JUMO digiLine CR/Ci

Intelligent electronic components with digital interface for conductivity sensors from JUMO



Modbus interface description

20276000T92Z001K000

V1.00/EN/00726129/2019-10-09

Contents

1 Safety information 5

1.1 Warning symbols 5
1.2 Note signs 5

2 Modbus protocol description 7

2.1 Master-slave principle 7
2.2 Transmission media for Modbus 8
2.3 Structure of an RTU Modbus telegram 8
2.4 Function codes 8
2.4.1 Reading n words 9
2.4.2 Writing one word 10
2.4.3 Writing n words 11
2.5 Data types 11
2.6 Examples of data transmission 12
2.6.1 Integer values 13
2.6.2 Float values 13
2.6.3 Character strings (texts) 15
2.7 Checksum (CRC16) 16
2.8 Error messages 17
2.8.1 Modbus error codes 17
2.8.2 Error messages for invalid values 18

3 Interfaces 19

3.1 Interface assignment 19
3.1.1 Terminating resistors 19

4 Configuring interfaces 21

5 Modbus over a serial interface 23

5.1 Modbus slave operation via RS485 serial interface 23

6 Modbus address tables 25

6.1 Nameplate 25
6.2 Measuring point info 26
6.3 Configuration of the interface 26
6.4 Calibration data 27
6.4.1 Calibration logbook entry 1 27
6.4.2 Calibration logbook entry 2 28
6.4.3 Calibration logbook entry 3 29
6.4.4 Calibration logbook entry 4 30
6.4.5 Calibration logbook entry 5 31
6.4.6 Calibration logbook entry 6 33
6.4.7 Calibration logbook entry 7 34

Contents

6.4.8 Calibration logbook entry 8 35
6.4.9 Calibration logbook entry 9 36
6.4.10 Calibration logbook entry 10 37
6.4.11 Relative cell constants 38
6.5 Configuration of conductivity input CR 38
6.6 Configuration of conductivity input Ci 40
6.7 Configuration of temperature input 42
6.8 Configuration of calibration timers 42
6.9 Configuration of sensor stress 42
6.10 Configuration of sensor monitoring 42
6.11 Configuration of display 43
6.12 Process values 45
6.12.1 Inputs 45
6.12.2 Calibration alarm 46
6.12.3 Sensor stress 46
6.12.4 Sensor monitoring 46
6.12.5 Date and time for time stamp 47
6.12.6 Summary of important process values 47

1.1 Warning symbols



DANGER!

This symbol indicates that **personal injury from electrocution** may occur if the appropriate precautionary measures are not taken.



WARNING!

This symbol in connection with the signal word indicates that **personal injury** may occur if the respective precautionary measures are not carried out.



CAUTION!

This symbol in connection with the signal word indicates that **material damage or data loss** will occur if the respective precautionary measures are not taken.



CAUTION!

This symbol indicates that **components could be destroyed** by electrostatic discharge (ESD = Electro Static Discharge) if the respective cautionary measures are not taken.

Only use the ESD packages intended for this purpose to return device inserts, assembly groups, or assembly components.



READ THE DOCUMENTATION!

This symbol, which is attached to the device, indicates that the associated **documentation for the device** must be **observed**. This is necessary to identify the nature of the potential hazard, and to take measures to prevent it.

1.2 Note signs



NOTE!

This symbol refers to **important information** about the product, its handling, or additional benefits.



REFERENCE!

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



FURTHER INFORMATION!

This symbol is used in tables and indicates that **further information** is provided after the table.



DISPOSAL!

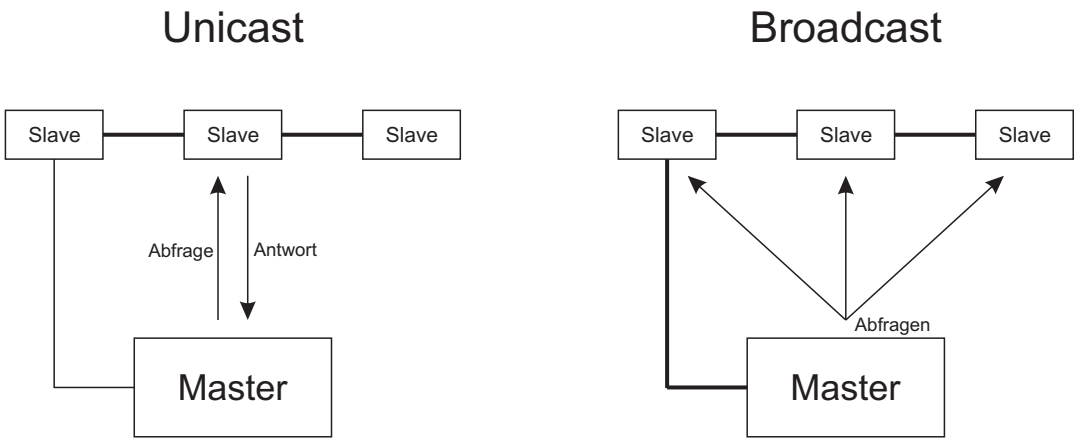
At the end of its service life, the device and any batteries present do not belong in the trash! Please ensure that they are **disposed of** properly and in an **environmentally friendly** manner.

1 Safety information

2.1 Master-slave principle

Communication between a master (e. g. SCADA system or PLC) and a JUMO digiLine CR/Ci as slave in in a Modbus takes place according to the master-slave principle in the form of data request/instruction - response. Slaves are identified by the device address. Master devices do not need an address.

⇨ chapter 5 "Modbus over a serial interface", Page 23



The master controls data exchange by cyclically querying the slaves on the overall bus. The slaves (e. g. JUMO digiLine CR/Ci) merely respond. The master has write and read access to the slaves. This enables data to be communicated between master and slave devices in real-time. Slaves cannot communicate with one another directly. In order to transmit data from slave to slave, the master must extract data from one slave and transfer it to the next.

