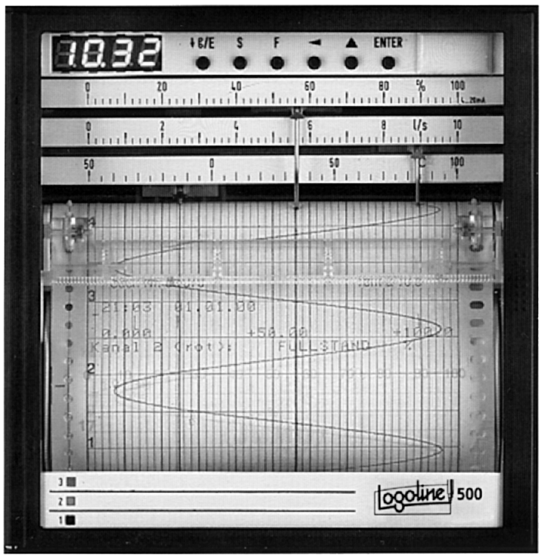
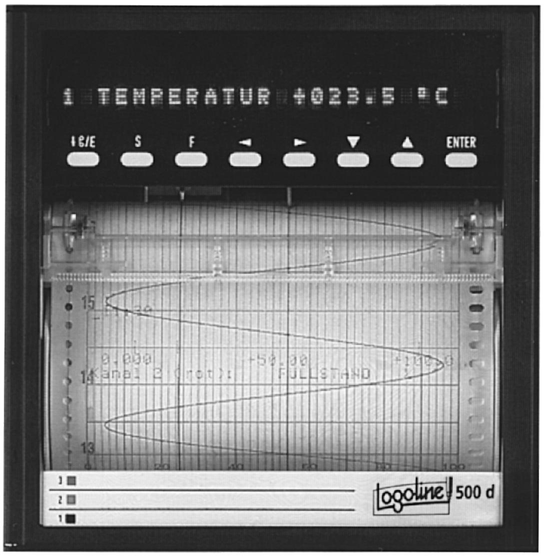




logoline 500

B 95.3530.2

Interface description

12.01/00340396

Contents

Contents

6 Address tables

1 Introduction

1.1 Preface

Please read this manual before starting up the interface. Keep the manual in a place which is accessible to all users at all times.

Please assist us to improve this manual.

Your suggestions will be most welcome.

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

All the information which is necessary to operate the interface is included in this manual. If any difficulties should still arise during start-up you are asked not to carry out any manipulations on the unit which are not permitted. You could endanger your rights under the instrument warranty!

Please contact the nearest JUMO office or the main factory.

When returning instrument modules, assemblies or components, the rules of EN 100 015 "Protection of electrostatic-sensitive devices" must be followed. Use only the appropriate ESD packaging to transport them.

Please note that we cannot be held liable for any damage caused by **ESD** (**e**lectro**s**tatic **d**ischarge).

1.2 Typographical conventions

1.2.1 Warning signs

The signs for Danger and Warning are used in this manual under the following conditions:

Danger — This symbol is used where there may be danger to personnel if the instructions are disregarded or not followed accurately!

Warning — This symbol is used where there may be damage to equipment or data if the instructions are disregarded or not followed accurately!

Warning — This symbol is used where special precautions must be observed for the handling of electrostatic-sensitive components.

1.2.2 Note signs

Note — This symbol is used if your special attention is drawn to a remark.

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

abc[1]	**Footnote**	Footnotes are comments which refer to certain points in the text. Footnotes consist of two parts: The marking in the text, and the footnote text. Markings in the text are arranged as continuous raised numbers. The footnote text (in a smaller typeface) is placed at the bottom of the page, and begins with a number and a full stop.

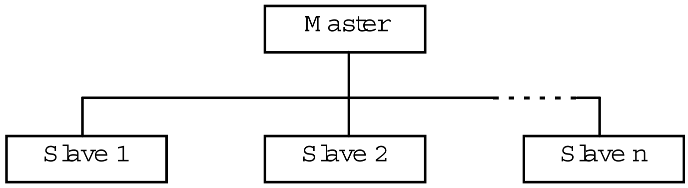
1.2.3 Presentation

0x0010	**Hex. number**	A hexadecimal number is identified by a preceding “0x” (this example = decimal 16).

2 Protocol description

2.1 Master-slave principle

The communication between a PC (master) and an instrument (slave) using MODbus/Jbus 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 instrument address. A maximum of 255 slaves can be accessed.

