



## DataVU 5 - Interface Manual Modbus

59482/1



|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>                                     | <b>5</b>  |
| 1.1      | Preface .....   | 5         |
| 1.2      | Typographical conventions .....                         | 6         |
| 1.2.1    | Warning signs .....                                     | 6         |
| 1.2.2    | Note signs .....  | 6         |
| 1.2.3    | Representation modes .....                              | 6         |
| <b>2</b> | <b>General</b>  | <b>7</b>  |
| 2.1      | Areas of application .....                              | 7         |
| 2.2      | System requirements .....                               | 7         |
| 2.3      | Identifying the interface .....                         | 7         |
| <b>3</b> | <b>Connecting the interface</b>                         | <b>8</b>  |
| 3.1      | Connection diagram .....                                | 8         |
| 3.2      | RS232 .....   | 10        |
| 3.3      | Switching between RS232 and RS485 .....                 | 10        |
| <b>4</b> | <b>Protocol description</b>                             | <b>11</b> |
| 4.1      | Master-slave principle .....                            | 11        |
| 4.2      | Transmission mode (RTU) .....                           | 11        |
| 4.3      | Device address .....                                    | 12        |
| 4.4      | Timing sequence for communication .....                 | 12        |
| 4.5      | Structure of the data blocks .....                      | 14        |
| 4.6      | Difference between MODbus and Jbus .....                | 14        |
| 4.7      | Checksum (CRC16) .....                                  | 15        |
| 4.8      | Configuration of the interface on the back panel .....  | 16        |
| 4.9      | Configuration of the interface on the front panel ..... | 16        |
| 4.10     | Password protection for the serial interface .....      | 17        |
| <b>5</b> | <b>Functions</b>  | <b>18</b> |
| 5.1      | Read n words .....                                      | 19        |
| 5.2      | Write one word .....                                    | 20        |
| 5.3      | Write n words .....                                     | 21        |

---

# Contents

---

|           |   |           |
|-----------|---|-----------|
| <b>6</b>  | <b>Data flow</b>                        | <b>22</b> |
| 6.1       | Transmission format .....               | 22        |
| <b>7</b>  | <b>Error messages</b>                   | <b>25</b> |
| 7.1       | Error handling .....                    | 25        |
| 7.2       | Error messages for invalid values ..... | 26        |
| <b>8</b>  | <b>Address tables</b>                   | <b>27</b> |
| 8.1       | Device data .....                       | 27        |
| 8.2       | Process data .....                      | 28        |
| <b>9</b>  | <b>Special process data</b>             | <b>32</b> |
| 9.1       | MODbus flag .....                       | 32        |
| <b>10</b> | <b>Index</b>                            | <b>33</b> |

---

## 1.1 Preface

Please read this Interface Description before commissioning the interface for the instrument. Keep the interface description in a place which is accessible to all users at all times.



All the necessary information for operating the interface is contained in this interface description. However, if any difficulties should still arise during start-up, please do not carry out any unauthorized manipulations. You could endanger your rights under the instrument warranty!

Please contact the nearest subsidiary or the head office in such a case.



When returning modules, assemblies or components, the regulations of EN 100 015 “Protection of electrostatically sensitive components” must be observed. Use only the appropriate **ESD** packaging for transport.

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

**ESD** = electrostatic discharge

---

# 1 Introduction

---

## 1.2 Typographical conventions

### 1.2.1 Warning signs

The symbols for **Danger** and **Caution** are used in these operating instructions under the following conditions:



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



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



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

---

### 1.2.2 Note signs



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

abc<sup>1</sup>

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

A marker in the text, and the footnote text.

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

---

### 1.2.3 Representation modes

0x0010

**Hexadecimal number** A hexadecimal number is identified by being preceded by an “0x” (here: 16 decimal).

---

### 2.1 Areas of application

The serial interface – RS232 or RS485/ (MODbus, Jbus) – is used for communication with supervisory systems, such as a bus system or a PC. It is used to perform various functions, such as:

- to read out measurements from the paperless recorder
- to read out device and process data from the paperless recorder.

### 2.2 System requirements

The following items are required for operating the serial interface:

- paperless recorder with a serial interface (extra code)
- connecting cable, e.g.
  - PC interface with TTL/RS232 converter and adapter
  - PC interface with USB/RS232 converter and adapter
- setup or evaluation program, e.g.
  - setup program
  - PC evaluation software PCA3000
  - PCA communications software PCC
- PC or notebook

### 2.3 Identifying the interface

The serial interface RS232 / RS485 (MODbus, Jbus) is available as extra code. To see whether the device already has a serial interface, use the menu *Device info* → *Option Digital I/O* to investigate the configuration.



| Instrument info    |            |
|--------------------|------------|
| Version number     | 208.01.01  |
| UdN number         |            |
| Serial number      | 0000000000 |
|                    | 0000000000 |
| Input-card 1       | 3 inputs   |
| Input-card 2       | 3 inputs   |
| Digital I/O option | Yes        |
| Maths option       | Enabled    |
| Counter/int.option | Enabled    |
| Error              | No         |
| Power-off date     | 31.05.06   |
| Power-off time     | 13:56:45   |
| Power-on date      | 31.05.06   |
| Power-on time      | 13:57:03   |

If the extra code *Digital I/O option* is available (Yes), then the instrument already has a serial interface.