As a rule, the master directs its queries to individual slaves. For this, it must communicate with the individual slaves by means of their Unicast address. Queries can be directed to all slaves on the bus also in the form of a broadcast message. In this case, the broadcast address is used as the slave address. Slaves do not respond to broadcast queries. Data collisions would otherwise result in serial bus systems. For this reason, use of broadcast addresses only makes sense with function codes for writing data. Broadcasts cannot be used with function codes for reading data.

NOTE!

The JUMO digiLine CR/Ci can only be operated as a slave.

2 Modbus protocol description

2.2 Transmission media for Modbus

Serial interface

For data communication via a **serial interface**, the Modbus specification provides the transmission modes **RTU mode** (**R**emote **T**erminal **U**nit) and ASCII mode (transmission of data in ASCII format). The JUMO digiLine CR/Ci only supports the **RTU-Mode**. Here, the data are transmitted via the serial bus in binary format (RS422/485).

⇨ chapter 5 "Modbus over a serial interface", Page 23

2.3 Structure of an RTU Modbus telegram

Modbus telegrams have the following structure:

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

Every telegram has four fields:

Slave address	Device address of a specific slave
Function code	Function selection (read/write words)
Data field	Contains the information (depending on the function code) - word address/bit address - number of words/number of bits - word value(s)/bit value(s)
Checksum	Detection of transmission errors

2.4 Function codes

Function overview

The functions described in the following (from the Modbus standard) are available for extracting measured values, device and process data, and for writing data.

Function code		**Function**	**Limit**
Hex	**Dec.**		
03 or 04	3 or 4	Reading n words	Max. 125 words (250 bytes)
06	6	Writing one word	Max. 1 word (2 bytes)
10	16	Writing n words	Max. 123 words (246 bytes)

2.4.1 Reading n words

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

Data request

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

Example

Reading the device name. In the example here, the device name is "dl CR". Since the device name is up to 10 characters long (including ZERO characters as end identifier) and 2 ASCII characters of the device name are saved per word, it is necessary here to read 5 words, i.e. 10 bytes. For these and other Modbus addresses, see chapter 6 "Modbus address tables", Page 25.

Hex code for the data request:

01	03	00 00	00 05	85 C9
Slave	Function	Address of 1st word	Number of words	CRC

Hex code for the response (values in byte format):

01	03	0 A	64 6C 20 43 52 00 00 00 00 00	E6 B1
Slave	Function	Bytes read	Text in ASCII	CRC

2.4.2 Writing one word

The data blocks for the instruction and response are identical when writing a word.



CAUTION!

Write operations in some R/W parameters result in them being saved in the EEPROM or flash memory. These memory modules have only a limited number of write cycles (approx. 100,000 or 10,000).

Thus, frequent writing of certain variables can result in a memory error.

▶ The number of write operations should therefor be kept as low as possible.

Instruction

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

Response

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

Example

In this example, a command for the value "Time for CIP recognition" of the device name is to be written. In this case, the slave address of the device is 1, the word address is 0x5A42 (chapter 6.10 "Configuration of sensor monitoring", Page 42) and the word to write should be "2700" (2700 s corresponds to 45 min).

Hex code for the instruction:

01	06	5A 42	0A 8C	3C 03
Slave	Function	Word address	Value	CRC

Hex code for the response:

01	06	5A 42	0A 8C	3C 03
Slave	Function	Word address	Value	CRC

2.4.3 Writing n words

Instruction

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

Response

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

Example

Writing the word "North plant" (UTF-8 encoding with end identifier: 0x41 0x6E 0x6C 0x61 0x67 0x65 0x20 0x4E 0x6F 0x72 0x64 0x00) after word address 0x021E as description of the measuring point.

⇨ chapter 6.2 "Measuring point info", Page 26

Hex code for the instruction:

01	10	02 1E	00 06	0C	41 6E 6C 61 67 65 20 4E 6F 72 64 00	F8 70
Slave	Function	Address 1st word	Number of words	Number of bytes	Text in UTF-8 encoding	CRC

Hex code for the response:

01	10	02 1E	00 06	21 B5
Slave	Function	Address 1st word	Number of words	CRC

2.5 Data types

Data type	Description	Access	Possible Function codes	Number Modbus register
char	Low byte of a word as an integer value; the high byte is not used. Value ranges for numerical interpretation: 0 to 255 for unsigned data -128 to 127 for signed data For Boolean interpretation: Least significant bit in a word as bit value 0000 0000 0000 000**1** = 1 or TRUE 0000 0000 0000 000**0** = 0 or FALSE	read only	03, 04	1
		read/ write	03, 04, 06, 16	
short	Word (16 bit) as integer value Value ranges: 0 to 65535 for unsigned data -32768 to 32767 for signed data	read only	03, 04	1
		read/ write	03, 04, 06, 16	

