bool	Alarm positive tolerance band, group 6
0x1BCD	R	bool	Alarm positive tolerance band, group 7
0x1BCE	R	bool	Alarm positive tolerance band, group 8
0x1BCF	R	bool	Alarm positive tolerance band, group 9
0x1BD0	R	bool	Alarm negative tolerance band, group 1
0x1BD1	R	bool	Alarm negative tolerance band, group 2
0x1BD2	R	bool	Alarm negative tolerance band, group 3
0x1BD3	R	bool	Alarm negative tolerance band, group 4
0x1BD4	R	bool	Alarm negative tolerance band, group 5
0x1BD5	R	bool	Alarm negative tolerance band, group 6
0x1BD6	R	bool	Alarm negative tolerance band, group 7
0x1BD7	R	bool	Alarm negative tolerance band, group 8
0x1BD8	R	bool	Alarm negative tolerance band, group 9
0x1BD9	R	bool	Read out memory alarm CF
0x1BDA	R	bool	Alarm, CF card full
0x1BDB	R	bool	Memory alarm, interface
0x1BDC	R	bool	Login
0x1BDD	R	bool	Malfunction
0x1BDE	R	bool	Reserve 1
0x1BDF	R	bool	Fieldbus error
0x1BE0	R	bool	Reserve 2
0x1BE1	R	bool	CF card inserted
0x1BE2	R	bool	CF card removed
0x2000	-	-	Acyclical writing of 4 byte (special function, for Profibus only)
0x2002	-	-	Acyclical writing of 22 byte (special function, for Profibus only)
0x2010	-	-	Acyclical reading of 4 byte (special function, for Profibus only)
0x2012	-	-	Acyclical reading of 22 byte (special function, for Profibus only)
0x9000	R/W	char 1204	Recipe of the active machine/batch 0
0x9400	R/W	char 1204	Recipe of the active machine/batch 1
0x9800	R/W	char 1204	Recipe of the active machine/batch 2

A

B

C

D

E

F

H

I

M

O

P

R

S

T

U

W

03.08/00578773

WEST Control Solutions

Austria	T: +43 (0) 2236 691 121
China	T: +86 22 8398 8098
France	T: +33 (1) 77 80 90 40
Germany	T: +49 (0) 561 505 1307
UK	T: +44 (0) 1273 606 271
USA	T: +1 800 866 6659

Email	Inquiries@West-CS.com
Website	www.West-CS.com

PMA WEST CAL Partlow