2.2 Transfer mode (RTU)

The transfer mode which is used is the RTU mode (Remote Terminal Unit). The transfer of the data is made in binary format (hexadecimal) with 8 bits, 16 bits for integers, and 32 bits for float values. The LSB (least significant bit) is transferred first. The ASCII operating mode is not supported.

Data format

The data format describes the structure of a byte which is transferred. The following format options are available for data transfer:

Data word	Parity bit	Stop bit 1 or 2 bits	No. of bits
8 bits	—	1	9
8 bits	—	2	10
8 bits	even	1	10
8 bits	odd	1	10
8 bits	always 0	1	10

2.3 Instrument address

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

☞ A maximum of 31 slaves can be accessed via the RS422/ RS485 interface.

2.4 Timing of the communication

The start and end of a data block are identified by pauses in transmission. The maximum permitted interval between two consecutive characters is three times the transmission time of a single character.

The character transmission time (the time taken to transmit one character) depends on the baud rate and the data format which is used.

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

character transmission time [msec] = 1000 •9 bits/(baud rate)

For the other data formats it is:

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

Sequence

Date request from master transmission time = n characters •1000 •x bi ts / (baud rate)
Marker for end of data request 3 characters •1000 • x bits/(baud rate)
Processing of the data request by the slave (max. 250msec)
Response of slave transmission time = n characters •1000 •x bits/(baud rate)
Marker for end of response 3 characters •1000 • x bits/(baud rate)

Example

Marker for end of data request or end of response for 10/9 bit data format

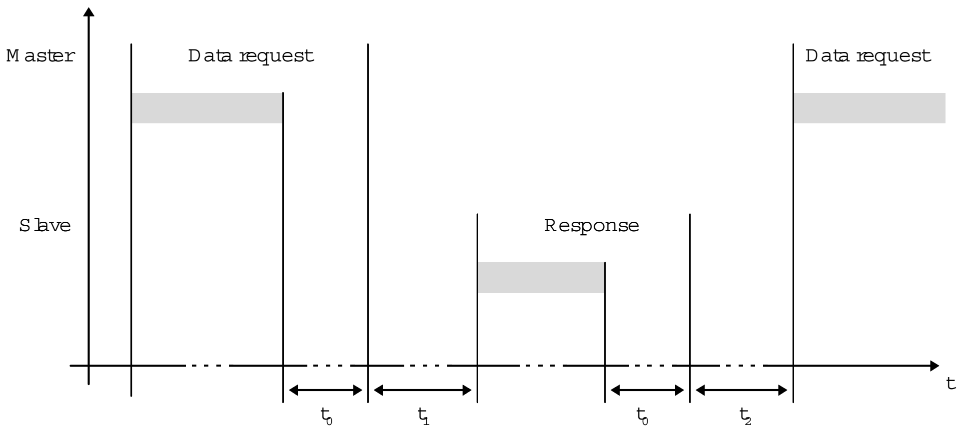
Waiting time = 3 characters • 1000 • 10 bits/(baud rate)

Baud rate	**Data format** [bit]	**Waiting time** [msec]
187k	10	0.160
	9	0.144
125k	10	0.240
	9	0.216
38400	10	0.781
	9	0.703
19200	10	1.563
	9	1.406
9600	10	3.125
	9	2.813
4800	10	6.250
	9	5.625
2400	10	12.500
	9	11.250
1200	10	25.000
	9	22.500
600	10	50.000
	9	45.000
300	10	100.000
	9	90.000
150	10	200.000
	9	180.000

2.4.1 Timing sequence of a data request

Timing scheme

A data request runs according to the following timing scheme:



t_0 End marker = 3 characters
(the time depends on the baud rate)

t_1 This time depends on the internal processing.
The maximum processing time is 250 msec.

☞ A minimum response time can be set in the instrument, under the menu item "Interface". This preset time (0—500 msec) is the minimum time which will be waited before a response is transmitted. If a smaller value is set, then the response time may be longer than the preset value (because 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, in order to switch over the interface driver from receive to transmit. This parameter is not required by the RS422 interface.

t_2 This time is needed by the instrument, to switch from transmit back to receive. The master must wait for this time before presenting a new data request. This time must always be observed, even when the new data request is directed to a different instrument.

RS422 interface: t_2 = 1 msec
RS485 interface: t_2 = 10 msec

2.4.2 Communication during the internal processing time of the slave

No data requests from the master can be permitted during the internal processing time of the slave. Any data requests which are made in this period will be ignored by the slave.