<table>
<tr><th>Data type</th><th>Description</th><th>Access</th><th>Possible Function codes</th><th>Number Modbus register</th></tr>
<tr><td rowspan="2">float</td><td rowspan="2">2 words as 32-bit floating-point number with encoding to IEEE 754, where the correct order of transmitting the 4 bytes must be observed during transmission. In the configuration of the RS485 interface, the desired encoding can be selected under the "Floating-point format" setting.
S = Sign bit
E = Exponent (two's complement)
M = 23-bit normalized mantissa
IEEE 754 Big Endian
<table><tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td></tr><tr><td>SE7EEEEEE1</td><td>E0M23MMMMMM16</td><td>M15MMMMMMM8</td><td>M7MMMMMMM0</td></tr></table>IEEE 754 Little Endian
<table><tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td></tr><tr><td>M7MMMMMMM0</td><td>M15MMMMMMM8</td><td>E0M23MMMMMM16</td><td>SE7EEEEEE1</td></tr></table>Standard modbus coding
<table><tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td></tr><tr><td colspan="2">1. Modbus register</td><td colspan="2">2. Modbus register</td></tr><tr><td>M15MMMMMMM8</td><td>M7MMMMMMM0</td><td>SE7EEEEEE1</td><td>E0M23MMMMMM16</td></tr></table></td><td>read only</td><td>03, 04</td><td rowspan="2">2</td></tr>
<tr><td>read/ write</td><td>03, 04, 16</td></tr>
<tr><td rowspan="2">long</td><td rowspan="2">Double word (32-bit) as unsigned integer value
Value range: 0 to 4.294.967.295</td><td>read only</td><td>03, 04</td><td rowspan="2">2</td></tr>
<tr><td>read/ write</td><td>03, 04, 16</td></tr>
<tr><td rowspan="2">string[n]</td><td rowspan="2">Character string for a length of up to n bytes. The character encoding of the individual variables can be found in the Modbus address tables in the last chapter of this document. UTF-8, ISO 8859-1 or ASCII encoding appear as character set codings.
Each of the n Modbus register words contains 2 successive bytes of the character string.
Note that the character string must always have "\0" (ASCII code 0x00) as the terminating code. In addition, the character string length must have an even number of bytes overall. To ensure this in the case of an odd number of bytes, including "/0", an additional "/0" must be appended.</td><td>read only</td><td>03, 04</td><td rowspan="2">n/2</td></tr>
<tr><td>read/ write</td><td>03, 04, 06, 16</td></tr>
</table>

2.6 Examples of data transmission

The function 0x03 or 0x04 (writing of n words) is used to extract integers, floating-point values and text values.

Data request

Slaveaddress	Function 0x03 or 0x04	Address of first word	Number of words	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Integer values are transferred via Modbus in the following format:
First the high, then the low byte.

Response

Slaveaddress	Function 0x03 or 0x04	Number of bytes read	Word value(s)	Checksum CRC
1 byte	1 byte	1 byte	x bytes	2 bytes

2.6.1 Integer values

Example

In this example, the time for CIP recognition should be read at address 0x5A42 (chapter 6.10 "Configuration of sensor monitoring", Page 42). The value here should be "2400" (word value 0x0960).

Data request:

01	03	5A 42	00 01	36 C6
Slave	Function	Address of 1st word	Number of words	CRC

Response (values in Modbus floating-point format):

01	03	02	09 60	BE 3C
Slave	Function	Bytes read	Integer value	CRC

2.6.2 Float values

For floating-point values, this device functions with the IEEE 754 standard format (32-bit), but with the difference that bytes 1 and 2 are interchanged with bytes 3 and 4.

Single floating-point format (32-bit) according to standard IEEE 754

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

S - Prefix sign bit

E - Exponent (two's complement)

M - 23-bit normalized mantissa

Modbus floating-point format

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

Example

In this example, the value "Temperature value CIP recognition" should be read at address 0x5A40 of the device. The value should be 70.25 (0x428C8000 in IEEE-754 format).

Data request:

01	03	5A 40	00 02	D7 07
Slave	Function	Address of 1st word	Number of words	CRC

Response (values in Modbus floating-point format):

01	03	04	80 00	42 8C	E2 F6
Slave	Function	Bytes read	Floating-point value		CRC

After being transmitted from the device, the bytes for the floating-point value must be interchanged accordingly. Many compilers (for example, Microsoft® Visual C++) store the floating-point values in the following sequence:

Floating-point value

Address x	**Address x+1**	**Address x+2**	**Address x+3**
MMMMMMMM	MMMMMMMM	EMMMMMMM	SEEEEEEE
Byte 4	Byte 3	Byte 2	Byte 1

NOTE!

The sequence of the bytes depends on how floating-point values are saved in the application concerned. It may be necessary for the bytes to be exchanged in the interface program accordingly.

2.6.3 Character strings (texts)

Character strings are coded in UTF8 for transmission.

NOTE!

A "\0" (ASCII code 0x00) must always be transmitted as the terminating code. Subsequent characters have no meaning.
Since texts are transmitted word by word (16-bit register), 0x00 is also added to the end for an uneven number of bytes (incl. "\0").
The maximum lengths for character strings specified in the address tables contain the final "\0". This means that, in the case of "String[60]", the text, including "\0", must not be longer than 60 bytes.
In the case of 19 Unicode characters having a length of 3 bytes each, only 2 bytes remain for the 20th character. 1 byte is needed for the terminating code.

Example

Querying the text from address 0x021E, this address holds the character string for the description of the measuring point "North plant" (ASCII code: 0x41 0x6E 0x6C 0x61 0x67 0x65 0x20 0x4E 0x6F 0x72 0x64).

Hex code for the query:

01	03	02 1E	00 07	65 B6
Slave	Function	Address 1st word	Number of words	CRC

Hex code for the response:

01	03	00 0E	41 6E 6C 61 67 65 20 4E 6F 72 64 00 00 AA	C5 DF
Slave	Function	Bytes read	Word values (ASCII characters)	CRC

NOTE!

The values (here: 00 AA) before the CRC sum (here: C5DF) are not taken into consideration, since they are after the end identifier "\0".

2.7 Checksum (CRC16)

Calculation principle

Transmission errors are detected with the aid of the checksum (CRC16). If an error is detected during evaluation, the device concerned does not respond.

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

NOTE!

The low byte of the checksum is transmitted first!
Example: The CRC16 checksum CC DD is transmitted and represented in the sequence DD CC.

Example

Querying the binary value of the sensor stress pre-alarm at the address 0x2601:

Instruction: read a word from address 0x2601

01	03	26 01	00 01	DE 82
Slave	Function	address	Read one word	CRC

Response (CRC16 = 0x8479)

01	03	02	00 01	79 84
Slave	Function	Number of bytes	Word 1	CRC

Word 1 = 1 means that the binary value of the pre-alarm for the sensor stress = 1.

2.8 Error messages

2.8.1 Modbus error codes

Requirements for Modbus communication

The following conditions must be met for a slave to receive, process, and respond to queries:

- Baud rate and data format of master and slave must match.
- The correct slave address must be used in the query.
- Slave devices respond only after a successful checksum check of the query by the slave. Otherwise, the query is rejected by the slave.
- The instruction from the master must be complete and conform to the Modbus protocol.
- The number of words to be read must be greater than 0.