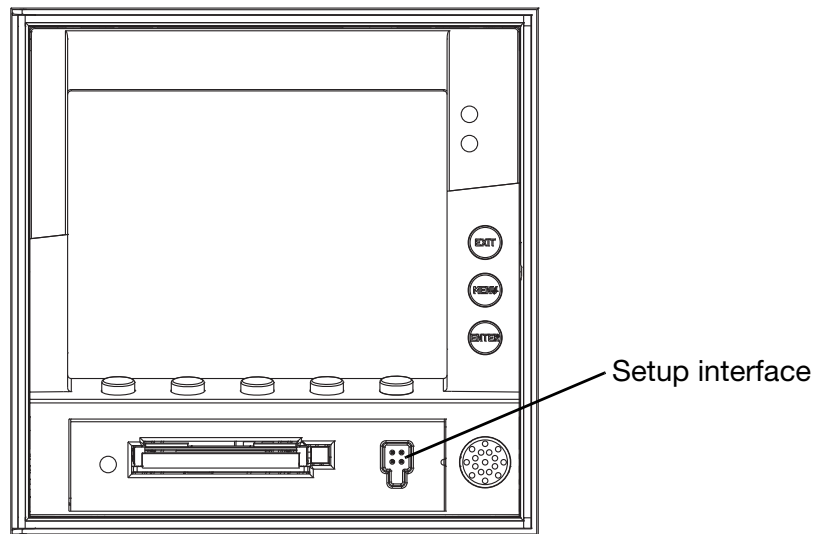
## 3 Connecting the interface

---

### 3.1 Connection diagram

---

Front panel of  
the  
paperless  
recorder



Connection  
diagram



The connection on the front panel can only be made through the setup interface with the connecting cable for “PC interface with TTL/RS232 converter and adapter”.

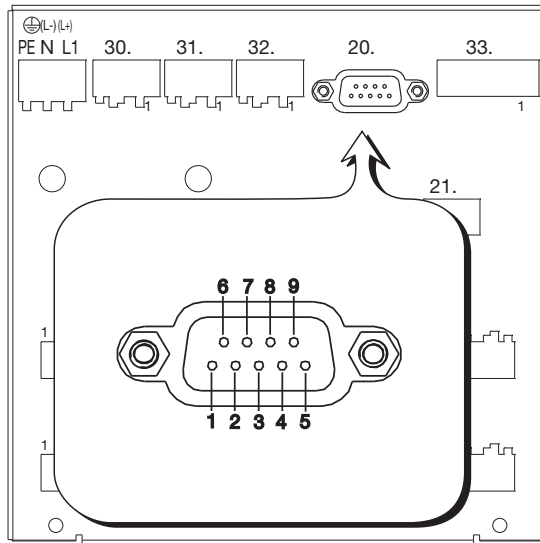
If the PC or notebook does not have a serial interface, then the connecting cable “PC interface with USB/RS232 converter and adapter” is also required.

---



# 3 Connecting the interface

Rear view of the paperless recorder



Connection diagram

| Connector 20 |               |
|--------------|---------------|
| RS232        | RS485         |
| 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 using a twisted-pair connecting cable with shielding!

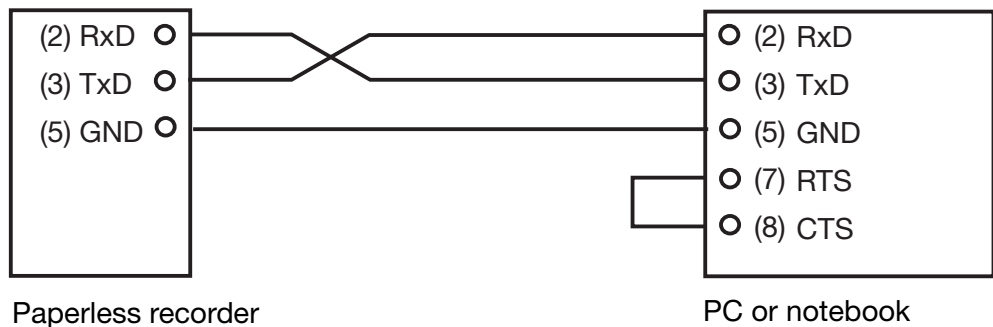
# 3 Connecting the interface

## 3.2 RS232

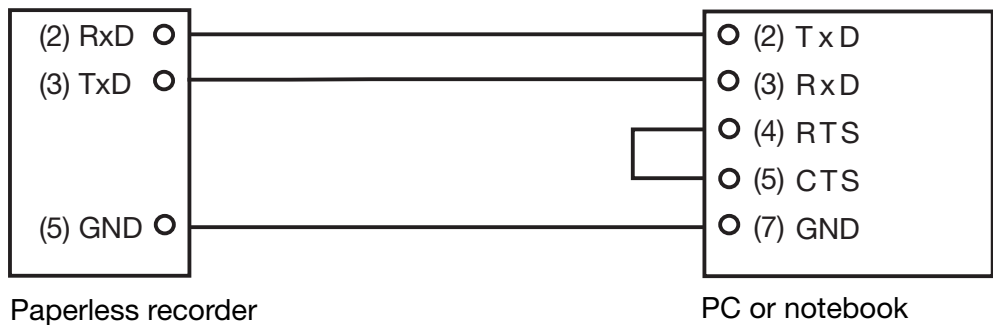
In the case of the RS232 interface, the handshake lines (RTS, CTS) are not used. The RTS line from the master (PC or notebook), which is the CTS line for the paperless recorder, will be ignored. The response is sent back immediately by the recorder. The CTS line of the master (RTS on the recorder) remains open.

If the program that is used evaluates the handshake lines, then 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



Only the signal lines shown above are to be connected. If this is not observed, then the recorder may switch over to RS485 mode.

