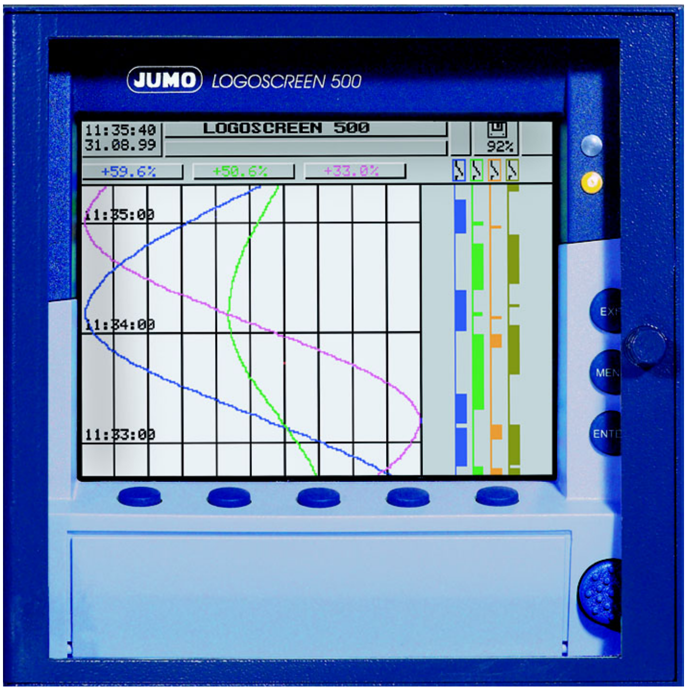


JUMO LOGOSCREEN 500

Paperless recorder

B 95.5015.2

Interface description

03.07/00379094

Contents

Contents

1.1 Preface

Please read these Operating Instructions before starting up the interface. Keep these operating instructions in a place which is accessible to all users at all times.

Please assist us to improve these operating instructions where necessary.

Your suggestions will be appreciated.

Phone	in Germany	(0661) 6003-725
	from abroad	(+49) 661 6003-0
Fax	in Germany	(0661) 6003-681
	from abroad	(+49) 661 6003-607

All the necessary information for operating the interface is contained in these operating instructions. However, if any problems should still arise during start-up, you are asked not to carry out any unauthorized manipulations. You could endanger your rights under the warranty!

Please contact the nearest subsidiary or the main factory.

When returning chassis, modules or components, the rules 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** (electrostatic discharge).

1.2 Typographical conventions

1.2.1 Warning signs

The signs of **Danger** and **Warning** are used in these Operating Instructions under the following conditions:

Danger This sign is used when there may be **danger to personnel** if the instructions are disregarded or not followed accurately!

Warning This sign is used when there may be **damage to equipment or data** if the instructions are disregarded or not followed accurately!

Warning This sign is used where special care is required when handling electrostatically sensitive components.

1.2.2 Note signs

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

Reference This sign refers to further information in other handbooks, chapters or sections.

abc[1] **Footnote** Footnotes are notes which refer to certain points in the text. Footnotes consist of two parts:

Marking in the text and the footnote text.

The text markings are arranged as continuous raised (superscript) numbers.

1.2.3 Representation

0x0010 **Hexadecimal number** A hexadecimal number is identified by being preceded by a “0x” (here: 16 decimal).

2.1 Application

The RS232 or RS485 serial interfaces are available for communication with supervisory systems (e. g. bus system or PC). They can, for example, be used to

- read out measurements from the paperless recorder
- read out instrument and process data from the paperless recorder

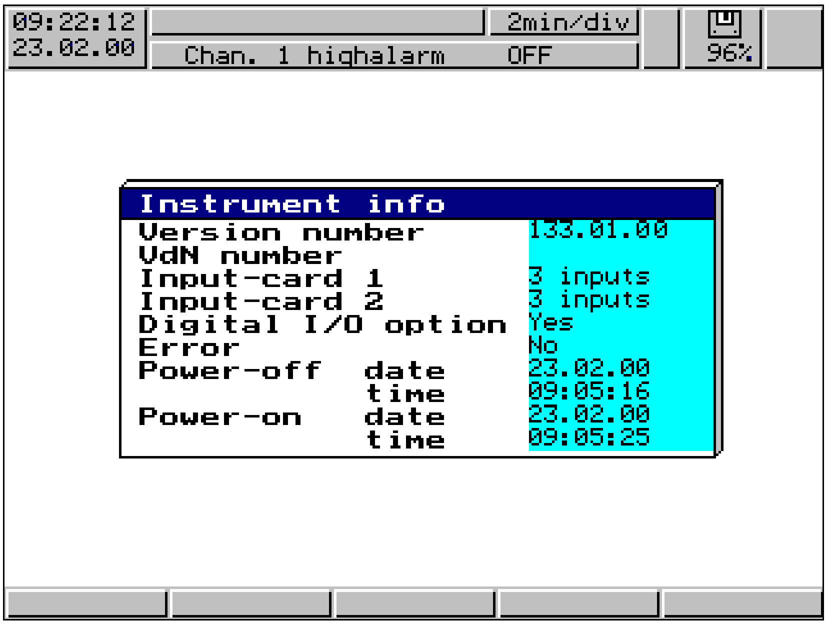
2.2 System requirements

The following items are necessary for operating the serial interface:

- paperless recorder with program version[1] from 133.01.01 (including serial interface)
- master (e.g. PC)
- connecting cable
- evaluation program PCA or others

2.3 Identifying the interface

The RS232 and RS485 interfaces can be supplied as extra code.
Whether they are implemented in the paperless recorder, can be requested via the menu *Instrument info* ➔ *Digital-I/O option*.



If extra code *Digital-I/O option* is available (*Yes*), then the interfaces are also available.

1. The program version can be requested in the menus of the paperless recorder, via *Instrument info* ➔ *Version number*.

2.4 Evaluation program PCA and communication server

PCA

The evaluation program PCA (from program version 108.03.01), which can be supplied as accessory, enables the graphical display of the measurement data acquired by the paperless recorder.

Die PCA version number can be obtained from within the PCA evaluation program, under the menu item *Help* ➔ *Info*

Communication server

Using the communication server (from program version 139.01.01), the measurement data stored in the recorder memory can be read out via the serial interface. A baud rate of 38400 bps when transferring data is recommended. The baud rate can be set on the recorder via the parameter *Configuration* ➔ *Interface* ➔ *Baud rate*.

The menu item *Archive* ➔ *Read in measurement data via interface* is available to read out the measurement data in the evaluation program PCA (in conjunction with the communication server).

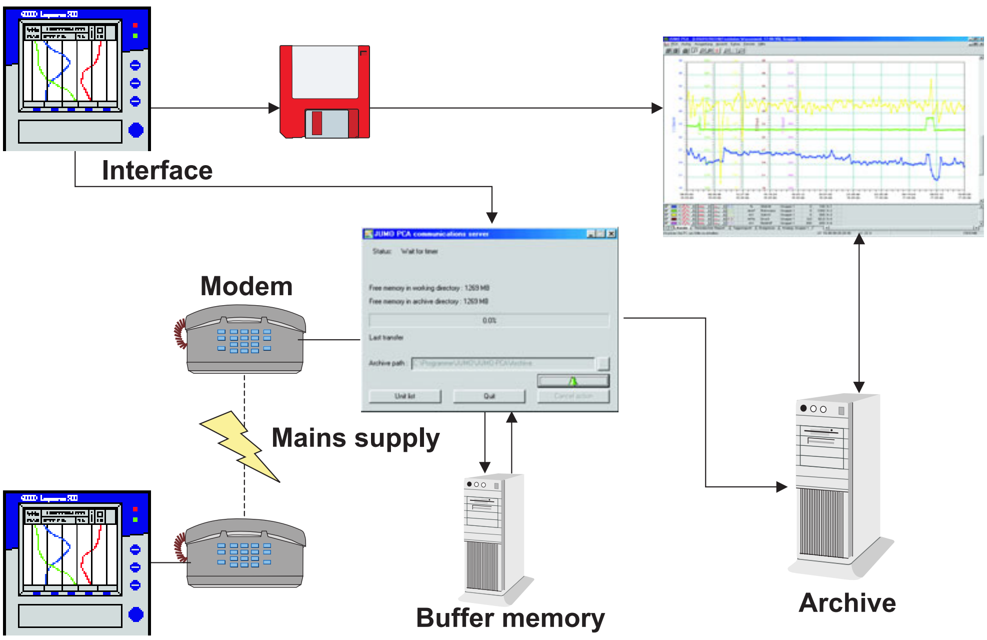
The version number of the communication server can be obtained as follows:

✱ Call up icon in the title bar by using the left mouse button

✱ Click on function *Info via PCA communication server ...*

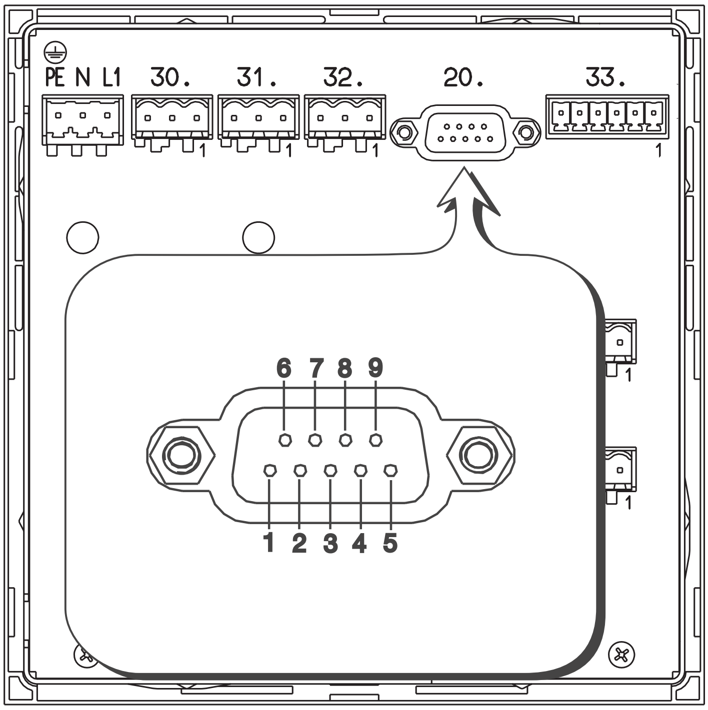
The data are read out "time-controlled" only. An "on-line connection" between the PC and the paperless recorder is not possible.

Overview of PCA and communication server



3.1 Connection diagram

Rear view of the paperless recorder



Interface

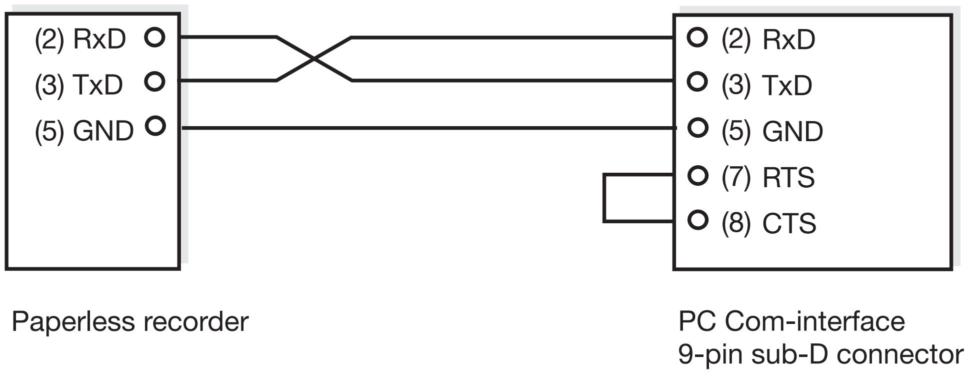
Connection diagram

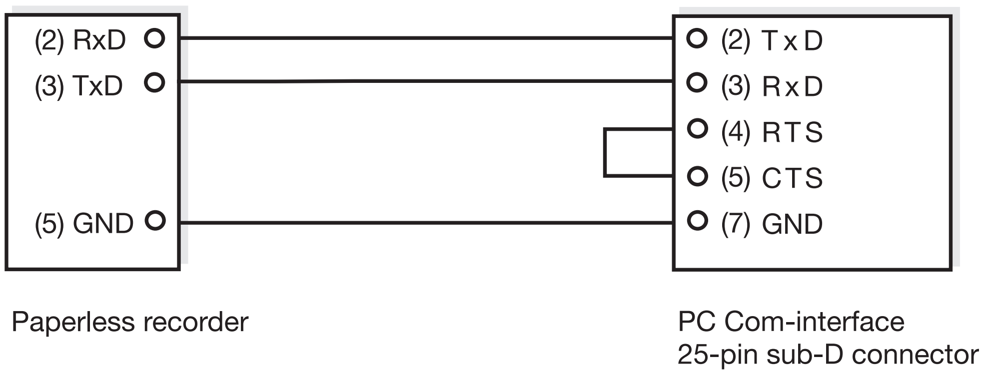
Connector 20			
RS232		**RS485**	
1 ○		1 ○	
2 ○	RxD	2 ○	
3 ○	TxD	3 ○	TxD+/RxD+
4 ○		4 ○	
5 ○	GND	5 ○	GND
6 ○		6 ○	
7 ○		7 ○	
8 ○		8 ○	TxD-/RxD-
9 ○		9 ○	

☞ The use of a twisted connecting cable with screening is recommended.

3.2 RS232

The handshake connections (RTS, CTS) are not used with the RS232 interface. The RTS connection from the master (CTS on the paperless recorder) is ignored; the answer is sent directly from the recorder. The CTS connection of the master (RTS on the paperless recorder) remains open. If the master evaluates the handshake connections, they have to be bridged in the cable.