Error codes

If the data request from the master has been received by the slave without transmission errors but cannot be processed, the slave responds with an error code. The following error codes may occur:

- 01 = Invalid function; the function codes supported by this device are listed in chapter 2.4 "Function codes", Page 8.
- 02 = Invalid address or too many words or bits are to be read or written
- 03 = Value is outside the admissible range
- 08 = Value is write-protected

Response to malfunction

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

The function code is OR-gated with 0x80. As a result, the highest-value bit (msb) is set to 1.

Example

Data query:

01	06	23 45	00 01	52 5B
Slave	Write word	Word address	Word value	CRC

Response (with error code 2):

01	86	02	C3 A1
Slave	OR function	Errors	CRC

Response with error code 02, because the address 0x2345 does not exist.

2.8.2 Error messages for invalid values

For measured values in float format, the error number itself is displayed as a value, i.e. instead of the measured value, the error number is returned.

Error code for float values	Error
1.0×10^{37}	Underrange
2.0×10^{37}	Overrange
3.0×10^{37}	Not a valid input value
4.0×10^{37}	Division by zero
5.0×10^{37}	Mathematical error
6.0×10^{37}	Invalid compensation temperature
7.0×10^{37}	Invalid floating-point value
8.0×10^{37}	Integrator or statistics destroyed

Example

Reading the compensated conductivity measured value at Modbus address 0x8002:

Data request:

08	03	80 02	00 02	4C 92
Slave	Function	Word address	Number of words	CRC

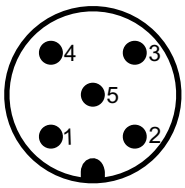
Response:

08	03	04	BD C2	7D 70	C6 17
Slave	Function	Bytes read	Error code		CRC

The measured value provided by the analog input 0x7D70BDC2 (=2.0×10^{37}) indicates that an overrange situation exists.

3.1 Interface assignment

five-pole M12 plug connector for RS485 (JUMO digiLine or Modbus RTU)

Pin	Potential	Symbol
1	+5 V	Plug connection 
2	+ 24 V	
3	GND	
4	RS 485 B (RxD/TxD-)	
5	RS 485 A (RxD/TxD+)	
The connection to the serial interface of a master with screw or spring-cage terminals is made with the aid of the JUMO digiLine device connection cable (see Accessories)		

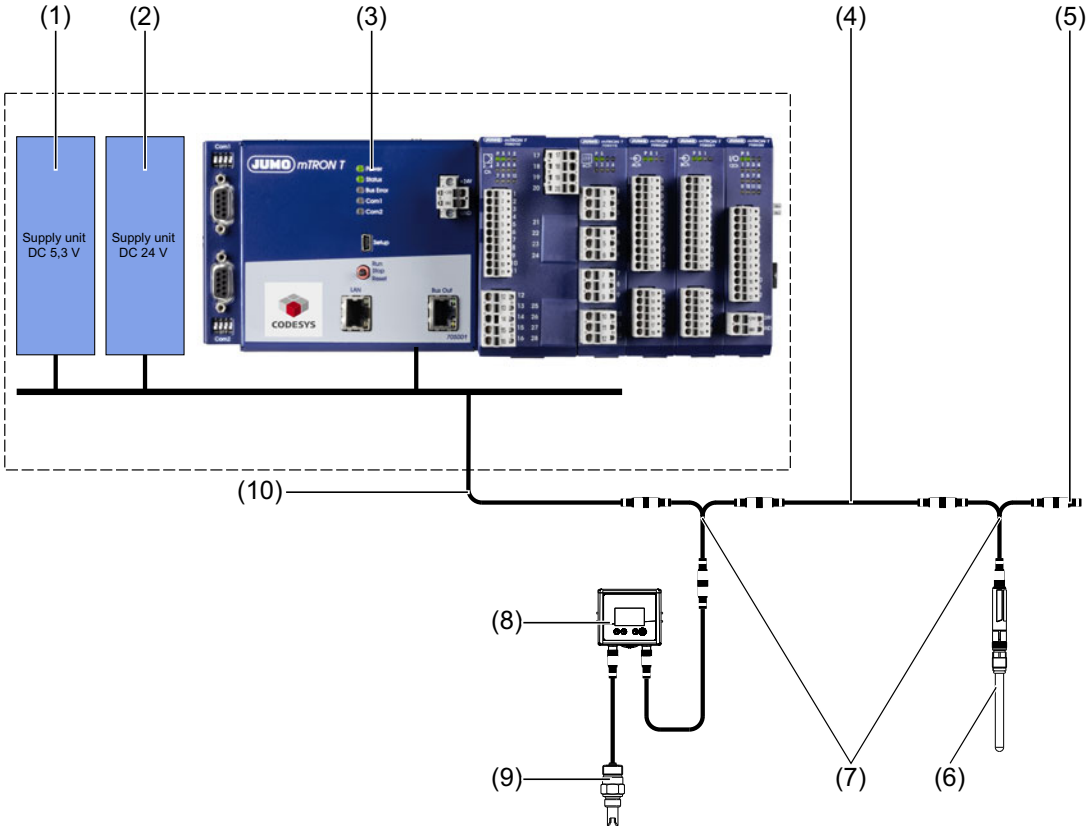
NOTE!

Preassembled connecting cables are available for connecting sensors to JUMO digiLine electronics. Use these to assure a reliable cable connection and data transmission in the bus system. In the order details provided in the operating manual, you can find a list of cables and connectors available as accessories.

3.1.1 Terminating resistors

A JUMO digiLine bus cable needs to be terminated with terminating connectors at the ends distant from the master unit (part no.: 00461591). These are simply connected to the last free M12 coupling at the final Y-splitter (part no.: 00638327) in the bus. At the master unit end of the bus cable, termination of the RS485 bus cable must be made in accordance with the operating manual for the Modbus master unit.