## 3.3 Switching between RS232 and RS485

The changeover between the RS232 and RS485 interfaces is made through the recorder parameter

*Configuration level* → *Interface* → *Interface type*

or, using the setup program

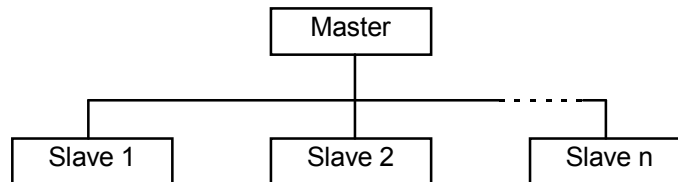
*Edit* → *Configuration level* → *Interface* → *Interface type*

## 4 Protocol description

---

### 4.1 Master-slave principle

The communication between a master (PC or notebook) and a slave device (paperless recorder), using MODbus/Jbus, takes place according to the master-slave principle, in the form of a data request/instruction – response.



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



The paperless recorder cannot be used as the master, it can only be operated as a slave.

---

### 4.2 Transmission mode (RTU)

The transmission mode used is the RTU mode (Remote Terminal Unit). Data are transmitted in binary format (hexadecimal) with 8 bits, as 16-bit integer values, or as 32-bit float values.

#### Data format

The data format describes the structure of a transmitted byte.

| Data word | Parity bit | Stop bit | Bit number |
|-----------|------------|----------|------------|
| 8 bits    | —          | 1        | 9          |
| 8 bits    | —          | 2        | 10         |
| 8 bits    | even       | 1        | 10         |
| 8 bits    | odd        | 1        | 10         |



The data format that is to be used can be selected, see Chapter 4.8 Configuration of the interface on the back panel, page 16.

---

## 4 Protocol description

---

### 4.3 Device address

The device address for the paperless recorder can be set between 1 and 254 (decimal), see Chapter 4.8 Configuration of the interface on the back panel, page 16.



A maximum of 31 paperless recorders can be addressed via the RS485 interface.

Device address 0 is reserved.

If only **one** recorder is connected to the PC or notebook, then it can also be accessed through device address 255 (even if a different address has been configured). The paperless recorder will always respond to instructions for device address 255.

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

---

### 4.4 Timing sequence for communication

#### Character transmission time

The start and end of a data block are marked by transmission pauses. The character transmission time (the time taken to transmit one character) depends on the baud rate and the data format that is used.

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

$$\text{character transmission time [msec]} = 1000 * 9 \text{ bits} / (\text{baud rate})$$

For the other data formats it is:

$$\text{character transmission time [msec]} = 1000 * 10 \text{ bits} / (\text{baud rate})$$

#### Example

| Baud rate [bps] | Data format [bit] | Character transmission time [msec] |
|-----------------|-------------------|------------------------------------|
| 38400           | 10                | 0.260                              |
|                 | 9                 | 0.234                              |
| 19200           | 10                | 0.521                              |
|                 | 9                 | 0.469                              |
| 9600            | 10                | 1.042                              |
|                 | 9                 | 0.938                              |

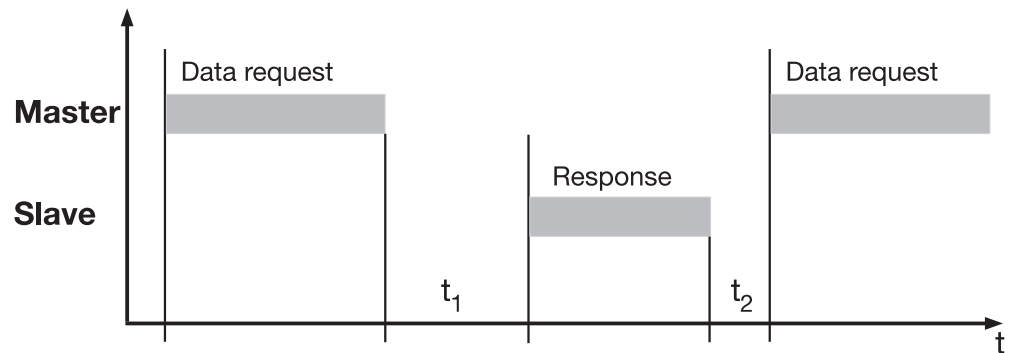
The baud rate is selectable, see Chapter 4.8 Configuration of the interface on the back panel, page 16.

---

## 4 Protocol description

### Timing sequence

A data request runs according to the following timing sequence:



$t_1$  Internal waiting time of the paperless recorder before checking the data request, and the internal processing time.

min.: 12.5 msec  
typical: 12.5 – 30 msec  
max. 1 sec



A minimum response time can be set in the instrument, under the menu item *Configuration* → *Interface*. This preset time is the minimum waiting time that must elapse before an answer is transmitted (0 – 500 msec). If a smaller value is set, then the response time may be longer than the preset value (because the internal processing time is longer), the recorder will then answer as soon as the internal processing is completed. A preset time of 0 msec means that the recorder answers with the maximum possible speed.

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

$t_2$  This is the waiting time which the master has to observe before initiating a new data request.

for RS232 at least 3.5 x the transmission time for one character (this time depends on the baud rate)

for RS485 25msec

While  $t_1$  and  $t_2$  are running, the master must not present any further data requests, since the paperless recorder will either ignore them or declare them to be invalid.

## 4 Protocol description

---

### 4.5 Structure of the data blocks

**Data structure** All data blocks have the same structure:

| Slave address | Function code | Data field | Checksum<br>CRC16 |
|---------------|---------------|------------|-------------------|
| 1 byte        | 1 byte        | x byte(s)  | 2 bytes           |

Each data block contains four fields:

**Slave address** the device address of a specific paperless recorder

**Function code** function selection (read/write a word)

**Data field** contains the information:  
- word address  
- word number  
- word value

**Checksum** detection of transmission errors

---

### 4.6 Difference between MODbus and Jbus

The MODbus protocol is compatible with the Jbus protocol. The structure of the data blocks is identical.



