JUMO Wtrans

Receiver T01.EC1/EC3

B 90.2931.2.0

Interface Description
Modbus

JUMO

11.07/00502573

Contents

Contents

1.1 Preface

This operating manual is addressed to the system manufacturer with adequate technical background and PC related knowledge.

Please read this operating manual prior to commissioning the instrument. Keep the manual in a place accessible to all users at all times. Your comments are appreciated and may assist us in improving this manual.

Warranty

All necessary settings are described in this operating manual. Should problems be encountered during commissioning, please refrain from carrying out any manipulations that are not described in the manual. Any such intervention will jeopardize your warranty rights. Please contact the nearest subsidiary or the head office.

Electrostatic discharge

When accessing the inner parts of the unit and returning plug-in modules, assemblies or components, please observe the regulations according to EN 61340-5-1 and EN 61340-5-2 „Protection of electrostatic sensitive devices“. Only use **ESD** packaging for transport.

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

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

1.2 Typographical conventions

1.2.1 Warning signs

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!

ESD

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.

Reference

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

Footnote

abc[1]

Footnotes are remarks that **refer to specific points** in the text. Footnotes consist of two parts:
A marker in the text and the foot note text itself.
The markers in the text are arranged as continuous superscript numbers.

1.2.3 Carry out action

Instruction to act

✱

This symbol indicates that an **action to be performed** is described. The individual steps are marked by this asterisk, e.g.:

✱ Start PLC software

✱ Click on hardware catalog

Vital text

This text contains important information, and it is vital that you read it before going any further.

String of commands

File ➔ save as

Small arrows between words indicate a **string of Commands** to be executed in succession.

1.2.4 Number types

Hexadecimal number

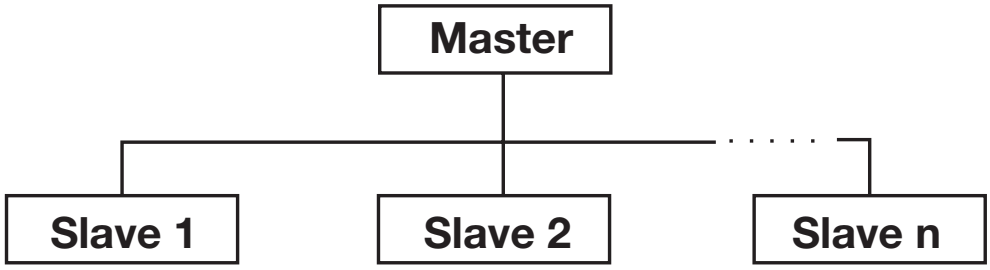
0x0010

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

1 Introduction

2.1 Master-Slave principle

Communication between a master (e.g. PC) and a slave (e.g. measuring and control system) using Modbus takes place according to the master-slave principle, in the form of data request/instruction - response.



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

The device described (receiver) described in this manual is only used as a Modbus slave.

2.2 Transmission mode (RTU)

The transmission mode used is the RTU mode (Remote Terminal Unit). The data is transmitted in the binary format (hexadecimal) with 8, 16 or 32 bits for integer values and 32 bits for float values. The ASCII operating mode is not supported.

Data format

The data format describes the structure of a character transmitted. The following data format options are available:

Data word	Parity bit	Stop bit 1/2 bit	Number of bits
8 bits	no	1	9
8 bits	even	1	10
8 bits	odd	1	10
8 bits	no	2	10

2.3 Timing of the communication

Start and end of a telegram are marked by transmission pauses. The maximum permitted interval between two consecutive characters is three times the transmission time required for a single character.

The character transmission time (time required to transmit one single character) depends on the baud rate and the data format used (stop bits and parity bit).