Example of connecting 2 JUMO sensors to JUMO digiLine electronics on a JUMO mTRON T



(1) Stabilized power supply unit with DC 5.3 V output for feeding sensors with JUMO digiLine pH/ORP/T five-pole (current limiting via 3 A fuse required)

(2) Stabilized power supply unit with DC 24 V output for feeding sensors with JUMO digiLine CR and JUMO mTRON T (current limiting via 3 A fuse required)

(3) JUMO mTRON T central processing unit with activated PLC function and RS422/485 Modbus RTU (see order data for JUMO mTRON T)

(4) JUMO M12 connecting cable, 5-pole and A-coded

(5) JUMO M12 terminating connector, 5-pole for bus termination

(6) JUMO sensors with JUMO digiLine pH/ORP/T five-pole

(7) JUMO Y-splitter, 5-pole with 2× M12 cable sockets and 1× M12 connector, each of which is A-coded

(8) JUMO digiLine CR in device version with RS485 interface

(9) Conductivity sensor

(10) JUMO digiLine master connecting cable with open wire ends at one end for connection to devices with screw or spring-cage terminals (see Accessories); for connection to the Modbus system, refer to the wiring diagram in the following.

Notes



CAUTION!

Improper installation or the wrong settings on equipment can result in unexpected operating states of a plant.

This can disrupt processes from functioning properly or result in damage.

▶ For this reason, it is always necessary to provide safety devices that are independent of the device and to allow settings to be made only by qualified personnel.



NOTE!

Changes to the configuration settings described in this chapter for JUMO digiLine electronics can be made on a PC using the JUMO DSM software.

Operation, configuration, and parameterization procedures are explained in detail in the operating manual for the JUMO digiLine CR/Ci.

⇨ Operating manual for JUMO digiLine CR/Ci

4 Configuring interfaces

Settings for serial interfaces

For all user devices on a bus to be able to communicate with one another, their interface settings must match. The table below shows the setting options for the serial interfaces on the JUMO digiLine electronics.



NOTE!

The JUMO digiLine protocol assigns the interface parameters automatically during startup (Plug & Play). For operation on a Modbus master, the interface parameters must be set prior to initial commissioning with the JUMO DSM software.

Configuration item	Selection/settings	Description
Baud rate	9600 19200 38400	Transmission speed (symbol rate) of the serial interface
Data format	8 - 1 - no parity 8 - 1 - odd parity 8 - 1 - even parity 8 - 2 - no parity	Format of the data word Useful bit - stop bit - parity
Floating Point	Standard IEEE754_LITTLE IEEE754_BIG	Selectable transmission format for float values (floating-point numbers)
Minimum response time	0 to 500 ms	Minimum time from receipt of a query to transmission of a response This parameter is used to adjust the response speed of the device to slower bus users.
Device address	1 to 247	A bus user's unique identifier 0 = Broadcast address[a] 1 to 247 = Unicast addresses[b]

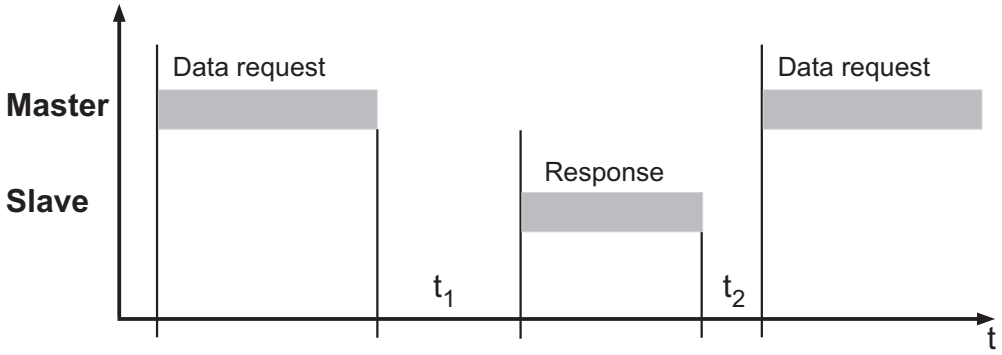
[a] Device addressing is specified in the Modbus standard. The broadcast address must not be used as a slave address. It is intended for broadcast messages.

[b] Unicast addresses are intended for use as slave addresses. They are used to identify the slave devices uniquely so that the master can communicate with them explicitly.

5.1 Modbus slave operation via RS485 serial interface

Chronological sequence of communication

A scanning cycle on a bus proceeds with the following timing:



t_1	Device's internal waiting period prior to checking of the data query and the internal processing time: min.: 3.5 byte times + minimum response time typically: 5 ms max.: 25 ms + minimum response time
t_2	The master has to observe this waiting period before starting a new data request: 3.5 characters or at least 2 ms

NOTE!

The minimum response time can be set in the configuration.
This set time must elapsed before a response is sent (0 to 500 ms). If processing of a query from a master is completed in the slave before the minimum response time has elapsed, the response is not sent until the "minimum response time" has elapsed.

NOTE!

During t_1 and t_2 and during the response time of the slave, no data requests may be generated by the master. Requests during t_1 and t_2 are ignored by the slave. Requests during the response time invalidate all the data currently on the bus.

NOTE!

The terminating code after a data query or data response is 3.5 characters long. The time needed for these 3.5 characters depends on the baud rate.