2.4.3 Communication during the response time of the slave

No data requests from the master can be permitted during the response time of the slave. Any data request which is made in this period will result in the invalidation of all the data on the bus at that moment.

2.5 Structure of the data blocks

All data blocks have the same structure:

Data structure

Slave address	Function code	Data field	Checksum 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)

Checksum recognition of transmission errors

2.6 Error handling

Error codes There are five error codes:

1	invalid function
2	invalid parameter address
3	parameter value outside range of values
4	slave not ready
8	write access to parameter is denied

Response on error event

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, in other words, the MSB (most significant bit) is set to 1.

Example

Data request:

01	02	00	00	00	00	CRC16

Response:

01	82	01	CRC16

Special cases

The slave will not respond in the following cases:

- the checksum (CRC16) is not correct
- the instruction from the master is incomplete or over-defined
- the number of words or bits to be read is zero
- the instrument is being configured from the keypad

2.7 Differences between MODbus and 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 MODbus addresses are shifted by one compared with those of Jbus.

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

2.8 Checksum (CRC16)

The checksum (CRC16) is used to recognise transmission errors. If an error is identified in the evaluation, the corresponding instrument will not respond.

Method of calculation

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

Data request: Read two words, starting at address 1 (CRC16 = 0x0E97)

14	03	00	01	00	02	97	0E

Response: (CRC16 = 0x953E)

14	03	04	03	E8	01	F4	3E	95
			Word 1		Word 2			

Example 2

Instruction: Set bit at address 24 (CRC16 = 0xF80E)

14	05	00	18	FF	00	0E	F8

Response (same as instruction):

14	05	00	18	FF	00	0E	F8

2.9 Interface

Configuration of the interface in the instrument.

Display	**edit**	**select/enter**	**with key**	**accept**	**continue with**
INTERFACE	ENTER	PROTOCOL: J-BUS Select protocol: JBUS MODBUS	▲, ▼	ENTER	⇒ 2
	⇒ 2	BAUD: 9.6 kbaud Select baud rate (values in kbaud): 0.15, 0.3, 1.2, 2.4, 4.8, 9.6, 19.2, 38.4, 12.5, 187.5 kbaud	▲, ▼	ENTER	⇒ 3
	⇒ 3	DATA FORMAT: 8/1/NONE Select data format (data bits / stop bits / parity): 8/1/NONE, 8/1/ODD, 8/1/EVEN, 8/2 NONE, 8/1/ZERO	▲, ▼	ENTER	⇒ 4
	⇒ 4	ADDRESS: 001 Select address: 1 ... 255	▲, ▼, ◀, ▶	ENTER check: 1 ≤ address ≤ 255	⇒ 5
	⇒ 5	MIN. RESPONSE TIME: 000 msec select minimum response time: 0 ... 999 msec	▲, ▼, ◀, ▶	ENTER check: 0 ≤ response time ≤ 999	▲ forwards ▼ backwards

☞ In the Logoline 500 version with a scale, the interface is configured through the setup program.

3 Functions

The following functions are available for the instrument:

Function number	Function	
0x01/0x02	read n bits	(max. 256 bits)
0x03/0x04	read n words	(max. 80 words)
0x05	write one bit	
0x06	write one word	
0x0F	write n bits	(max. 256 bits)
0x10	write n words	(max. 80 words)

There are no separate areas available for bits and words of the system variables. The bit and word areas overlap, and can be read from and written to both as bit areas and word areas.

The bit address is calculated as follows:

bit address = word address •16 + bit number

3.1 Read n bits

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

Data request

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

Response

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

Example

Read the position of the first 4 binary inputs (process data),
⇨ Section 6.1.2 "Process data"

Bit address = (base address + process data address) •16
= (0x002F + 0x0008) •0x10
= 0x0370

Data request

0A	01	03	70	00	04	CRC16

Response

0A	01	01	0F	CRC16

☞ Regardless of the number of bits to be read, at least 8 bits (1 byte) are always read, since the response is made in bytes.

In the example above, this means that the bits 0x0370.0x0377 will be read.

0x0377	0x0376	0x0375	0x0374	0x0373	0x0372	0x0371	0x0370

8 bits = 1 byte

For all the irrelevant bits (0x0374.0x0377) the re sponse will be the value 0.