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

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

For the other data formats, this is:

character transmission time [ms]
= 1000 * (8 bits+parity bit+stop bit(s)) bit/baud rate

Example

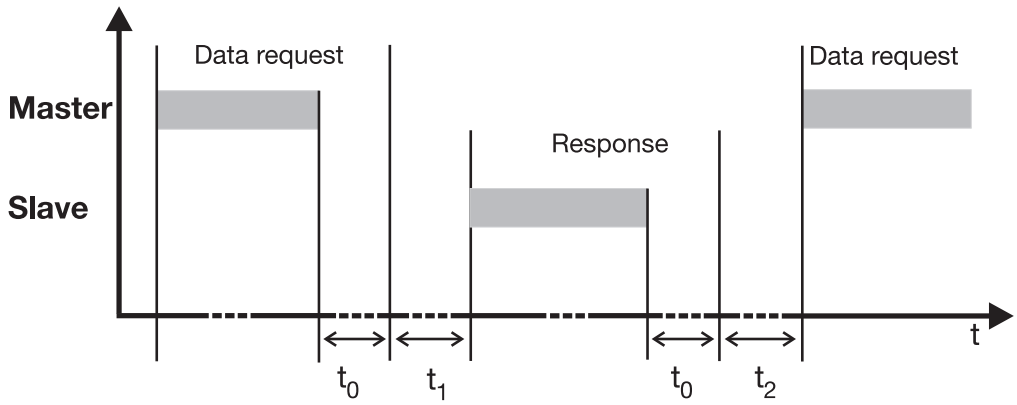
Baud rate [baud]	**Data format** [bit]	**Character transmission time** [ms]
38400	10	0,260
	9	0,234
19200	10	0,521
	9	0,469
9600	10	1,042
	9	0,938

Timing

Data request from master transmission time = n characters * 1000 * x bit/baud rate
Marker for end of data request 3 characters * 1000 * x bit/baud rate
Processing of data request by the slave (≤ 250ms)
Response of the slave transmission time = n characters * 1000 * x bit/baud rate
Marker for end of response 3 characters * 1000 * x bit/baud rate

Timing scheme

A data request runs according to the following timing scheme:



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

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

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

The minimum response time which can be set is required by the RS485 interface in the master, to be able to switch over the interface drivers from transmit to receive.

t_2 This time is needed by the slave to change from transmit back to receive.
The master has to observe this waiting time before presenting a new data request. This time must always be observed, even when the new data request is directed to another device.

RS485 interface: t_2 = 10ms

No data requests from the master are permitted during t_1 and t_2 and during the slave response time. Data requests made during t_1 are ignored by the slave; data requests during t_2 will result in the invalidation of all data currently on the bus.

2 Protocol description

2.4 Structure of a Modbus telegram

All telegrams have the same structure:

Data structure

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

Each telegram contains four fields:

Slave address device address of a specific slave

Function code function selection (read, write words)

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

- Word address or bit address
- Number of words or bits
- Word value(s) or bit value(s)

Checksum detection of transmission errors

2.5 Device address

The device address of the slave can be set between 1 and 254. The device address 0 is reserved for broadcast.

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

There are two different forms of data exchange:

Query

Data request/instruction by the master to a slave via the corresponding device address.

The slave addressed responds.

Broadcast

Instruction by the master to all slaves via the device address 0 (e.g. to transmit a specific value to all slaves).

The connected slaves do not respond. In such a case, the correct acceptance of the value by the slaves should be checked by a subsequent readout at each individual slave.

Data request with the device address 0 is meaningless.

2.6 Function codes

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

Function-overview

Function number	Function	Limitation
0x01 or 0x02	Read n bit	max. 256 bits (32 bytes)
0x03 or 0x04	Read n words	max. 80 words (160 bytes)
0x05	Write one bit	1 bit
0x06	Write one word	max. 1 word (2 bytes)
0x10	Write n words	max. 80 words (160 bytes)

Please refer to Chapter 2.9 Error processing, Page 21 if the controller does not react to these functions or emits an error code.

2.6.1 Read n bit

This function is used to read n ($n \leq 256$) bits starting from a specific address.

Data request

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

Response

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

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

Example

Reading one bit starting at bit address 0x1073 (word address 0x0107 * 0x0010 + bit address 3).

Data request:

01	02	1073	0001	4CD1

Response:

01	02	01	01	6048
			Bit value	

2.6.2 Read n words

This function is used to read n (n ≤ 80) words starting from a specific address.