Character transmission time

The beginning and end of a data block are identified by pauses in transmission. The character transmission time (time to transmit one character) depends on the baud rate.
The following results for a data format of 8 data bits, no parity bit, and one stop bit:

Character transmission time [ms] = 1000 × 10 bits ÷ baud rate

For other data formats, the following is the result:

Character transmission time [ms] = 1000 × 11 bits ÷ baud rate

Example

Identifier for end of data request or end of response

Waiting period = 3.5 characters * 1000 * 11 bits ÷ baud rate

Baud rate [baud]	**Data format** [bit]	**Character transmission time**[ms]
38400	11	0.287
	10	0.260
19200	11	0.573
	10	0.521
9600	11	1.146
	10	1.042

6.1 Nameplate

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
0	0000	string[10]	5	r	Device name	ASCII encoding with "ZERO" as end identifier
5	0005	string[12]	6	r	Software version	ASCII encoding with "ZERO" as end identifier
11	000B	short	1	r	Software number	
12	000C	char	1	r	Sensor major version	
13	000D	char	1	r	Sensor minor version	
14	000E	short	1	r	Trial version	
15	000F	char	1	r	Hardware major version	
16	0010	char	1	r	Hardware minor version	
17	0011	string[14]	7	r	VDN Version	ASCII encoding with "ZERO" as end identifier
24	0018	short	1	r	VdN number	
25	0019	short	1	r	VDN version number	
64	0040	char	1	r	Major version of JUMO digiLine protocol	
65	0041	char	1	r	Minor version of JUMO digiLine protocol	
128	0080	short	1	r	Hardware type basis	
129	0081	short	1	r	Software number of measuring microcontroller	
130	0082	char	1	r	Major software version of measuring microcontroller	
131	0083	char	1	r	Minor software version of measuring microcontroller	
132	0084	char	1	r	Test version of measuring microcontroller	
133	0085	char	1	r	Error index of measuring microcontroller	
134	0086	short	1	r	VDN version of measuring microcontroller	
135	0087	short	1	r	VDN number of measuring microcontroller	
136	0088	char	1	r	Hardware type of measuring microcontroller	
137	0089	char	1	r	Hardware major version of measuring microcontroller	
138	008 A	char	1	r	Hardware minor version of measuring microcontroller	

6 Modbus address tables

6.2 Measuring point info

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
512	0200	string[60]	30	r/w	Tag number	UTF-8 encoding + "ZERO" as end identifier The number of bytes must be even (If necessary, append an additional ZERO character)
542	021E	string[192]	96	r/w	Description	UTF-8 encoding + "ZERO" as end identifier The number of bytes must be even (If necessary, append an additional ZERO character)
638	027E	string[108]	54	r/w	Sensor origin	This variable is managed automatically by the JUMO digiLine master and may not be described otherwise! UTF-8 encoding + "ZERO" as end identifier The number of bytes must be even (If necessary, append an additional ZERO character)

6.3 Configuration of the interface

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
768	0300	char	1	r/w	Device address	
769	0301	enum	1	r/w	Baud rate	Enumeration: 0 = 9600 1 = 19200 2 = 38400
770	0302	enum	1	r/w	Data format	Enumertation: 0 = 8 – 1 – no parity 1 = 8 – 1 – odd parity 2 = 8 – 1 – even parity 3 = 8 – 2 – no parity
773	0305	short	1	r/w	Minimum response time	
775	0307	enum	1	r/w	Float value format	Enumeration: 0 = Standard Modbus format 1 = IEEE 754 Little Endian 2 = IEEE 754 Big Endian

6.4 Calibration data

NOTE!

The JUMO digiLine electronics saves 1 calibration logbook with 10 entry data sets. The addresses of the calibration logbook data are listed explicitly in the following tables. Since all calibration logbooks have the same storage capacity, addresses can also be calculated. The size of the address range in a calibration logbook is 0x0031 in hexadecimal or . 49 in decimal notation. Jumps from one calibration logbook to another can be calculated with multiples of this value.
Example: 0x0600 + (n - 1) * 0x0031 yields the Modbus address of the parameter "Entry type" in the n-th calibration logbook.

6.4.1 Calibration logbook entry 1

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1536	0600	enum	1	r	Entry type	Enumeration: 0 = no entry 1 = relative cell constant 2 = temperature coefficient
1537	0601	enum	1	r	Manual change	Enumeration: 0 = manual 1 = calibration
1538	0602	enum	1	r	Calibration evaluation	Enumeration: 0 = no evaluation 1 = OK 2 = warning
1539	0603	long	2	r	Operating hours counter reading	Date/time stamp
1541	0605	short	1	r	Year	
1542	0606	char	1	r	Month	
1543	0607	char	1	r	day	
1544	0608	char	1	r	Hour	
1545	0609	char	1	r	Minute	
1546	060 A	char	1	r	Seconds	
1547	060B	short	1	r	Time zone	
1548	060C	char	1	r	Sensor change counter reading	
1549	060D	float[6]	12	r	Calibration values	Array elements for entry type "relative cell constant": [0] = relative cell constant [1] = measuring range Array elements for entry type "temperature coefficient": [0] = temperature coefficient [1] = Temperature 1 [2] = Temperatuer 2 [3] = measuring range

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1561	0619	string[6][8]	24	r	Units	Calibration value units in the following format: For each unit of the 6 calibration values, 8 bytes are available for the unit character string, including the ZERO character as end identifier in UTF-8 encoding. The character string may be up to 5 characters long.

6.4.2 Calibration logbook entry 2

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1585	0631	enum	1	r	Entry type	Enumeration: 0 = no entry 1 = relative cell constant 2 = temperature coefficient
1586	0632	enum	1	r	Manual change	Enumeration: 0 = manual 1 = calibration
1587	0633	enum	1	r	Calibration evaluation	Enumeration: 0 = no evaluation 1 = OK 2 = warning
1588	0634	long	2	r	Operating hours counter reading	Date/time stamp
1590	0636	short	1	r	Year	
1591	0637	char	1	r	Month	
1592	0638	char	1	r	day	
1593	0639	char	1	r	Hour	
1594	063 A	char	1	r	Minute	
1595	063B	char	1	r	Seconds	
1596	063C	short	1	r	Time zone	
1597	063D	char	1	r	Sensor change counter reading	

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1598	063E	float[6]	12	r	Calibration values	Array elements for entry type "relative cell constant": [0] = relative cell constant [1] = measuring range Array elements for entry type "temperature coefficient": [0] = temperature coefficient [1] = Temperature 1 [2] = Temperatuer 2 [3] = measuring range
1610	064 A	string[6][8]	24	r	Units	Calibration value units in the following format: For each unit of the 6 calibration values, 8 bytes are available for the unit character string, including the ZERO character as end identifier in UTF-8 encoding. The character string may be up to 5 characters long.