The difference between MODbus and Jbus is, that the absolute data addresses are shifted by +1 for Jbus.

| Absolute address | Jbus address | Modbus address |
|------------------|--------------|----------------|
| 0                | 1            | 0              |
| 1                | 2            | 1              |
| 2                | 3            | 2              |
| ...              | ...          | ...            |

---

## 4.7 Checksum (CRC16)

**Calculation method**

The checksum (CRC16) is used to recognize transmission errors. If an error is detected during evaluation, the corresponding device will 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 checksum is transmitted first.

**Example 1**

Read out the measurement input 2 (present value = 58.272) of recorder 20 (0x14).

Data request to slave 0x14: read two words, starting at address 0x37 (CRC16 = 0x0077)

|    |    |      |      |      |
|----|----|------|------|------|
| 14 | 03 | 0037 | 0002 | 7700 |
|----|----|------|------|------|

Response (CRC = 0x1DFA):

|    |    |    |        |        |      |
|----|----|----|--------|--------|------|
| 14 | 03 | 04 | 1687   | 4269   | FA1D |
|    |    |    | Word 1 | Word 2 |      |

Word 1 + Word 2 produce the response 58.272.

**Example 2**

Request status of the relay outputs.

Instruction: read one word from address 0x31 (CRC16 = 0x00D7)

|    |    |      |      |      |
|----|----|------|------|------|
| 14 | 03 | 0031 | 0001 | D700 |
|----|----|------|------|------|

Response (CRC = 0x4774):

|    |    |    |        |      |
|----|----|----|--------|------|
| 14 | 03 | 02 | 0001   | 7447 |
|    |    |    | Word 1 |      |

Word 1 indicates that only Output 1 is active.

## 4 Protocol description

---

### 4.8 Configuration of the interface on the back panel

#### Configuration on the paperless recorder

- \* Select *Configuration* → *Interface* on the paperless recorder.  
The parameters for the configuration of the interface will now be available.

|                         | Parameter            | Value/selection  | Description  |
|-------------------------|----------------------|--|--|
| Interface type          | → Interface type     | RS 232,<br><b>RS 485</b>                               | see Chapter 3.3 Switching between RS232 and RS485, page 10.  |
| Protocol                | → Protocol           | <b>MODbus</b> ,<br>Jbus                                | see Chapter 4.6 Difference between MODbus and Jbus, page 14. |
| Baud rate               | → Baud rate          | 9600 bps,<br>19200 bps,<br><b>38400 bps</b>            | see Chapter 4.4 Timing sequence for communication, page 12.  |
| Transmission mode (RTU) | → Data format        | <b>8-1-none</b> ,<br>8-1-odd,<br>8-1-even,<br>8-2-none | see Chapter 4.2 Transmission mode (RTU), page 11.            |
| Device address          | → Device address     | <b>1</b> – 254   | see Chapter 4.3 Device address, page 12.                     |
| Min. response time      | → Min. response time | <b>0</b> – 500msec                                     | see Chapter 4.4 Timing sequence for communication, page 12.  |

#### Configuration through setup program

Configuration with the aid of the setup software is made through the menu item *Edit* → *Configuration level* → *Interface*.

---

### 4.9 Configuration of the interface on the front panel

The transmission parameters for the interface are fixed in the instrument, and cannot be altered.

- baud rate = 9600 bps
- data format = 8 data bits, 1 stop bit, no parity

The interfaces on the front and back panel have identical device addresses.



We recommend using device address 255 for the setup interface on the front panel. The instrument will then always answer, irrespective of the configured device address.



### 4.10 Password protection for the serial interface

Access to the serial interface can be protected by a password (1 – 9999).

On the paperless recorder:

(*Configuration* → *Device data* → *Code No. (password)* → RS232/RS485

or,

using the setup program:

(*Edit* → *Device data* → *Code numbers* → *Interface*)

If password protection is active (i.e. password is **not** 0), then it is only possible to communicate with the device when the password has been written to MODbus address 0x7007 in the recorder. This prevents the unauthorized reading of data from the paperless recorder.

When password protection is active, only addresses

0x0000 to 0x001E (software version etc.) and

0x7008 (flag to indicate whether password protection is active) can be read, and address

0x7007 (interface password) can be written.

When the correct password has been transmitted, the password protection is instantly removed.



If there is no transmission via the MODbus interface for 10 seconds, then the protection will be re-activated!

If an incorrect password is sent to the device, then MODbus communication remains blocked. In this case, the device responds with error code 02. A fresh attempt at password entry will only be permitted after 10 seconds, to make it more difficult to try out passwords!



Address 0x7008 can be queried, to see whether the password protection is active:

0 = password protection inactive

1 = password protection active

## 5 Functions

---

### Function summary

The functions described below can be used to read out measurements and other device and process data from the paperless recorder.

| Function number | Function       | Restriction                |
|-----------------|----------------|----------------------------|
| 0x03 or 0x04    | read n words   | max. 127 words (254 bytes) |
| 0x06            | write one word | max. 1 word (2 bytes)      |
| 0x10            | write n words  | max. 127 words (254 bytes) |



If the recorder does not respond to one of these functions, or reacts by generating an error code, then please refer to Chapter 7 Error messages, page 25.

No other MODbus functions are implemented in this instrument apart from those explained in this interface description.

---

## 5.1 Read n words

This function reads n words, starting from a defined address.