Data request

Slave address	Function x03 or 0x04	Address first word	Number of words	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	2 bytes

Example

Reading the displayed value of channels 1 and 2 (4 words in total) from word address = 0x0035 (displayed value, channel 1)

Data request:

01	03	00E7	0004	F43E

Response (values in the Modbus float format):

01	03	08	0000	41C8	0000	4120	4A9E
			Value of input 1 (25.0)		Value of input 2 (10.0)		

2.6.3 Write one bit

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

Instruction

Slave address	Function 0x05	Bit address	Bit value xx00	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes xx = 00 ➔ Bit is set to 0 xx = FF ➔ Bit is set to 1	2 bytes

Response

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

Example

Write one bit from bit address 0x2190
(Bit 0 of word address 0x0219).

Instruction:

01	05	2190	FF00	862B

Response (as instruction):

01	05	2190	FF00	862B

2.6.4 Write one word

For the Write Word function, the telegrams 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

Write the Modbus remote control value Bool 4 = 1

Word address = 0x021C

Instruction:

01	06	021C	0001	8874

Response (as instruction):

01	06	021C	0001	8874

2.6.5 Write n words

This function is used to write n (n ≤ 80) words starting from a specific address.

Instruction

Slave address	Function 0x10	Address first word	Number of words	Number of words	Word value(s)	ChecksumCRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	x byte	2 bytes

Response

Slave address	Function 0x10	Address first word	Number of words	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example

Write the Modbus remote control values Float 1 and Float 2

Word address = 0x0211 (Float 1)

Instruction:

01	10	0211	0004	08	0000	41C8	0000	4120	524D
					Float 1 (25,0)		Float 2 (10,0)		

Response:

01	10	0211	0004	9077

2.7 Transmission formats

16 bit Integer values

16 Bit integer values (and word values; 16 bit unsigned) are transmitted via Modbus in the following format:
The high byte first, followed by the low byte.

Example

Request of the word value of address 0x00A7, when this address contains the value "20" (word value 0x0014).

Request: 01 03 00A7 0001 (+ 2 bytes CRC16)
Response: 01 03 02 **0014** (+ 2 bytes CRC16)

32 bits unsigned Integer values

32 Bit unsigned integer values consist of two words and are transmitted via Modbus in the following format:
The high word first, followed by the low word.

Example

Request of the value of address 0x03B5, if value "207" (word value 0x000000CF) is written under this address.

Request: 01 03 03B5 0002 (+ 2 bytes CRC16)
Response: 01 03 04 **0000 00CF** (+ 2 bytes CRC16)

Float values

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

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

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

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

Modbus float format

Modbus address x		Modbus address x+1	
MMMMMMMM	MMMMMMMM	SEEEEEEE	EMMMMMMM
Byte 3	Byte 4	Byte 1	Byte 2

Example

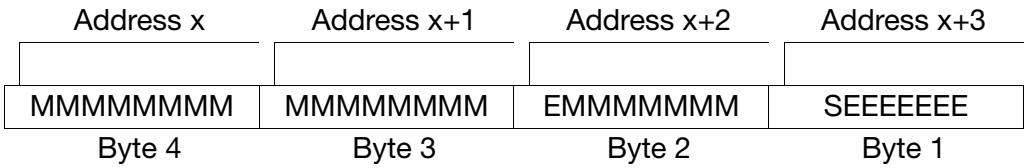
Request of the float value of address 0x00E7, when this address contains the value "55.0" (0x425C0000 in IEEE-754 format).

Request: 01 03 00E7 0002 (+ 2 bytes CRC16)
Response: 01 03 04 **0000 425C** (+ 2 bytes CRC16)

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

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

Float value



☞ Please find out the way float values are stored in your application. After the request, it might be necessary to swap the bytes over in the interface program you are using.

Character strings (texts)

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

☞ The address tables show the max. possible number of characters in a data type, e.g. "char10" (10 characters). If the data received contain a "\0" (ASCII code 0x00 as an end mark), any characters after this mark have no significance.

If no end mark is contained, the max. number of characters specified in the data type must be adhered to when writing. This prevents characters still contained in the memory from being appended to the text.