6.4.3 Calibration logbook entry 3

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1634	0662	enum	1	r	Entry type	Enumeration: 0 = no entry 1 = relative cell constant 2 = temperature coefficient
1635	0663	enum	1	r	Manual change	Enumeration: 0 = manual 1 = calibration
1636	0664	enum	1	r	Calibration evaluation	Enumeration: 0 = no evaluation 1 = OK 2 = warning

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1637	0665	long	2	r	Operating hours counter reading	Date/time stamp
1639	0667	short	1	r	Year	
1640	0668	char	1	r	Month	
1641	0669	char	1	r	day	
1642	066 A	char	1	r	Hour	
1643	066B	char	1	r	Minute	
1644	066C	char	1	r	Seconds	
1645	066D	short	1	r	Time zone	
1646	066E	char	1	r	Sensor change counter reading	
1647	066F	float[6]	12	r	Calibration values	Array elements for entry type "relative cell constant": [0] = relative cell constant [1] = measuring range Array elements for entry type "temperature coefficient": [0] = temperature coefficient [1] = Temperature 1 [2] = Temperatuer 2 [3] = measuring range
1659	067B	string[6][8]	24	r	Units	Calibration value units in the following format: For each unit of the 6 calibration values, 8 bytes are available for the unit character string, including the ZERO character as end identifier in UTF-8 encoding. The character string may be up to 5 characters long.

6.4.4 Calibration logbook entry 4

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1683	0693	enum	1	r	Entry type	Enumeration: 0 = no entry 1 = relative cell constant 2 = temperature coefficient
1684	0694	enum	1	r	Manual change	Enumeration: 0 = manual 1 = calibration

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1685	0695	enum	1	r	Calibration evaluation	Enumeration: 0 = no evaluation 1 = OK 2 = warning
1686	0696	long	2	r	Operating hours counter reading	Date/time stamp
1688	0698	short	1	r	Year	
1689	0699	char	1	r	Month	
1690	069 A	char	1	r	day	
1691	069B	char	1	r	Hour	
1692	069C	char	1	r	Minute	
1693	069D	char	1	r	Seconds	
1694	069E	short	1	r	Time zone	
1695	069F	char	1	r	Sensor change counter reading	
1696	06A0	float[6]	12	r	Calibration values	Array elements for entry type "relative cell constant": [0] = relative cell constant [1] = measuring range Array elements for entry type "temperature coefficient": [0] = temperature coefficient [1] = Temperature 1 [2] = Temperatuer 2 [3] = measuring range
1708	06AC	string[6][8]	24	r	Units	Calibration value units in the following format: For each unit of the 6 calibration values, 8 bytes are available for the unit character string, including the ZERO character as end identifier in UTF-8 encoding. The character string may be up to 5 characters long.

6.4.5 Calibration logbook entry 5

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1732	06C4	enum	1	r	Entry type	Enumeration: 0 = no entry 1 = relative cell constant 2 = temperature coefficient

6 Modbus address tables

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	**Hex.**					
1733	06C5	enum	1	r	Manual change	Enumeration: 0 = manual 1 = calibration
1734	06C6	enum	1	r	Calibration evaluation	Enumeration: 0 = no evaluation 1 = OK 2 = warning
1735	06C7	long	2	r	Operating hours counter reading	Date/time stamp
1737	06C9	short	1	r	Year	
1738	06CA	char	1	r	Month	
1739	06CB	char	1	r	day	
1740	06CC	char	1	r	Hour	
1741	06CD	char	1	r	Minute	
1742	06CE	char	1	r	Seconds	
1743	06CF	short	1	r	Time zone	
1744	06D0	char	1	r	Sensor change counter reading	
1745	06D1	float[6]	12	r	Calibration values	Array elements for entry type "relative cell constant": [0] = relative cell constant [1] = measuring range Array elements for entry type "temperature coefficient": [0] = temperature coefficient [1] = Temperature 1 [2] = Temperatuer 2 [3] = measuring range
1757	06DD	string[6][8]	24	r	Units	Calibration value units in the following format: For each unit of the 6 calibration values, 8 bytes are available for the unit character string, including the ZERO character as end identifier in UTF-8 encoding. The character string may be up to 5 characters long.

6.4.6 Calibration logbook entry 6

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1781	06F5	enum	1	r	Entry type	Enumeration: 0 = no entry 1 = relative cell constant 2 = temperature coefficient
1782	06F6	enum	1	r	Manual change	Enumeration: 0 = manual 1 = calibration
1783	06F7	enum	1	r	Calibration evaluation	Enumeration: 0 = no evaluation 1 = OK 2 = warning
1784	06F8	long	2	r	Operating hours counter reading	Date/time stamp
1786	06FA	short	1	r	Year	
1787	06FB	char	1	r	Month	
1788	06FC	char	1	r	day	
1789	06FD	char	1	r	Hour	
1790	06FE	char	1	r	Minute	
1791	06FF	char	1	r	Seconds	
1792	0700	short	1	r	Time zone	
1793	0701	char	1	r	Sensor change counter reading	
1794	0702	float[6]	12	r	Calibration values	Array elements for entry type "relative cell constant": [0] = relative cell constant [1] = measuring range Array elements for entry type "temperature coefficient": [0] = temperature coefficient [1] = Temperature 1 [2] = Temperatuer 2 [3] = measuring range
1806	070E	string[6][8]	24	r	Units	Calibration value units in the following format: For each unit of the 6 calibration values, 8 bytes are available for the unit character string, including the ZERO character as end identifier in UTF-8 encoding. The character string may be up to 5 characters long.