### Data request

| Slave address | Function<br>0x03 or 0x04 | Address<br>of first word | Word<br>number | Checksum<br>CRC16 |
|---------------|--------------------------|--------------------------|----------------|-------------------|
| 1 byte        | 1 byte                   | 2 bytes                  | 2 bytes        | 2 bytes           |

### Response

| Slave address | Function<br>0x03 or 0x04 | Number<br>of bytes<br>read | Word<br>value(s) | Checksum<br>CRC16 |
|---------------|--------------------------|----------------------------|------------------|-------------------|
| 1 byte        | 1 byte                   | 1 byte                     | x byte(s)        | 2 bytes           |

### Example

Read the first 3 analog inputs

For the addresses of the analog inputs, see Chapter 8.2 Process data, page 28.

Data request: (CRC16 = 1A57)

|    |    |      |      |      |
|----|----|------|------|------|
| 14 | 03 | 004D | 0006 | 571A |
|----|----|------|------|------|

Response: (CRC16 = 4750)

|    |    |    |                        |                        |                        |      |      |      |      |
|----|----|----|------------------------|------------------------|------------------------|------|------|------|------|
| 14 | 03 | 0C | 1999                   | 4348                   | 4CCC                   | 4348 | 2666 | 4396 | 5047 |
|    |    |    | Measurement 1<br>200.1 | Measurement 2<br>200.3 | Measurement 3<br>300.3 |      |      |      |      |

# 5 Functions

---

## 5.2 Write one word

For the “write word” function, the data blocks for instruction and response are identical.

### Instruction

| Slave address | Function<br>0x06 | Word address | Word value | Checksum<br>CRC16 |
|---------------|------------------|--------------|------------|-------------------|
| 1 byte        | 1 byte           | 2 bytes      | 2 bytes    | 2 bytes           |

### Response

| Slave address | Function<br>0x06 | Word address | Word value | Checksum<br>CRC16 |
|---------------|------------------|--------------|------------|-------------------|
| 1 byte        | 1 byte           | 2 bytes      | 2 bytes    | 2 bytes           |

---

### Example

Set the MODbus flag (bit 0 of address 0x0033).

For the address of the MODbus flag, see Chapter 8.2 Process data, page 28.  
For the function of the MODbus flag, see Chapter 9.1 MODbus flag, page 32.

Instruction: (CRC16 = C0BA)

|    |    |      |      |      |
|----|----|------|------|------|
| 14 | 06 | 0033 | 0001 | BAC0 |
|----|----|------|------|------|

Response (as instruction):

|    |    |      |      |      |
|----|----|------|------|------|
| 14 | 06 | 0033 | 0001 | BAC0 |
|----|----|------|------|------|

---

## 5.3 Write n words

### Instruction

| Slave address | Function<br>0x10 | Address of first word | Word number | Byte number | Word value(s) | Checksum<br>CRC16 |
|---------------|------------------|-----------------------|-------------|-------------|---------------|-------------------|
| 1 byte        | 1 byte           | 2 bytes               | 2 bytes     | 1 byte      | x byte(s)     | 2 bytes           |

### Response

| Slave address | Function<br>0x10 | Address of first word | Word number | Checksum<br>CRC16 |
|---------------|------------------|-----------------------|-------------|-------------------|
| 1 byte        | 1 byte           | 2 bytes               | 2 bytes     | 2 bytes           |

### Example

Write the word "Test" (ASCII coding: 0x54 0x65 0x73 0x74 0x00) to the address 0x0080... and so on, so that this text is entered in the event list.

Instruction: (CRC16 = BFC8)

|    |    |      |      |    |                   |      |
|----|----|------|------|----|-------------------|------|
| 14 | 10 | 0080 | 0003 | 06 | 54 65 73 74 00 00 | C8BF |
|----|----|------|------|----|-------------------|------|

Response: (CRC16 = 03F3)

|    |    |      |      |      |
|----|----|------|------|------|
| 14 | 10 | 0033 | 0001 | F303 |
|----|----|------|------|------|

## 6.1 Transmission format

**Integer values** When using MODbus, integer values are transmitted in the following format: first the HIGH byte, then the LOW byte.

**Example**

Request the integer value from address 0x0000, when this address contains the value "20" (ASCII coding: 0x3230).

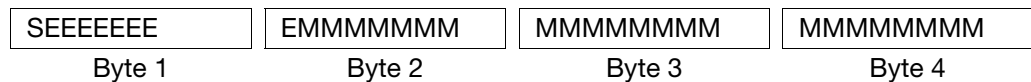
Request: 14030000000186CF (CRC16 = CF86)

Response: 140302**3230**A0F3 (CRC16 = xF3A0)

**Float values**

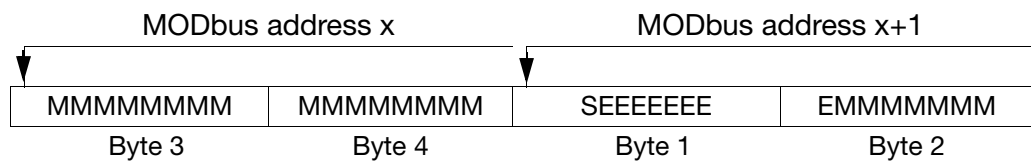
When using MODbus, float values are processed in the IEEE 754 standard format (32-bit), but with the difference that bytes 1 and 2 are swapped with bytes 3 and 4.

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



S = sign bit  
 E = exponent (complement to base 2)  
 M = 23-bit normalized mantissa

**MODbus float format**



**Example**

Request the float value from address 0x0066, when this address contains the value "550.0" (0x44098000 in IEEE 754 format).