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

Example for the char10 data type

Reading software version 1 (here: "216.01.01I") at address 0x0000 (a max. of 10 characters can be read)

ASCII code for "216.01.01I":
0x32, 0x31, 0x36, 0x2E, 0x30, 0x31, 0x2E, 0x30, 0x31, 0x6C

Request: 01 03 0000 0005 (+ 2 bytes CRC16)
Slave address = 01
Function = 03, i.e. read n words
Address = 0000
Number of words to be read = 0005, because of the maximum of 10 characters

Response: 01 03 0A **32 31 36 2E 30 31 2E 30 31 6C** (+ 2 bytes CRC16)
Slave address = 01
Function = 03, i.e. read n words
Number of bytes read = 10 = 0x0A

2.8 Checksum (CRC16)

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

Calculation scheme

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

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

Example

Data request: Read two words, starting at address 0x00CE (CRC16 = 0x92A5)

07	03	00	CE	00	02	A5	92
						CRC16	

Response: (CRC16 = 0xF5AD)

07	03	04	00	00	41	C8	AD	F5
			Word 1		Word 2		CRC16	

2.9 Error processing

2.9.1 Modbus error codes

Error codes

The following error codes exist:

1 invalid function

2 invalid parameter address or too many words are to be read or written

8 write access to parameter denied

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 OR-ed with 0x80, i.e. the MSB (most significant bit) is set to 1.

Example

Data request:

01	03	40	00	00	04	CRC16

Response (with error code 2):

01	83	02	CRC16

Special cases

The slave not responding can have the following causes:

- the baud rate and/or data format of Master and Slave are not compatible
- the device address used does not coincide with that of the slave address
- 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

To avoid errors in the event of malfunctions, the master should, in all cases, repeat the data request if a timeout has elapsed without response (e.g. 1 s).

2.9.2 Error codes in float values

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

Error code for float values	Error
1.0×10^{37}	Underrange
2.0×10^{37}	Overrange
3.0×10^{37}	No valid input value
4.0×10^{37}	Division by zero
5.0×10^{37}	Math error
6.0×10^{37}	Invalid terminal temperature of thermocouple
7.0×10^{37}	Still no minimum value (fly back)
-7.0×10^{37}	Still no maximum value (fly back)
8.0×10^{37}	Integrator or statistics destroyed
9.0×10^{37}	Radio timeout

Example

Data request:

01	03	00E7	0002	743C

Response:

01	03	04	8E52	7DB4	51ED

The measured value 0x7DB48E52 (=3.0×10^{37}) supplied by channel 1 (Modbus address 0x00E7) indicates that the input value is invalid.

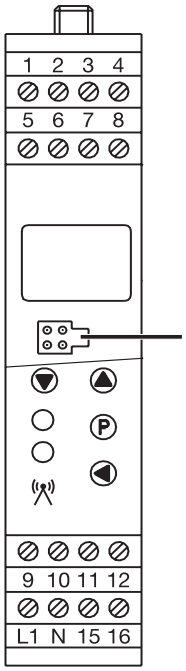
3.1 Connection diagram

The standard version of the device (receiver) described in this manual has two integrated interfaces:

- RS485 interface
- Setup interface

An interface connected to the front setup interface interrupts the communication via the RS485 interface, i.e. the setup interface has priority.

View on the terminal strips



Setup interface

RS485 interface screw connection

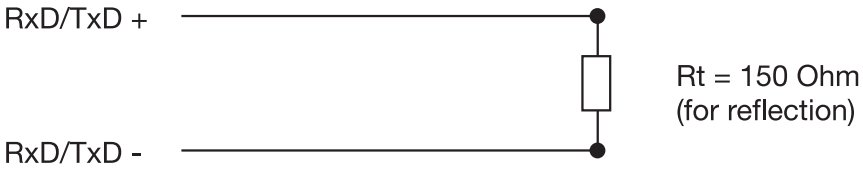
9	TxD+/RxD+	Transmit/receive data +
10	GND	Ground
11	TxD-/RxD-	Transmit/receive data -

3.2 Termination resistor

Reflections from cables always occur at the open cable ends (first and last stations of a bus system). The higher the baud rate selected, the stronger the reflection. A termination resistor is used to keep the reflections as small as possible.