3.2 Read n words

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

Data request

Slave address	Function 0x03 or 0x04	Address of first word	No. of words	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave address	Function 0x03 or 0x04	No. of bytes read	Word value(s)	Checksum CRC16
1 byte	1 byte	1 byte	x byte(s)	2 bytes

Example

Read the 3 measurement inputs

⇨ Section 6.1.2 "Process data"

Word address = base address + process data address
= 0x002F + 0x0002
= 0x0031

Data request:

14	03	00	31	00	06	CRC16

Response:

14	03	10	1999	4348	4CCC	4348	2666	4396	CRC16
			Meas. 1 200.1		Measurement 2 200.3		Meas. 3 300.3		

3.3 Write one bit

In the bit writing function, the data blocks for the instruction and the 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

☞ The bit values mean: FF00 = set bit
0000 = erase bit

Example

Set the status bit 0 of the data block “Text for printing”
⇨ Section 6.2.1 “Text for printing”

Bit address = (base address + address ‘Data structure status’) •16 + bit number
= (0x0075 + 0x0) •0x10 + 0x0
= 0x0750

Instruction:

14	05	07	50	FF	00	CRC16

Response (same as instruction):

14	05	07	50	FF	00	CRC16

3.4 Write one word

In the word writing function, the data blocks for the instruction and the 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

Write flag 1 for the display controller (= 0x0001)
⇨ Section 6.2.3 "Flags for the display controller"

Word address = base address + address flag 1
= 0x00E9 + 0x0002
= 0x00EB

Instruction:

14	06	00	EB	00	01	CRC16

Response (same as instruction):

14	06	00	EB	00	01	CRC16

3.5 Write n bits

Instruction

Slave address	Function 0x0F	Address of first bit	Number of bits	Number of bytes	Bit value(s)	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	x byte(s)	2 bytes

Response

Slave address	Function 0x15	Address of first bit	Number of bits	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example

Set the status bit 0 and bit 1 of the data block “Text for printing”

Status bit 0 = 1, Status bit 1 = 0
⇨ Section 6.2.1 “Text for printing”

Bit address = (base address + address ‘Data structure status’) •16 + bit number
= (0x0075 + 0x0) •0x10 + 0x0
= 0x0750

Instruction:

14	0F	07	50	00	02	01	01	CRC16

Response:

14	0F	07	50	00	02	CRC16

3.6 Write n words

Instruction

Slave address	Function 0x10	Address of first word	Number of words	Number of bytes	Word value(s)	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	x byte(s)	2 bytes

Response

Slave address	Function 16	Address of first word	Number of words	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example

Write “Text for printing”
(2 words: “ABC” = 0x4142, 0x4300)
⇨ Section 6.2.1 “Text for printing”

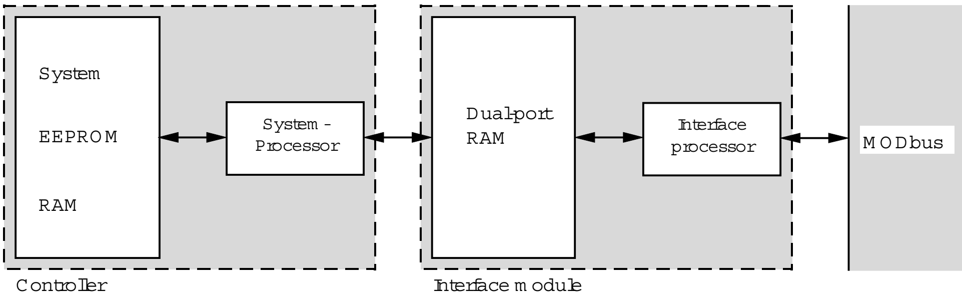
Word address = base address + process data address
= 0x0075 + 0x0002
= 0x0077

Instruction:

14	10	00	77	00	02	04	41	42	43	00	CRC16

Response:

14	10	00	77	00	02	CRC16



For data transmission to the MODbus, the system processor places the process values in a dual-port RAM. Not all the system variables which are present in the controlller are cyclically refreshed in the dual-port RAM. The dual-port RAM is divided into two areas:

System variables

These variables can be read and written directly by the MODbus driver (cyclic data).

The data are cyclically refreshed in the dual-port RAM (within the sampling time).