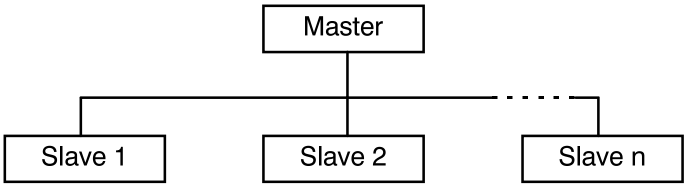
Only the signals listed above may be connected. If this is not the case, then a switchover to RS485 operation may occur.

3.3 Switching between RS232 and RS485

The recorder parameter *Configuration* ➔ *Interface* ➔ *Interface type* or the setup program (*Edit* ➔ *Interface* ➔ *Interface type*) can be used to switch between the RS232 and RS485 interfaces.

4.1 Master-slave principle

The communication between a PC (master) and an instrument (slave) using MOD/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 instrument address.

☞ The paperless recorder can only operate as slave, and not as master.

4.2 Transmission mode (RTU)

The transmission mode used is the RTU mode (Remote Terminal Unit). Data are transmitted in binary form (hexadecimal) with 8 bits, 16 bits for integers and 32 bits for float values.

Data format

The data format describes the arrangement of a byte transmitted. The data format can be as follows:

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

4.3 Instrument address

The instrument address of the slave can be set between 1 and 255 (decimal). The instrument address 0 is reserved.

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

The address is made in binary form (hexadecimal) in the transmission protocol.

4.4 Timing of the communication

Character transmission rate

The start and end of a data block are identified by transmission pauses. The character transmission time (time for the transmision of one 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 [msec] = 1000 * 9 bits / (baud rate)

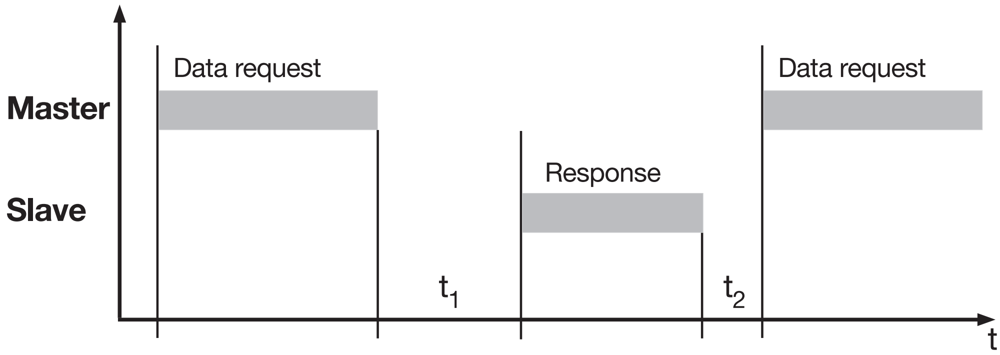
For other data formats this is:

character transmission time [msec] = 1000 * 10 bits / (baud rate)

Example

Baud rate [baud]	**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

Timing scheme A data request runs according to the following timing scheme:



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

min.:	12.5 msec
typically:	12.5 — 30 msec
max.:	2 sec

☞ A minimum response time can be set in the instrument, under the menu item *Configuration* ➔ *Interface*. This preset time is the minimum time which will be waited before a response is transmitted (0—500 msec). If a small value is set, then the response time may exceed the preset value (the internal processing time is longer) and the instrument answers as soon as the internal processing is completed. A preset time of 0 msec means that the instrument responds with the maximum possible speed.

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

t_2 Waiting time which the master has to keep before starting a new data request

on RS232	a minimum of 3.5 times the transmission time for 1 character (time depends on the baud rate)
on RS485	25 msec

The master must not make any data requests within t_1 and t_2, since if this is the case, the recorder will either ignore the request or declare it invalid.

4.5 Structure of the data blocks

Alle data blocks have the same structure:

Data structure

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

Each data block contains four fields:

Slave address instrument address of a specific slave

Function code function selection (read, write, bit, word)

Data field contains the information:

-bit address (word address)

-bit number (word number)

-bit value (word value)

Check sum recognition of transmission errors

4.6 Distinction MODbus/Jbus

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

☞ MODbus differs from Jbus in the absolute addresses of the data. The addresses of the MODbus are shifted by one compared with those of Jbus.

Absolute address	Jbus address	MODbus address
0	1	0
1	2	1
2	3	2
...	...	...

4.7 Check sum (CRC16)

The checksum (CRC16) serves to recognize transmission errors. If an error is identified during evaluation, the corresponding instrument does not answer.

Calculation scheme

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

The low byte of the checksum is transmitted first.

Example 1

Read out measurement input 2 (present value = 58.272).