A high-resistance termination resistor (in the kilo-Ohm range) integrated in the receiver is sufficient for short, interference-free cable lengths. For long, interference-prone cable lengths, one external resistor can be added to the bus as a termination. However, this must not be done with every slave because several slaves are wired in parallel.

Bus termination



3 RS485 interface

3.3 Configuration

The following table shows the possible settings of the RS485 interface carried out at the parameter level or with the setup program.

For more detailed information about configuration, please refer to operating manual B90.2931.0.

Parameter	**Bottom line display**	**Top line display**	**Range/Selection**
Baud rate	485.Bd	9600 19.2 38.4	**9600 bit(s)** 19200 bit(s) 38400 bit(s)
Data format (Data bit/parity/ Stop bit	485.Fo	8n1 8o1 8E1 8n2	**8/no/1** 8/odd/1 8/even/1 8/no/2
Minimum response time	485.tA	**30**	0 to 500ms
Device address	485.Ad	1	**1** to 254

The set-up interface is operated with a fixed setting and independent of the parameters of the RS485 interface (baud rate: 9600 bit(s), data format: 8n1, min. response time: 0ms, device address: 1).

When the communication takes place via the setup interface, the RS485 interface is inactive.

4.1 Data types and type of access

The Modbus address table contains descriptions of all process values (variables) including their addresses, data type and type of access.

Meaning:

char10	Text 10 characters
char4	Text 4 characters
float	Float value (4 bytes) as per IEEE 754
uint32	Unsigned Integer 32 Bit (2 words)
word	Unsigned Integer 16 Bit (1 word)
bool	Bool can be read and written as word, its value range is 0 to 1
byte	Byte (8 bit) can be read and written as word, its value range is 0 to 255
Bit x	Bit No. x
r/o	Read only access
r/w	Read/write access

4.2 Determination of the bit address

The Modbus address table contains the word addresses of the individual device and process values.

No specific bit addresses are contained, however, can be calculated with the aid of the following formula:

Bit address = Word address * 16 + bit number

Example

Word address according to the address table: 0x0107 (channel 1 - alarm output)

We want to find the bit address for the LowBat alarm (bit 3)

Bit address = 0x0107 * 0x0010 + 0x0003
= 0x1070 + 0x0003
= 0x1073

4 Modbus addresses

4.3 Modbus address table

Address (hex)	Type of variable	Access	Designation
0000	char 10	r/o	Software version 1, "216.xx.yy"
0033	char 4	r/o	Software version 2
0067	float	r/o	Raw value channel 1 (for Pt1000 transmitter in Ohm)
0069	float	r/o	Raw value channel 2 (for Pt1000 transmitter in Ohm)
006B	float	r/o	Raw value channel 3 (for Pt1000 transmitter in Ohm)
006D	float	r/o	Raw value channel 4 (for Pt1000 transmitter in Ohm)
006F	float	r/o	Raw value channel 5 (for Pt1000 transmitter in Ohm)
0071	float	r/o	Raw value channel 6 (for Pt1000 transmitter in Ohm)
0073	float	r/o	Raw value channel 7 (for Pt1000 transmitter in Ohm)
0075	float	r/o	Raw value channel 8 (for Pt1000 transmitter in Ohm)
0077	float	r/o	Raw value channel 9 (for Pt1000 transmitter in Ohm)
0079	float	r/o	Raw value channel 10 (for Pt1000 transmitter in Ohm)
007B	float	r/o	Raw value channel 11 (for Pt1000 transmitter in Ohm)
007D	float	r/o	Raw value channel 12 (for Pt1000 transmitter in Ohm)
007F	float	r/o	Raw value channel 13 (for Pt1000 transmitter in Ohm)
0081	float	r/o	Raw value channel 14 (for Pt1000 transmitter in Ohm)
0083	float	r/o	Raw value channel 15 (for Pt1000 transmitter in Ohm)
0085	float	r/o	Raw value channel 16 (for Pt1000 transmitter in Ohm)
0087	uint32	r/o	Channel 1 (update time in sec since power ON)
0089	uint32	r/o	Channel 2 (update time in sec since power ON)
008B	uint32	r/o	Channel 3 (update time in sec since power ON)
008D	uint32	r/o	Channel 4 (update time in sec since power ON)
008F	uint32	r/o	Channel 5 (update time in sec since power ON)
0091	uint32	r/o	Channel 6 (update time in sec since power ON)
0093	uint32	r/o	Channel 7 (update time in sec since power ON)
0095	uint32	r/o	Channel 8 (update time in sec since power ON)
0097	uint32	r/o	Channel 9 (update time in sec since power ON)
0099	uint32	r/o	Channel 10 (update time in sec since power ON)
009B	uint32	r/o	Channel 11 (update time in sec since power ON)
009D	uint32	r/o	Channel 12 (update time in sec since power ON)
009F	uint32	r/o	Channel 13 (update time in sec since power ON)
00A1	uint32	r/o	Channel 14 (update time in sec since power ON)
00A3	uint32	r/o	Channel 15 (update time in sec since power ON)
00A5	uint32	r/o	Channel 16 (update time in sec since power ON)
00A7	word	r/o	Channel 1 (transmit interval in 1/10s)
00A8	word	r/o	Channel 2 (transmit interval in 1/10s)
00A9	word	r/o	Channel 3 (transmit interval in 1/10s)