Data after data request

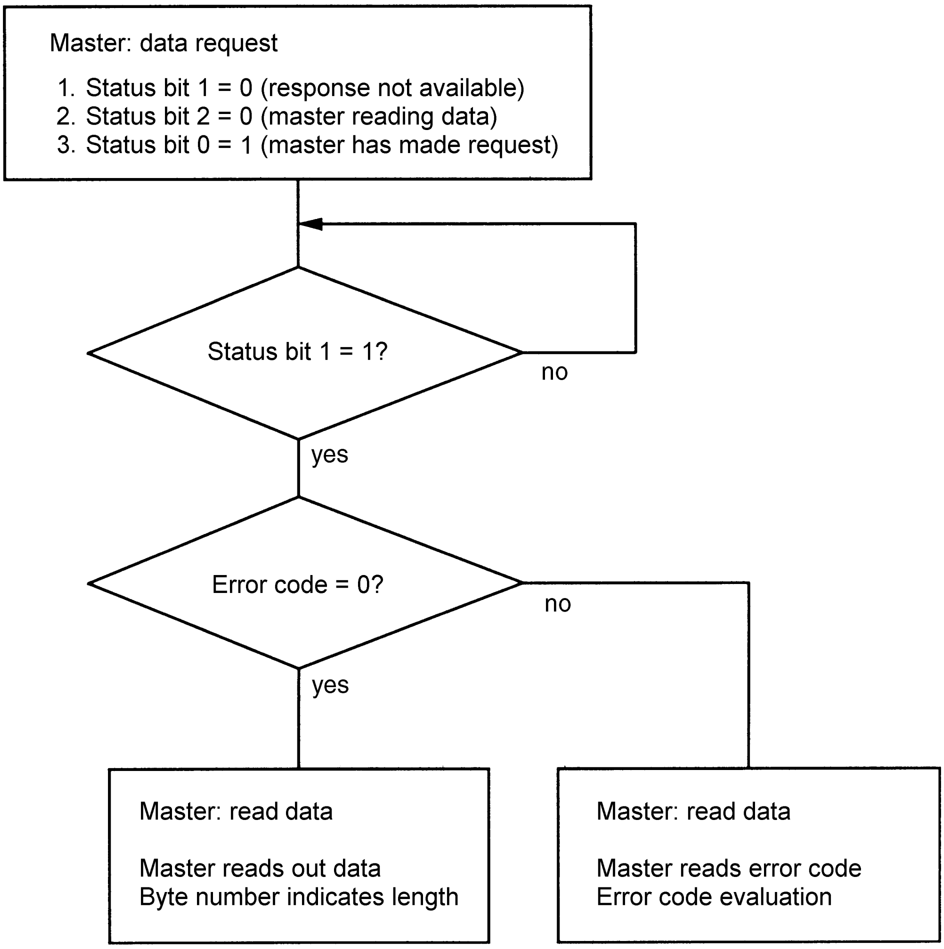
This area is not cyclically refreshed by the system processor (non-cyclic data).

Variables in this data area must be requested by the MODbus driver.

They are available only after processing by the system processor.

☞ The length information for *char* data types as given below is generally to be understood as being the string length *including* the string termination characters “\0”

4.1 Receive non-cyclic data from the instrument



Example Read the reserve flag for the display controller
⇨ Section 6.2.3 "Flags for the display controller"

Step 1: The data structure "Flags for the display controller" is requested

Set status bit 0 = 1, status bit 1 = 0 and status bit 2 = 0

MODbus command: Write 1 word

01	06	00	E9	00	01	CRC16

Response:

01	06	00	E9	00	01	CRC16

Step 2: Cyclic request (polling) whether the corresponding data structure is available

Read status bit 1

MODbus command: Read n bits

01	01	0E	91	00	01	CRC16

Response:

01	01	01	00	CRC16

Status bit 1 = 0 (data structure is not yet available)

01	01	01	01	CRC16

Status bit 1 = 1 (data structure is available)

Step 3: Read error code of the structure which is requested

MODbus command: Read 1 word

01	03	00	EA	00	01	CRC16

Response:

01	03	02	0000	CRC16

No error occurred

Step 4: Read reserve flag

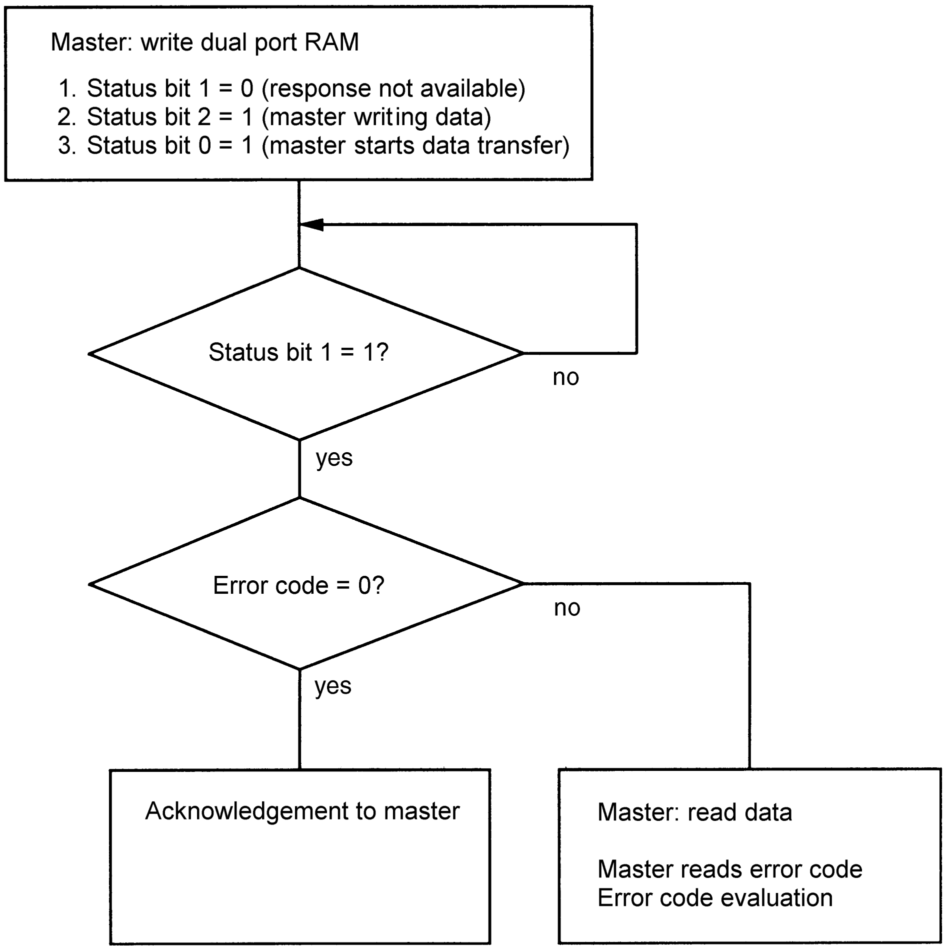
MODbus command: Read 1 word

01	03	00	EE	00	01	CRC16

Response:

01	03	02	0001	CRC16
			reserve flag is set	

4.2 Transmit non-cyclic data to the instrument



Example Write a float value for the maths module
(float-value 1 = 20.32)

Step 1: The data structure is requested

Set status: bit 0 = 1, bit 1 = 0, bit 2 = 0

MODbus command: Write 1 word

01	06	00F9	0001	CRC16

Response:

01	06	00F9	0005	CRC16

Step 2: Cyclic request (polling) whether the corresponding data structure is available

Read status bit 1

MODbus command: Read n bits

01	01	0F	91	00	01	CRC16

Response:

01	01	01	00	CRC16

Status bit 1 = 0 (data structure is not yet available)

01	01	01	01	CRC16

Status bit 1 = 1 (data structure is available)

Step 3: Read error code of the structure which is requested

MODbus command: Read 1 word

01	03	00FA	0001	CRC16

Response:

01	03	02	0000	CRC16

No error occurred

Step 4: Write 20.32 to float value 1
(20.32 is equivalent to 0x41A28F5C in IEEE-format)

MODbus command: Write 2 words

01	10	00FB	0002	04	8F5C	41A2	CRC16

Response:

01	06	00FB	0002	CRC16

Step 5: The data structure is transmitted

Set status: bit 0 = 1, bit 1 = 0, bit 2 = 1

MODbus command: Write 1 word

01	06	00F9	0005	CRC16

Response:

01	06	00F9	0005	CRC16

Step 6: Cyclic polling whether the corresponding data structure is available

Read status bit 1

MODbus command: Read n bits

01	01	00F9	00	01	CRC16

Response:

01	01	01	00	CRC16

Status bit 1 = 0 (data structure not yet transmitted)

01	01	01	01	CRC16

Status bit 1 = 1 (data structure has been transmitted)

Step 7: Read the error code of the transmitted structure

MODbus command: Read 1 word

01	03	00FA	0002	CRC16

Response:

01	03	02	0000	CRC16

No error occurred

4.3 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. : Write the integer value 1 (= 0x0001) to the address 0x00EB:
010600EB**0001**383E

Float values

Float values are handled on the MODbus with 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) 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 normalised 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. : Write the float value 550.0
(= 0x44098000 in IEEE-754 format) to the address 0x00FB:
011000FB00020480004409679E

The bytes of the float value must be appropriately swapped before/after the transmission to/from the instrument.