Data request: Read two words from address 0x37 (CRC16 = 0x0077)

14	03	0037	0002	7700

Response (CRC = 0x1DFA):

14	03	04	1687	4269	FA1D
			Word 1	Word 2	

Word 1 and Word 2 result in the answer 58.272.

Example 2

Poll 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 means that only output 1 is active.

4.8 Configuring the interface

The interface is configured from the keys of the recorder, or via the setup program.

Configuration from the keys

The *Configuration* menu must be called up first and the parameter *Interface* selected. The parameters for configuring the interface are now available.

	Parameter	Value/selection	Description
Interface type	Configuration ➔ Interface ➔ Interface type	RS232, **RS485**	Select interface
Protocol	Configuration ➔ Interface ➔ Protocol	**MODBUS**, JBUS	Select protocol ⇨ Section 4.6 “Distinction MODbus/Jbus”
Baud rate	Configuration ➔ Interface ➔ Baud rate	9600 baud, 19200 baud, **38400 baud**	Select baud rate
Data format	Configuration ➔ Interface ➔ Data format	**8-1- none**, 8-1- odd, 8-1- even, 8-2- none	Select data format (data bit - stop bit - parity)
Instrument address	Configuration ➔ Interface ➔ Instrument address	**1**—255	Select address
Min. response time	Configuration ➔ Interface ➔ Min. response time	**0**—500msec	Select minimum response time

☞ The instrument address must also be observed when communicating via the RS232 interface, although it is not a bus interface.

Configuration via setup program

The menu item *Edit* ➔ *Interface* is available for configuration using the setup software.

4.9 Password protection for the serial interface

Password protection for the serial interface is available for paperless recorders with instrument software version 133.03.xx or above.

A password for the serial interface (0 — 9999) can be entered on the recorder (*Configuration ➔ Instrument data ➔ Code No.(Password)* ➔ RS232/RS485, or via the setup program (*Edit ➔ Instrument data ➔ Code numbers ➔ Interface*). If a password is not equal to 0, communication with the instrument can only take place if the password has been written to the recorder first, to MODbus address 0x7000. This will prevent any unauthorized readout of data, for example when communicating with an instrument via the modem connection.

The locking by password is removed after the correct password has been transmitted. It is reactivated only after 10 seconds have elapsed without any communication via the MODbus interface.

If a wrong password has been transmitted to the instrument, the MODbus communication remains inhibited. In this case, the instrument responds with error code 03. A further attempt at password entry will only be permitted after 10 seconds. This is intended to prevent passwords from being tried out.

Measurements and other instrument and process data can be read out from the paperless recorder using the functions described below.

Overview of functions

Function number	**Function**	
0x01/0x02	read n bits	(max. 256 bits)
0x03/0x04	read n words	(max. 127 words)
0x05	write one bit	
0x06	write one word	
0x10	write n words	(max. 127 words)

There are no separate areas for bit and word for the system variables. The bit and word areas overlap and can be read and written both as bit area and as word area.

Address calculation

The word address is calculated as follows:

word address = base address + variables address

The bit address is calculated as follows:

bit address = word address * 16 + bit number

Example: word address for the measurement from analog input 6:

word address = 0x0035 + 0x000A = 0x003F

Example: bit address of high alarm on channel 6:

bit address = (0x002F + 0x0005) · 0x0010 + 0x000D = 0x034D

5 Functions

5.1 Read n bits

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

Data request

Slave address	Function 0x01 or 0x02	Address first bit	Bit number	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

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

Example

Read the status of the 4 logic inputs (process data)
⇨ Section 8.2 “Process data”

Bit address = (base address + process data address) * 16 + bit number
= (0x002F + 0x0000) * 0x10 + 0x08 = 0x02F8

Data request: (CRC16 = 0xFBBC)

0A	01	02F8	0004	BCFB

Response: (CRC16 = A813)

0A	01	01	0F	13A8

In every case, at least 8 bits (1 byte) are read, irrespective of the number of bits to be read, since the response is made in bytes.

In the example above, this means that the bits 0x02F8—0x02FF are read.

0x02FF	0x02FE	0x02FD	0x02FC	0x02FB	0x02FA	0x02F9	0x02F8

8 bits = 1 byte

For all irrelevant bits (0x02FC—0x02FF), the response is the value 0.

5.2 Read n words

The functions reads n words from a defined address.