Request: 140300350002D6C0 (CRC16 = C0D6)

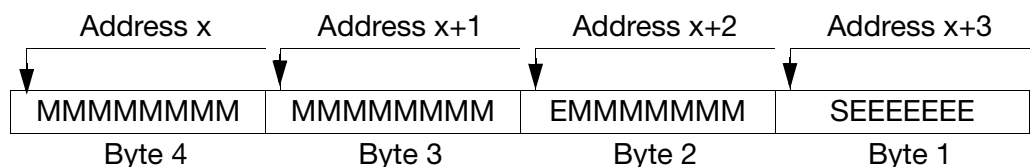
Response: 140304**80004409**6434 (CRC16 = 3464)

After the transmission from the device, the bytes of the float value must be swapped accordingly.



Many compilers (e. g. Microsoft Visual C++) store the float values in the following order:

**Float value**

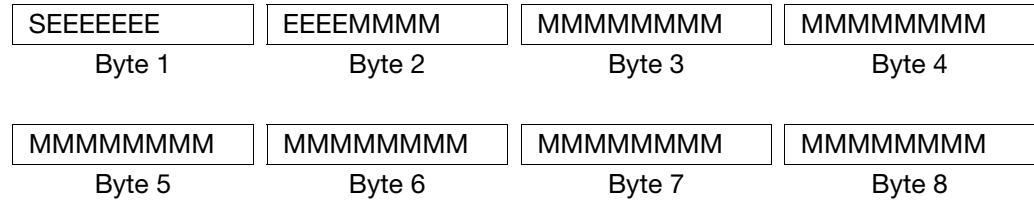


Please check how float values are stored in your application. If necessary, the bytes will have to be swapped accordingly in your interface program, after they have been fetched from the paperless recorder.

## Double values

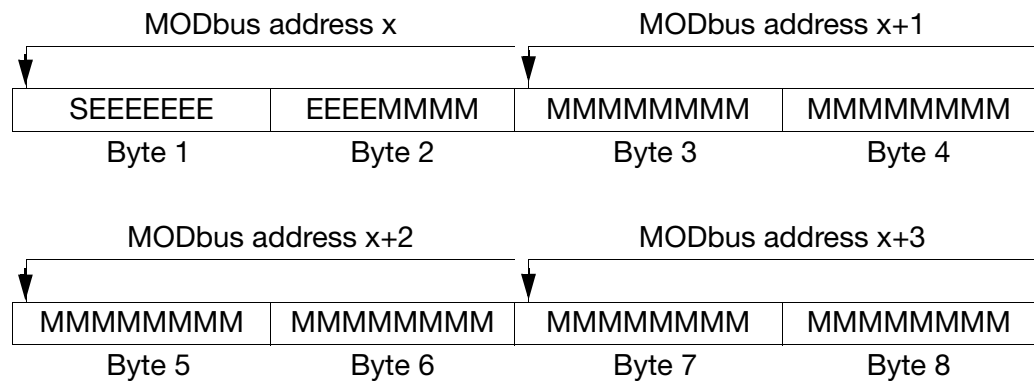
When using MODbus, double values are also processed in the IEEE-754 standard format (32bits). Unlike float values, no bytes are swapped for double values.

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



S = sign bit  
 E = exponent (complement to base 2)  
 M = 52-bit normalized mantissa

### MODbus double format



## Example

Request the double value from address 0x0035, when this address contains the value "1234567.89" (0x4132D687E3D70A3D in IEEE 754 format).

Request: 140300660004A6D3 (CRC16 = D3A6)

Response: 1403084**132D687E3D70A3DE**1C1 (CRC16 = C1E1)



Please check how double values are stored in your application. If necessary, the bytes will have to be swapped accordingly in your program, after they have been fetched from the paperless recorder.

## 6 Data flow

---

### Character strings (texts)

Character strings are transmitted in ASCII format.



A “\0” (ASCII code 0x00) must always be transmitted as the last character, to mark the end of the string. Any following characters are meaningless.

If a string of characters is transmitted to the recorder without the terminating “/0”, then the instrument will overwrite the last character with “/0” of its own accord!

Since text transmission is made as words (16-bit), if there is an odd number of characters (incl. “\0”), an additional 0x00 will be added on.

The maximum lengths given in the address tables (see Address tables, page 27 onwards) for character strings include the terminating “/0”, i.e. the text for “char 11” can have a maximum length of 10 characters.

---

### Example

Request the text from address 0x000E, when this address contains the character string “**LS500cf**”

(ASCII code: **0x4C, 0x35, 0x30, 0x30, 0x63, 0x66**, 0x20, 0x00).

Request: 1403000E0005E6CF (CRC = CFE6)

Response: 14030A**4C533530306366**2000AA91D6 (CRC16= D691)



Instead of “AA” in front of the CRC sum, there could be any value – since it comes after the “/0”, it will be ignored.

---



## 7.1 Error handling

### No response from the recorder

The slave will not respond if one of the following errors occurs:

- The baud rate and/or data format for the master (PC or notebook) does not match the slave (paperless recorder).
- the device address for the recorder does not match the address contained in the protocol
- the checksum (CRC16) is not correct
- the MODbus function is not supported by the device
- the instruction from the master is incomplete or over-defined
- the number of words to be read is zero
- communication is active on the setup interface.

In these cases, when the timeout of 2 seconds has expired, the data will have to be retransmitted.

### Error codes

If the data request from the master is received by the paperless recorder without any transmission error, but cannot be processed, then the recorder will answer with an error code.

The following error codes may appear:

- 02 invalid address or too many words to be read or written, or access to interface protected by password
- 03 value is outside the permitted range
- 08 write-protected value

### Response in the event of an error

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

The function code is ORed with 0x80, which means that the MSB (most significant bit) is set to 1.