Address (hex)	Type of variable	Access	Designation
00AA	word	r/o	Channel 4 (transmit interval in 1/10s)
00AB	word	r/o	Channel 5 (transmit interval in 1/10s)
00AC	word	r/o	Channel 6 (transmit interval in 1/10s)
00AD	word	r/o	Channel 7 (transmit interval in 1/10s)
00AE	word	r/o	Channel 8 (transmit interval in 1/10s)
00AF	word	r/o	Channel 9 (transmit interval in 1/10s)
00B0	word	r/o	Channel 10 (transmit interval in 1/10s)
00B1	word	r/o	Channel 11 (transmit interval in 1/10s)
00B2	word	r/o	Channel 12 (transmit interval in 1/10s)
00B3	word	r/o	Channel 13 (transmit interval in 1/10s)
00B4	word	r/o	Channel 14 (transmit interval in 1/10s)
00B5	word	r/o	Channel 15 (transmit interval in 1/10s)
00B6	word	r/o	Channel 16 (transmit interval in 1/10s)
00B7	float	r/o	Channel 1 (transmitter battery voltage in Volt)
00B9	float	r/o	Channel 2 (transmitter battery voltage in Volt)
00BB	float	r/o	Channel 3 (transmitter battery voltage in Volt)
00BD	float	r/o	Channel 4 (transmitter battery voltage in Volt)
00BF	float	r/o	Channel 5 (transmitter battery voltage in Volt)
00C1	float	r/o	Channel 6 (transmitter battery voltage in Volt)
00C3	float	r/o	Channel 7 (transmitter battery voltage in Volt)
00C5	float	r/o	Channel 8 (transmitter battery voltage in Volt)
00C7	float	r/o	Channel 9 (transmitter battery voltage in Volt)
00C9	float	r/o	Channel 10 (transmitter battery voltage in Volt)
00CB	float	r/o	Channel 11 (transmitter battery voltage in Volt)
00CD	float	r/o	Channel 12 (transmitter battery voltage in Volt)
00CF	float	r/o	Channel 13 (transmitter battery voltage in Volt)
00D1	float	r/o	Channel 14 (transmitter battery voltage in Volt)
00D3	float	r/o	Channel 15 (transmitter battery voltage in Volt)
00D5	float	r/o	Channel 16 (transmitter battery voltage in Volt)
00D7	bool	r/w	Channel 1 (reset fly back)
00D8	bool	r/w	Channel 2 (reset fly back)
00D9	bool	r/w	Channel 3 (reset fly back)
00DA	bool	r/w	Channel 4 (reset fly back)
00DB	bool	r/w	Channel 5 (reset fly back)
00DC	bool	r/w	Channel 6 (reset fly back)
00DD	bool	r/w	Channel 7 (reset fly back)
00DE	bool	r/w	Channel 8 (reset fly back)
00DF	bool	r/w	Channel 9 (reset fly back)
00E0	bool	r/w	Channel 10 (reset fly back)