Data request

Slave address	Function 0x03 or 0x04	Address first word	Word number	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

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

Example

Read the first 3 measurement inputs
⇨ Section 8.2 “Process data”

Word address= base address + process data address
= 0x0035 + 0x0000 = 0x0035

Data request: (CRC16 = 03D7)

14	03	0035	0006	D703

Response: (CRC16 = 4750)

14	03	0C	1999	4348	4CCC	4348	2666	4396	5047
			Measurement 1 200.1		Measurement 2 200.3		Measurement 3 300.3		

5.3 Write one bit

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

Instruction

Slave address	Function 0x05	Bit address	Bit value XX 00	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave address	Function 0x05	Bit address	Bit value XX 00	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

☞ For the bit value,
the following applies: FF00 = set bit
0000 = erase bit

Example

Set the Modbus flag (bit 0) below the base address 0x002F
⇨ Section 8.2 “Process data”
⇨ Section 9.1 “Modbus flag”

Bit address = (base address + address “Modbus flag”) * 16 + bit number
= (0x002F + 0x0004) * 0x10 + 0x0
= 0x0330

Instruction: (CRC16 = B48E)

14	05	0330	FF00	8EB4

Response (as instruction):

14	05	0330	FF00	8EB4

5.4 Write one word

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

Instruction

Slave address	Function 0x06	Word address	Word value	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave address	Function 0x06	Word address	Word value	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example

Set the Modbus flag (bit 0 to address 0x0033)
⇨ Section 8.2 “Process data”
⇨ Section 9.1 “Modbus flag”

Word address= base address + address "Modbus flag"
= 0x002F + 0x0004 = 0x0033

Instruction: (CRC16 = C0BA)

14	06	0033	0001	BAC0

Response (as instruction):

14	06	0033	0001	BAC0

5.5 Write n words

Instruction

Slave address	Function 0x10	Address first word	Word number	Byte number	Word value(s)	Check sum CRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	x byte(s)	2 bytes

Response

Slave address	Function 0x10	Address first word	Word number	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example

Set the Modbus flag (bit 0 to address 0x0033)
⇨ Section 8.2 “Process data”
⇨ Section 9.1 “Modbus flag”

Word address= base address + address “Modbus flag”
= 0x002F + 0x0004 = 0x0033

Instruction: (CRC16 = C390)

14	10	0033	0001	02	0001	90C3

Response: (CRC16 = 03F3)

14	10	0033	0001	F303

6.1 Transmission format

Integer values

Integer values are transmitted over the MODbus in the following format:
first the high byte, then the low byte.

e. g.: Polling the int-value of address 0x0000, if the value 18 (0x0012) is written below this address.
Request: 010300000001840A (CRC16 = 0x0A84)
Response: 010302**0012**3849 (CRC16 = 4938)

Float values

Float values are handled on the MODbus with the IEEE-754 standard format (32 bits), but with the difference that bytes 1 and 2 are swapped with bytes 3 and 4.

Single-float format (32 bits) to standard IEEE 754

SEEEEEEE	EMMMMMMM	MMMMMMMM	MMMMMMMM
byte 1	byte 2	byte 3	byte 4

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

MODbus-float format

MODbus address x		MODbus address x+1	
MMMMMMMM	MMMMMMMM	SEEEEEEE	EMMMMMMM
byte 3	byte 4	byte 1	byte 2

e. g.: Polling the float value of address 0x0035, if the value 550.0 (0x44098000 in IEEE-754 format) is written below this address.
Request: 010300350002D405 (CRC16 = 05D4)
Response: 010304**80004409**20F5 (CRC16 = F520)

After transmission from the instrument, the bytes of the float value have to be swapped accordingly.

Many compilers (e. g. Microsoft C++, Turbo C++, Turbo Pascal, Keil C51) record the float values in the following order:

float value

Address x	Address x+1	Address x+2	Address x+3
MMMMMMMM	MMMMMMMM	EMMMMMMM	SEEEEEEE
byte 4	byte 3	byte 2	byte 1

Please find out how float values are saved in your application. If necessary, the bytes have to be swapped accordingly in the interface program, after acquiring them from the paperless recorder.

Double values

Double values are also handled on the MODbus in the IEEE-754 standard format (32 bits), but, unlike float values, no bytes are swapped.

Double-float format (32 bits) to standard IEEE 754

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 (complement to base 2)
M - 52 bit normalized mantissa

MODbus-double format

MODbus address x		MODbus address x+1	
SEEEEEEE	EEEEMMMM	MMMMMMMM	MMMMMMMM
byte 1	byte 2	byte 3	byte 4

MODbus address x+2		MODbus address x+3	
MMMMMMMM	MMMMMMMM	MMMMMMMM	MMMMMMMM
byte 5	byte 6	byte 7	byte 8

e. g.: Polling the double value of address 0x0066, if the value 1234567.89 (0x4132D687E3D70A3D in IEEE-754-Format) is written below this address.
Request: 010300660004A416 (CRC16 = 16A4)
Response: 010308**4132D687E3D70A3D**A4CD (CRC16 = CDA4)

Please find out how double values are saved in your application. If necessary, the bytes have to be swapped accordingly in your interface program, after acquiring them from the paperless recorder.