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

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

5 Error messages

5.1 Error messages from the interface

The error numbers can be found under the error code in the data blocks of the non-cyclic data.

⇨ Section 6.2.1 –Section 6.2.4

Error code	**Error: Setup command processing**
0x0014	Command busy flag not reset by master
0x0015	Invalid command
0x0016	Error on data acceptance
0x0017	No cyclic data available
0x0018	Invalid structure length
0x0019	Invalid header
0x001C	Write error in the serial EEPROM (Calib.)

5.2 Error messages for invalid values

For setpoints/process values and values which are derived from them, the convention is that the error number is represented in the value itself, i.e. the error number is recorded instead of the measurement.

Error number	Error
200000.0	Measurement overrange/underrange

5.3 System and running errors

The system or running errors are part of the process data (cyclic data).

⇨ Section 6.1.2

The numbers which are recorded there have the following meaning:

System error

If one of the following errors occurs, then the measurement acquisition and recording is interrupted. All other process data are no longer valid. If error number 13 occurs, then the instrument is still in the initialisation phase, during which valid process data are not yet available. The instruction to the instrument should be repeated after a short delay.

Error number	Error
0	no error
1	reserved
2	reserved
3	reserved
4	EEPROM fault
5	reserved
6	reserved
7	reserved
8	A/D converter fault
9	reserved
10	Hall sensor fault
11	reserved
12	reserved
13	initialisation phase

Running errors

Please see the descriptions in the Operating Manual B 95.3530 for the response of the instrument to one or more of the following errors.

All other data can continue to be read out from the instrument.

Error number	Error
0	no error
1	reserved
2	reserved
3	no paper
4	relay module does not respond
5	clock must be reset
6	battery low/empty
7	reserved
8	reserved

☞ It is possible for several runing errors to occur at the same time. The error which is displayed is always the one with the highest priority.

Event	**Priority**
no paper	higher
relay module does not respond	↑
clock must be reset	↓
battery low	
no error	lower

For instance, if all the paper has been used up and the battery is low, then error no. 3 (no paper) will be signalled as a running error.

6 Address tables

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

The abbreviations used are:

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 termination characters “\0”
bit x	bit no. x
float	float value (4 byte)

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.

6 Address tables

6.1 Cyclic data

6.1.1 Instrument data

Base address: 0x0000

Address	Access	Data type	Signal designation
0x0000	R/O	int	instrument group (7)
0x0001	R/O	int	instrument type (0)
0x0002	R/O	char 9	instrument name ("L500/540")
0x0007	R/O	char 11	software version
0x000D	R/O	char 13	VdN number
0x0014	R/O	char 10	production number
0x0019	R/O	char 15	date/time of last change configuration
0x0021	R/O	char 15	date/time of last change parameter
0x0029	R/O	12 byte	reserve

6.1.2 Process data

Base address: 0x002F

Address	Access	Data type	Signal designation
0x0000	R/O	int	system error ⇨ Chapter 5 "Error messages"
0x0001	R/O	int	running error ⇨ Section 5.3 "System and running errors"
0x0002	R/O	float	measurement input 1
0x0004	R/O	float	measurement input 2
0x0006	R/O	float	measurement input 3
0x0008	R/O	int	state of the logic inputs: 0 = open / 1 = closed
	R/O	bit0	logic input 1
	R/O	bit1	logic input 2
	R/O	bit2	logic input 3
	R/O	bit3	logic input 4
	R/O	bit4	logic input 5

Address	Access	Data type	Signal designation
	R/O	bit5	logic input 6
	R/O	bit6	logic input 7
	R/O	bit7	logic input 8
	R/O	bit8.15	free
0x0009	R/O	int	relay states: 0 = not active / 1 = activated
	R/O	bit0	relay output 1
	R/O	bit1	relay output 2
	R/O	bit2	relay output 3
	R/O	bit3	relay output 4
	R/O	bit4	relay output 5
	R/O	bit5	relay output 6
	R/O	bit6	relay output 7
	R/O	bit7	relay output 8
	R/O	bit8.15	free
0x000A	R/O	float	event counter 1
0x000C	R/O	float	event counter 2
0x000E	R/O	int	present chart speed
0x000F	R/O	int	Stop status: 0 = no stop / 1 = stop
0x0010	R/O	int	paper end: 0 = paper available / 1 = no paper
0x0011	R/O	char 20	present date and time ("yyyy-mm-dd-hh:mm:ss")
0x001B	R/O	char 20	last switch-off of the recorder (date/time)
0x0025	R/O	char 20	last switch-on of the recorder (date/time)
0x002F	R/O	int	number of mains switch-off events
0x0030	R/O	int	number of operating hours
0x0031	R/O	int	polling flag for the text transmitted for printing: 0 = no print request 1 = print request for the text is still present

6.2 Non-cyclic data

Status bits

The following address tables (for non-cyclic data) are only read from or written to on request of the MODbus drivers. The state of these address tables (non-cyclic data) is shown in the status word. The status word and error code are each placed at the start of the data block for the non-cyclic data.