### Example

Data request: (CRC16 = 792C)

|    |    |      |      |      |
|----|----|------|------|------|
| 14 | 03 | 1234 | 0001 | C279 |
|----|----|------|------|------|

Response: (CRC16 = 35D1)

|    |    |    |      |
|----|----|----|------|
| 14 | 83 | 02 | D135 |
|----|----|----|------|

# 7 Error messages

---

## 7.2 Error messages for invalid values

For measurements, the error number is shown in the value itself, i.e. the error number is inserted instead of the measurement.

| Error code for float values | Error code for double values | Error                   |
|-----------------------------|------------------------------|-------------------------|
| -200000.0                   | -8000000000000000000.0       | underrange              |
| 200000.0                    | 8000000000000000000.0        | overrange               |
| 200003.0                    | 8000000000000000003.0        | otherwise invalid value |

### Example

---

Data request: (CRC16 = D956)

|    |    |      |      |      |
|----|----|------|------|------|
| 14 | 03 | 004D | 0002 | 56D9 |
|----|----|------|------|------|

Response: (CRC16 = 03D8)

|    |    |    |      |      |      |
|----|----|----|------|------|------|
| 14 | 03 | 04 | 5000 | 4843 | D803 |
|----|----|----|------|------|------|

The measurement 0x48435000 (= 200000.0), provided by analog input 1, shows that an overrange has appeared.

---

## 8 Address tables

---

All process values (variables) together with their addresses, data types and access modes are described below.

References are as follows:

|         |  |
|---------|--|
| R       | read access only   |
| W       | write access only  |
| R/W     | read and write access  |
| char    | ASCII character (8 bits)   |
| byte    | byte (8 bits)  |
| int     | integer (16 bits)  |
| char xx | character string of length xx;<br>xx = length, including the string termination character “\0” |
| Bit x   | bit No. x  |
| float   | float value (4 bytes)  |
| double  | double value (8 bytes)   |

The process values are divided into logical groups.

In the following address tables, bit 0 is always the least significant bit.

### 8.1 Device data

| Address | Access | Data type | Signal designation      |
|---------|--------|-----------|-------------------------|
| 0x0000  | R      | char 11   | Software version        |
| 0x0006  | R      | char 13   | VdN number              |
| 0x000E  | R      | char 9    | Device name (“LS500cf”) |
| 0x0013  | R      | char 21   | Serial No.              |

## 8 Address tables

---

### 8.2 Process data

| Address | Access | Data type | Signal designation   |
|---------|--------|-----------|--|
| 0x002F  | R      | int       | status of the binary inputs  |
|         | R      | Bit0-7    | free   |
|         | R      | Bit8      | binary input 1<br>0 = open / 1 = closed  |
|         | R      | Bit9      | binary input 2   |
|         | R      | Bit10     | binary input 3   |
|         | R      | Bit11     | binary input 4   |
|         | R      | Bit12-15  | free   |
| 0x0030  | R      | int       | additional binary signals  |
|         | R      | Bit0      | memory alarm for internal memory for read-out via CF card<br>0 = no alarm<br>1 = memory nearly full          |
|         | R      | Bit1      | memory alarm for internal memory for read-out via serial interface<br>0 = no alarm<br>1 = memory nearly full |
|         | R      | Bit2-7    | free   |
|         | R      | Bit8      | combination alarm<br>0 = no alarm<br>1 = at least 1 limit infringed in the device                            |
|         | R      | Bit9      | memory alarm, CF card<br>0 = no alarm<br>1 = CF card nearly full   |
|         | R      | Bit10     | fault<br>0 = no fault<br>1 = fault   |
|         | R      | Bit11     | Low combination alarm<br>0 = no Low alarm<br>1 = at least 1 Low alarm present                                |
|         | R      | Bit12     | High combination alarm<br>0 = no High alarm<br>1 = at least 1 High alarm present                             |
|         | R      | Bit13     | counter/integrator combination alarm<br>0 = no alarm<br>1 = at least 1 counter/integrator limit infringement |
|         | R      | Bit14     | CF card<br>0 = no CF card in slot<br>1 = CF card is plugged in   |
|         | R      | Bit15     | free   |
| 0x0031  | R      | int       | status of relay outputs and logic channels   |
|         | R      | Bit0      | relay output 1<br>0 = inactive<br>1 = active   |

## 8 Address tables

| Address | Access | Data type | Signal designation   |
|---------|--------|-----------|--|
|         | R      | Bit1      | relay output 2   |
|         | R      | Bit2      | relay output 3   |
|         | R      | Bit3-7    | free   |
|         | R      | Bit8      | logic channel 1<br>0 = false<br>1 = true                                   |
|         | R      | Bit9      | logic channel 2  |
|         | R      | Bit10     | logic channel 3  |
|         | R      | Bit11     | logic channel 4  |
|         | R      | Bit12     | logic channel 5  |
|         | R      | Bit13     | logic channel 6  |
|         | R      | Bit14-15  | free   |
| 0x0032  | R      | int       | counter/integrator alarms  |
|         | R      | Bit0-7    | free   |
|         | R      | Bit8      | alarm, counter/integrator channel 1<br>0 = no alarm<br>1 = limit infringed |
|         | R      | Bit9      | alarm, counter/integrator channel 2  |
|         | R      | Bit10     | alarm, counter/integrator channel 3  |
|         | R      | Bit11     | alarm, counter/integrator channel 4  |
|         | R      | Bit12     | alarm, counter/integrator channel 5  |
|         | R      | Bit13     | alarm, counter/integrator channel 6  |
|         | R      | Bit14-15  | free   |
| 0x0033  | R/W    | int       | flag for control of<br>various device functions                            |
|         | R/W    | Bit0      | MODbus flag<br>0 = false<br>1 = true                                       |
|         | R/W    | Bit1-15   | free   |
| 0x0034  | R      | int       | alarms for analog channels   |
|         | R      | Bit0      | Low alarm, channel 1<br>0 = no alarm<br>1 = underrange                     |
|         | R      | Bit1      | Low alarm, channel 2   |
|         | R      | Bit2      | Low alarm, channel 3   |
|         | R      | Bit3      | Low alarm, channel 4   |
|         | R      | Bit4      | Low alarm, channel 5   |
|         | R      | Bit5      | Low alarm, channel 6   |
|         | R      | Bit6-7    | free   |
|         | R      | Bit8      | High alarm, channel 1<br>0 = no alarm<br>1 = overrange                     |
|         | R      | Bit9      | High alarm, channel 2  |
|         | R      | Bit10     | High alarm, channel 3  |
|         | R      | Bit11     | High alarm, channel 4  |
|         | R      | Bit12     | High alarm, channel 5  |
|         | R      | Bit13     | High alarm, channel 6  |
|         | R      | Bit14-15  | free   |