Texts

Texts are transmitted in the ASCII format.

☞ The last sign transmitted must always be a “\0” (ASCII code 0x00) as a stop marker.

Since texts are also transmitted word by word (16 bits), an additional 0x00 is transmitted when there is an uneven number of characters (including “\0”).

e.g.: Polling text of address 0x0007, if the character string “133.01.01” (ASCII code: 0x31, 0x33, 0x33, 0x2E, 0x30, 0x31, 0x2E, 0x30, 0x31, 0x20, 0x00) is below this address.

Request: 0103000700067409
(CRC = 0974)
Response: 01030C**3133332E30312E3031200000**914D
(CRC16= 4D91)

7.1 Error handling

Error codes

If the data request by the master was received by the paperless recorder (without transmission error), but could not be processed, the recorder responds with an error code.
Three error codes are available:

1	invalid function
2	invalid parameter address
3	data value outside the permissible range

If a bit or word number that is larger than the maximum permitted is read by the master, the recorder also sends error code 2.

Response on error

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

The function code is linked by OR with 0x80, i. e. the MSB (most significant bit) is set to 1.

Example

Data request: (CRC16 = 0B1C)

01	09	0000	0001	1C0B

Response: (CRC16 = 5086)

01	89	01	8650

Special cases

The slave does not respond to the following errors:

- the baud rate and/or data format for master and paperless recorder do not match
- the instrument address of the paperless recorder does not correspond to the one in the protocol (in this case, the data request by the master has to be sent again after a time-out of 2 sec)
- the checksum (CRC16) is not correct
- the instruction by the master is incomplete or over-defined
- the number of words or bits to be read is zero
- there is an ongoing communication via the setup interface

7.2 Error messages for invalid values

For measurements, the convention is that the error number is represented in the value itself, i. e. the error number is entered instead of the measured value.

For float values

Error number	Error
-200000.0	underrange
200000.0	overrange
200003.0	other invalid value

For double values

Error number	Error
-90000000000000000.0	underrange
90000000000000000.0	overrange
80000000000000003.0	other invalid value

Example

Data request: (CRC16 = 05D4)

01	03	0035	0002	D405

Response: (CRC16 = C29C)

01	03	04	5000	4843	9CC2

The measurement (0x48435000 = 200000.0) supplied by analog input 1 indicates an overrange condition.

All process values (variables) together with their addresses, the data type and the access mode are described below.

References are as follows:

R/O	read 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; xx = length including string stop character “\0”
bit x	bit No. x
float	float value (4 bytes)
double	float value (8 bytes)

The process values are divided into logical areas.

The absolute MODbus address is given by the base address of the appropriate area and the address offset.

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

8.1 Instrument data

Base address: 0x0000

Address	Access	Data type	Signal designation
0x0000	R/O	int	instrument group (18)
0x0001	R/O	int	instrument type (0)
0x0002	R/O	char 9	instrument name (“LS500”)
0x0007	R/O	char 11	software version
0x000D	R/O	char 13	VdN number
0x0014	R/O	char 10	serial number
0x0019	R/O	char 15	date/time of last alteration configuration
0x0021	R/O	char 15	date/time of last alteration configuration

8.2 Process data

Base address: 0x002F

Address	Access	Data type	Signal designation
0x0000	R/O	int	status of the logic inputs
	R/O	bit0-7	not used
	R/O	bit8	logic input 1 0 = open / 1 = closed
	R/O	bit9	logic input 2
	R/O	bit10	logic input 3
	R/O	bit11	logic input 4
	R/O	bit12-15	not used
0x0001	R/O	int	other logic signals
	R/O	bit0-7	not used
	R/O	bit8	combination alarm 0 = no alarm 1 = at least 1 limit infringed in instrument
	R/O	bit9	disk reserve signal 0 = disk reserve not yet reached 1 = change diskette
	R/O	bit10	error 0 = no error / 1 = error
	R/O	bit11	low combination alarm 0 = no low alarm 1 = at least 1 low alarm present
	R/O	bit12	high combination alarm 0 = no high alarm 1 = at least 1 high alarm present
	R/O	bit13	counter/integrator combination alarm 0 = no alarm 1 = at least 1 counter/int. limit infringed
	R/O	bit14-15	not used
0x0002	R/O	int	relay outputs and logic channels
	R/O	bit0	relay output 1 0 = not active / 1 = active
	R/O	bit1	relay output 2
	R/O	bit2	relay output 3