Status			**Significance for the master**
bit 0	bit 1	bit 2	
0	0	X	master has not made a data request
1	0	X	master has made a data request to the instrument
		X	the data request is being processed
0	1	X	processing finished, the response is available
		X	in the buffer, ready for the master
1	1	X	not valid
		0	data transmission from dual-port RAM to instrument
		1	data transmission from instr. to dual-port RAM
bit 3... bit 15 are not used			

6.2.1 Text for printing

Base address: 0x0075

Address	**Access**	**Data type**	**Signal designation**
0x0000	R/W	int	status of the data structure
	R/W	bit0	1 = process data request
	R/W	bit1	0 = response not yet present / 1 = processing ended, response is present in the buffer
	R/W	bit2	0 = transmission interface ➔ recorder 1 = transmission recorder ➔ interface
		bit3.15	free
0x0001	R/W	int	error code ⇨ Chapter 5 "Error Messages"
0x0002	R/W	char36	text for output to paper

6.2.2 Text for display

Base address: 0x009D

Address	Access	Data type	Signal designation
0x0000	R/W	int	status of the data structure
	R/W	bit0	1 = process data request
	R/W	bit1	0 = response not yet present 1 = processing ended, response is present in the buffer
	R/W	bit2	0 = transmission interface ➔ recorder 1 = transmission recorder ➔ interface
		bit3.15	free
0x0001	R/W	int	error code ⇨ Chapter 5 "Error messages"
0x0002	R/W	char 36	text 1 for display
0x0014	R/W	char 36	text 2 for display
0x0026	R/W	char 36	text 3 for display

6.2.3 Flags for the display controller

Base address: 0x00E9

Address	Access	Data type	Signal designation
0x0000	R/W	int	status of the data structure
	R/W	bit0	1 = processing data request
	R/W	bit1	0 = response not yet present 1 = processing ended, reponse is present in the buffer
	R/W	bit2	0 = transmission interface ➔ recorder 1 = transmission recorder ➔ interface
		bit3.15	free
0x0001	R/W	int	error code ⇨ Chapter 5 "Error messages"
0x0002	R/W	int	flag 1: 0 = do not display text 1 1 = display text 1

0x0003	R/W	int	flag 2: 0 = do not display text 2 1 = display text 2
0x0004	R/W	int	flag 3: 0 = do not display text 3 1 = display text 3
0x0005	R/W	int	reserve

6.2.4 float values for the maths module

Base address: 0x00F9

Address	Access	Data type	Signal designation
0x0000	R/W	int	status of the data structure
	R/W	bit0	1 = processing data request
	R/W	bit1	0 = response not yet present / 1 = processing ended, response present in the buffer
	R/W	bit2	0 = transmission interface ➔ recorder 1 = transmission recorder ➔ interface
		bit3.15	free
0x0001	R/W	int	error code ⇨ Chapter 5 “Error messages”
0x0002	R/W	float	float value 1 = freely usable in the maths module
0x0004	R/W	float	float value 2 = freely usable in the maths module
0x0006	R/W	float	float value 3 = freely usable in the maths module

6.2.5 Reset recognition

Base address: 0x0406

Address	Access	Data type	Signal designation
0x0000	R/W	int	reset recognition of the interface module

This storage address be used for a reset recognition of the interface module. A value other than zero must be entered and then cyclically polled. If a reset appears at the interface module of the recorder, then the byte will be erased (set to zero). The polling thus enables a reset to be recognised.

A reset of the interface module can have the following reasons:

- Mains off
- Configuration of the instrument from the keypad (if the codeword has been entered properly)
- Setup connector is plugged into the instrument.

M. K. JUCHHEIM GmbH & Co
36035 Fulda
Germany
Phone (int. +49) 661 60 03-0
Fax (int. +49) 661 60 03-607
Telex 49701 juf d
email JUMO_de@e-mail.com