4 Modbus addresses

Address (hex)	Type of variable	Access	Designation
00E1	bool	r/w	Channel 11 (reset fly back)
00E2	bool	r/w	Channel 12 (reset fly back)
00E3	bool	r/w	Channel 13 (reset fly back)
00E4	bool	r/w	Channel 14 (reset fly back)
00E5	bool	r/w	Channel 15 (reset fly back)
00E6	bool	r/w	Channel 16 (reset fly back)
00E7	float	r/o	Displayed value channel 1 (for Pt1000 transmitter in °C/°F)
00E9	float	r/o	Displayed value channel 2 (for Pt1000 transmitter in °C/°F)
00EB	float	r/o	Displayed value channel 3 (for Pt1000 transmitter in °C/°F)
00ED	float	r/o	Displayed value channel 4 (for Pt1000 transmitter in °C/°F)
00EF	float	r/o	Displayed value channel 5 (for Pt1000 transmitter in °C/°F)
00F1	float	r/o	Displayed value channel 6 (for Pt1000 transmitter in °C/°F)
00F3	float	r/o	Displayed value channel 7 (for Pt1000 transmitter in °C/°F)
00F5	float	r/o	Displayed value channel 8 (for Pt1000 transmitter in °C/°F)
00F7	float	r/o	Displayed value channel 9 (for Pt1000 transmitter in °C/°F)
00F9	float	r/o	Displayed value channel 10 (for Pt1000 transmitter in °C/°F)
00FB	float	r/o	Displayed value channel 11 (for Pt1000 transmitter in °C/°F)
00FD	float	r/o	Displayed value channel 12 (for Pt1000 transmitter in °C/°F)
00FF	float	r/o	Displayed value channel 13 (for Pt1000 transmitter in °C/°F)
0101	float	r/o	Displayed value channel 14 (for Pt1000 transmitter in °C/°F)
0103	float	r/o	Displayed value channel 15 (for Pt1000 transmitter in °C/°F)
0105	float	r/o	Displayed value channel 16 (for Pt1000 transmitter in °C/°F)
0107	byte	r/o	Channel 1 (alarm output) Bit 0 = 0x01 : Radio timeout Bit 1 = 0x02 : Alarm monitoring 1 Bit 2 = 0x04 : Alarm monitoring 2 Bit 3 = 0x08 : LowBat from transmitter

Address (hex)	Type of variable	Access	Designation
0108	byte	r/o	Channel 2 (alarm output)
0109	byte	r/o	Channel 3 (alarm output)
010A	byte	r/o	Channel 4 (alarm output)
010B	byte	r/o	Channel 5 (alarm output)
010C	byte	r/o	Channel 6 (alarm output)
010D	byte	r/o	Channel 7 (alarm output)
010E	byte	r/o	Channel 8 (alarm output)
010F	byte	r/o	Channel 9 (alarm output)
0110	byte	r/o	Channel 10 (alarm output)
0111	byte	r/o	Channel 11 (alarm output)
0112	byte	r/o	Channel 12 (alarm output)
0113	byte	r/o	Channel 13 (alarm output)
0114	byte	r/o	Channel 14 (alarm output)
0115	byte	r/o	Channel 15 (alarm output)
0116	byte	r/o	Channel 16 (alarm output)
0117	float	r/o	Channel 1 (fly back, min.)
0119	float	r/o	Channel 2 (fly back, min.)
011B	float	r/o	Channel 3 (fly back, min.)
011D	float	r/o	Channel 4 (fly back, min.)
011F	float	r/o	Channel 5 (fly back, min.)
0121	float	r/o	Channel 6 (fly back, min.)
0123	float	r/o	Channel 7 (fly back, min.)
0125	float	r/o	Channel 8 (fly back, min.)
0127	float	r/o	Channel 9 (fly back, min.)
0129	float	r/o	Channel 10 (fly back, min.)
012B	float	r/o	Channel 11 (fly back, min.)
012D	float	r/o	Channel 12 (fly back, min.)
012F	float	r/o	Channel 13 (fly back, min.)
0131	float	r/o	Channel 14 (fly back, min.)
0133	float	r/o	Channel 15 (fly back, min.)
0135	float	r/o	Channel 16 (fly back, min.)
0137	float	r/o	Channel 1 (fly back, max.)
0139	float	r/o	Channel 2 (fly back, max.)
013B	float	r/o	Channel 3 (fly back, max.)
013D	float	r/o	Channel 4 (fly back, max.)
013F	float	r/o	Channel 5 (fly back, max.)
0141	float	r/o	Channel 6 (fly back, max.)
0143	float	r/o	Channel 7 (fly back, max.)
0145	float	r/o	Channel 8 (fly back, max.)