Address	Access	Data type	Signal designation
	R/O	bit3-7	not used
	R/O	bit8	logic channel 1 0 = false / 1 = true
	R/O	bit9	logic channel 2
	R/O	bit10	logic channel 3
	R/O	bit11	logic channel 4
	R/O	bit12	logic channel 5
	R/O	bit13	logic channel 6
	R/O	bit14-15	not used
0x0003	R/O	int	counter/integrator alarms
	R/O	bit0-7	not used
	R/O	bit8	alarm counter/integrator channel 1 0 = no alarm / 1 = limit infringed
	R/O	bit9	alarm counter/integrator channel 2
	R/O	bit10	alarm counter/integrator channel 3
	R/O	bit11	alarm counter/integrator channel 4
	R/O	bit12	alarm counter/integrator channel 5
	R/O	bit13	alarm counter/integrator channel 6
	R/O	bit14-15	not used
0x0004	R/W	int	flag for operating different instrument functions
	R/W	bit0	Modbus flag 0 = false / 1 = true
	R/W	bit1-15	not used
0x0005	R/O	int	alarms
	R/O	bit0	low alarm channel 1 0 = no alarm 1 = below limit
	R/O	bit1	low alarm channel 2
	R/O	bit2	low alarm channel 3
	R/O	bit3	low alarm channel 4
	R/O	bit4	low alarm channel 5
	R/O	bit5	low alarm channel 6

Address	Access	Data type	Signal designation
	R/O	bit6-7	not used
	R/O	bit8	high alarm channel 1 0 = no alarm 1 = limit exceeded
	R/O	bit9	high alarm channel 2
	R/O	bit10	high alarm channel 3
	R/O	bit11	high alarm channel 4
	R/O	bit12	high alarm channel 5
	R/O	bit13	high alarm channel 6
	R/O	bit14-15	not used

Base address: 0x0035

Address	Access	Data type	Signal designation
0x0000	R/O	float	measurement input 1 (analog input 1)
0x0002	R/O	float	measurement input 2 (analog input 2)
0x0004	R/O	float	measurement input 3 (analog input 3)
0x0006	R/O	float	measurement input 4 (analog input 4)
0x0008	R/O	float	measurement input 5 (analog input 5)
0x000A	R/O	float	measurement input 6 (analog input 6)
0x000C	R/O	float	counter/integrator channel 1 [1]
0x000E	R/O	float	counter/integrator channel 2 [1]
0x0010	R/O	float	counter/integrator channel 3 [1]
0x0012	R/O	float	counter/integrator channel 4 [1]
0x0014	R/O	float	counter/integrator channel 5 [1]
0x0016	R/O	float	counter/integrator channel 6 [1]

1. In the paperless recorder, these are double values (8 bytes). Since only float values (4 bytes) can be read for the address, only a restricted resolution is possible (restriction of the count range).
   The values can be read in double format, under the base address 0x0066.

Base address: 0x0066

Address	Access	Data type	Signal designation
0x0000	R/O	double	counter/integrator channel 1
0x0004	R/O	double	counter/integrator channel 2
0x0008	R/O	double	counter/integrator channel 3
0x000C	R/O	double	counter/integrator channel 4
0x0010	R/O	double	counter/integrator channel 5
0x0014	R/O	double	counter/integrator channel 6

Base addresse: 0x7000

Address	Access	Data type	Signal designation
0x0007	W/O	int	password for polling the current and stored measurement data
0x0008	R/O	int	info flag, whether readout of measurement data is locked by password: 0 = measurement data can be read out 1 = entry of correct password required

9.1 Modbus flag

The Modbus flag, like other logic signals, (e.g. logic inputs or alarms) can be used to operate different recorder functions. In order to be able to use the Modbus flag, the entry “Modbus flag” has to be selected when configuring.

A conceivable application for the Modbus flag is, for example, the activation of event operation via the serial interface.

A

B

C

D

E

F

I

J

W

10 Index

M. K. JUCHHEIM GmbH & Co

Street address:
Moltkestraße 13 - 31
36039 Fulda, Germany
Delivery address:
Mackenrodtstraße 14
36039 Fulda, Germany
Postal address:
36035 Fulda, Germany
Phone: +49 661 6003-0
Fax: +49 661 6003-607
E-mail: mail@jumo.net
Internet: www.jumo.net

JUMO Instrument Co. Ltd.

JUMO House
Temple Bank, Riverway
Harlow, Essex CM20 2TT, UK
Phone: +44 1279 635533
Fax: +44 1279 635262
E-mail: sales@jumo.co.uk
Internet: www.jumo.co.uk

JUMO PROCESS CONTROL INC.

885 Fox Chase, Suite 103
Coatesville, PA 19320, USA
Phone: 610-380-8002
1-800-554-JUMO
Fax: 610-380-8009
E-mail: info@JumoUSA.com
Internet: www.JumoUSA.com