## 8 Address tables

---

| Address | Access | Data type | Signal designation                        |
|---------|--------|-----------|---|
| 0x0035  | R      | float     | analog channel 1                          |
| 0x0037  | R      | float     | analog channel 2                          |
| 0x0039  | R      | float     | analog channel 3                          |
| 0x003B  | R      | float     | analog channel 4                          |
| 0x003D  | R      | float     | analog channel 5                          |
| 0x003F  | R      | float     | analog channel 6                          |
| 0x0041  | R      | float     | counter/integrator channel 1 <sup>1</sup> |
| 0x0043  | R      | float     | counter/integrator channel 2 <sup>1</sup> |
| 0x0045  | R      | float     | counter/integrator channel 3 <sup>1</sup> |
| 0x0047  | R      | float     | counter/integrator channel 4 <sup>1</sup> |
| 0x0049  | R      | float     | counter/integrator channel 5 <sup>1</sup> |
| 0x004B  | R      | float     | counter/integrator channel 6 <sup>1</sup> |
| 0x004D  | R      | float     | analog input 1                            |
| 0x004F  | R      | float     | analog input 2                            |
| 0x0051  | R      | float     | analog input 3                            |
| 0x0053  | R      | float     | analog input 4                            |
| 0x0055  | R      | float     | analog input 5                            |
| 0x0057  | R      | float     | analog input 6                            |
| 0x0059  | R      | float     | math channel 1                            |
| 0x005B  | R      | float     | math channel 2                            |
| 0x005D  | R      | float     | math channel 3                            |
| 0x005F  | R      | float     | math channel 4                            |
| 0x0061  | R      | float     | math channel 5                            |
| 0x0063  | R      | float     | math channel 6                            |

---

<sup>1</sup> The values in the recorder are double values (8 bytes).  
Since only float values (4 bytes) can be read for this address, only a limited resolution is possible (limitation of count range).  
The values can be read in double format at address 0x0066.

## 8 Address tables

---

| Address | Access | Data type | Signal designation           |
|---------|--------|-----------|------------------------------|
| 0x0066  | R      | double    | counter/integrator channel 1 |
| 0x006A  | R      | double    | counter/integrator channel 2 |
| 0x006E  | R      | double    | counter/integrator channel 3 |
| 0x0072  | R      | double    | counter/integrator channel 4 |
| 0x0076  | R      | double    | counter/integrator channel 5 |
| 0x007A  | R      | double    | counter/integrator channel 6 |

| Address | Access | Data type | Signal designation                                |
|---------|--------|-----------|---|
| 0x0080  | R/W    | char 21   | message text<br>(for the entry in the event list) |

| Address | Access | Data type     | Signal designation   |
|---------|--------|---------------|--|
| 0x7007  | W      | short integer | password for communication   |
| 0x7008  | R      | short integer | info flag, whether or not communication is prevented by password:<br>0 = measurement data can be read<br>1 = password entry required |

---

### 9.1 MODbus flag

The MODbus flag can be used just like other binary signals (e.g. binary inputs or alarms) to operate various functions in the paperless recorder. In order to be able to make use of the MODbus flag, the entry “MODbus flag” must be selected in the configuration of the paperless recorder.

One conceivable application for the MODbus flag would be, for example, the activation of event mode through the serial interface.

---



### A

Address tables 27  
Alarm 29  
Areas of application 7

### B

Baud rate 12, 16

### C

Checksum (CRC16) 15  
Configuration through setup program 16  
Configuration via keypad 16  
Connecting cable 10  
Connection diagram 8–9

### D

Data structure 14  
Date format 11  
Device address 12, 16  
Device data 27  
Double values 23

### E

Error handling 25  
Errors 25

### F

Float values 22  
Functional summary 18

### I

Integer values 22  
Interface type 7, 16

### J

Jbus 14

## M

Measurement 26  
Measurement input 15  
Min. response time 13, 16  
MODbus flag 20, 29, 32

## O

Overrange 26

## P

Parity bit 11  
Program version 7  
Protocol 16

## R

Read command 19  
Relay outputs 15  
RS232 9–10  
RS485 9–10

## S

Stop bit 11  
System requirements 7

## T

Terminal assignment 10  
Timing sequence 13

## U

Underrange 26

## W

Write command 20–21



05.06/00578805

---

**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](mailto:Inquiries@West-CS.com)  
**Website** [www.West-CS.com](http://www.West-CS.com)



**WEST**  **Partlow**

---