4 Modbus addresses

Address (hex)	Type of variable	Access	Designation
0147	float	r/o	Channel 9 (fly back, max.)
0149	float	r/o	Channel 10 (fly back, max.)
014B	float	r/o	Channel 11 (fly back, max.)
014D	float	r/o	Channel 12 (fly back, max.)
014F	float	r/o	Channel 13 (fly back, max.)
0151	float	r/o	Channel 14 (fly back, max.)
0153	float	r/o	Channel 15 (fly back, max.)
0155	float	r/o	Channel 16 (fly back, max.)
020F	bool	r/o	Relay output 1 (current state)
0210	bool	r/o	Relay output 2 (current state)
0211	float	r/w	Modbus remote control value float 1
0213	float	r/w	Modbus remote control value float 2
0215	float	r/w	Modbus remote control value float 3
0217	float	r/w	Modbus remote control value float 4
0219	bool	r/w	Modbus remote control value bool 1
021A	bool	r/w	Modbus remote control value bool 2
021B	bool	r/w	Modbus remote control value bool 3
021C	bool	r/w	Modbus remote control value bool 4
0222	byte	r/o	LED states Bit 0 = 0x01 : green (bicolor LED) Bit 1 = 0x02 : red (bicolor LED) Bit 2 = 0x04 : yellow
0223	byte	r/o	Error flags Bit 0 = 0x01 = Configuration was automatically initialized with factory data Bit 1 = 0x02 = Calibration constants were automatically initialized with factory data
0224	bool	r/o	Multi-input fault warning
0225	bool	r/o	Multi-input RadioTimeOuts channels 1 to 16
0226	bool	r/o	Multi-input MinMax1Alarm channels 1 to 16
0227	bool	r/o	Multi-input MinMax2Alarm channels 1 to 16
0228	bool	r/o	Multi-input LowBatAlarm channels 1 to 16
0229	bool	r/o	Multi-input MinMaxAlarm channels 1 to 16
03B5	uint32	r/w	Channel 1 (linked transmitter ID)
03B7	uint32	r/w	Channel 2 (linked transmitter ID)
03B9	uint32	r/w	Channel 3 (linked transmitter ID)

Address (hex)	Type of variable	Access	Designation
03BB	uint32	r/w	Channel 4 (linked transmitter ID)
03BD	uint32	r/w	Channel 5 (linked transmitter ID)
03BF	uint32	r/w	Channel 6 (linked transmitter ID)
03C1	uint32	r/w	Channel 7 (linked transmitter ID)
03C3	uint32	r/w	Channel 8 (linked transmitter ID)
03C5	uint32	r/w	Channel 9 (linked transmitter ID)
03C7	uint32	r/w	Channel 10 (linked transmitter ID)
03C9	uint32	r/w	Channel 11 (linked transmitter ID)
03CB	uint32	r/w	Channel 12 (linked transmitter ID)
03CD	uint32	r/w	Channel 13 (linked transmitter ID)
03CF	uint32	r/w	Channel 14 (linked transmitter ID)
03D1	uint32	r/w	Channel 15 (linked transmitter ID)
03D3	uint32	r/w	Channel 16 (linked transmitter ID)
03D5	uint32	r/o	Seconds since power ON
03D7	byte	r/o	HW identification, 0 to 15

4 Modbus addresses

JUMO GmbH & Co. KG
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 CM 20 2 TT, UK
Phone: +44 1279 635533
Fax: +44 1279 635262
e-mail: sales@jumo.co.uk
Internet: www.jumo.co.uk

JUMO Process Control, Inc.
8 Technology Boulevard
Canastota, NY 13032, USA
Phone: 315-697-JUMO
1-800-554-JUMO
Fax: 315-697-5867
e-mail: info@jumo.us
Internet: www.jumo.us