6.4.7 Calibration logbook entry 7

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1830	0726	enum	1	r	Entry type	Enumeration: 0 = no entry 1 = relative cell constant 2 = temperature coefficient
1831	0727	enum	1	r	Manual change	Enumeration: 0 = manual 1 = calibration
1832	0728	enum	1	r	Calibration evaluation	Enumeration: 0 = no evaluation 1 = OK 2 = warning
1833	0729	long	2	r	Operating hours counter reading	Date/time stamp
1835	072B	short	1	r	Year	
1836	072C	char	1	r	Month	
1837	072D	char	1	r	day	
1838	072E	char	1	r	Hour	
1839	072F	char	1	r	Minute	
1840	0730	char	1	r	Seconds	
1841	0731	short	1	r	Time zone	
1842	0732	char	1	r	Sensor change counter reading	
1843	0733	float[6]	12	r	Calibration values	Array elements for entry type "relative cell constant": [0] = relative cell constant [1] = measuring range Array elements for entry type "temperature coefficient": [0] = temperature coefficient [1] = Temperature 1 [2] = Temperatuer 2 [3] = measuring range
1855	073F	string[6][8]	24	r	Units	Calibration value units in the following format: For each unit of the 6 calibration values, 8 bytes are available for the unit character string, including the ZERO character as end identifier in UTF-8 encoding. The character string may be up to 5 characters long.

6.4.8 Calibration logbook entry 8

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1879	0757	enum	1	r	Entry type	Enumeration: 0 = no entry 1 = relative cell constant 2 = temperature coefficient
1880	0758	enum	1	r	Manual change	Enumeration: 0 = manual 1 = calibration
1881	0759	enum	1	r	Calibration evaluation	Enumeration: 0 = no evaluation 1 = OK 2 = warning
1882	075 A	long	2	r	Operating hours counter reading	Date/time stamp
1884	075C	short	1	r	Year	
1885	075D	char	1	r	Month	
1886	075E	char	1	r	day	
1887	075F	char	1	r	Hour	
1888	0760	char	1	r	Minute	
1889	0761	char	1	r	Seconds	
1890	0762	short	1	r	Time zone	
1891	0763	char	1	r	Sensor change counter reading	
1892	0764	float[6]	12	r	Calibration values	Array elements for entry type "relative cell constant": [0] = relative cell constant [1] = measuring range Array elements for entry type "temperature coefficient": [0] = temperature coefficient [1] = Temperature 1 [2] = Temperatuer 2 [3] = measuring range
1904	0770	string[6][8]	24	r	Units	Calibration value units in the following format: For each unit of the 6 calibration values, 8 bytes are available for the unit character string, including the ZERO character as end identifier in UTF-8 encoding. The character string may be up to 5 characters long.

6.4.9 Calibration logbook entry 9

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1928	0788	enum	1	r	Entry type	Enumeration: 0 = no entry 1 = relative cell constant 2 = temperature coefficient
1929	0789	enum	1	r	Manual change	Enumeration: 0 = manual 1 = calibration
1930	078 A	enum	1	r	Calibration evaluation	Enumeration: 0 = no evaluation 1 = OK 2 = warning
1931	078B	long	2	r	Operating hours counter reading	Date/time stamp
1933	078D	short	1	r	Year	
1934	078E	char	1	r	Month	
1935	078F	char	1	r	day	
1936	0790	char	1	r	Hour	
1937	0791	char	1	r	Minute	
1938	0792	char	1	r	Seconds	
1939	0793	short	1	r	Time zone	
1940	0794	char	1	r	Sensor change counter reading	
1941	0795	float[6]	12	r	Calibration values	Array elements for entry type "relative cell constant": [0] = relative cell constant [1] = measuring range Array elements for entry type "temperature coefficient": [0] = temperature coefficient [1] = Temperature 1 [2] = Temperatuer 2 [3] = measuring range
1953	07A1	string[6][8]	24	r	Units	Calibration value units in the following format: For each unit of the 6 calibration values, 8 bytes are available for the unit character string, including the ZERO character as end identifier in UTF-8 encoding. The character string may be up to 5 characters long.

6.4.10 Calibration logbook entry 10

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
1977	07B9	enum	1	r	Entry type	Enumeration: 0 = no entry 1 = relative cell constant 2 = temperature coefficient
1978	07BA	enum	1	r	Manual change	Enumeration: 0 = manual 1 = calibration
1979	07BB	enum	1	r	Calibration evaluation	Enumeration: 0 = no evaluation 1 = OK 2 = warning
1980	07BC	long	2	r	Operating hours counter reading	Date/time stamp
1982	07BE	short	1	r	Year	
1983	07BF	char	1	r	Month	
1984	07C0	char	1	r	day	
1985	07C1	char	1	r	Hour	
1986	07C2	char	1	r	Minute	
1987	07C3	char	1	r	Seconds	
1988	07C4	short	1	r	Time zone	
1989	07C5	char	1	r	Sensor change counter reading	
1990	07C6	float[6]	12	r	Calibration values	Array elements for entry type "relative cell constant": [0] = relative cell constant [1] = measuring range Array elements for entry type "temperature coefficient": [0] = temperature coefficient [1] = Temperature 1 [2] = Temperatuer 2 [3] = measuring range
2002	07D2	string[6][8]	24	r	Units	Calibration value units in the following format: For each unit of the 6 calibration values, 8 bytes are available for the unit character string, including the ZERO character as end identifier in UTF-8 encoding. The character string may be up to 5 characters long.

6.4.11 Relative cell constants

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
25600	6400	float[4]	8	r/w	relative Ci cell constant for each measuring range	The array elements 0 to 3 correspond to the relative cell constants of measuring ranges 1 to 4
25608	6408	float	2	r/w	relative Ci cell constant for all measuring ranges	
25610	640 A	float[4]	8	r/w	relative CR cell constant for each measuring range	The array elements 0 to 3 correspond to the relative cell constants of measuring ranges 1 to 4
25618	6412	float	2	r/w	relative CR cell constant for all measuring ranges	
25620	6414	float	2	r/w	relative cell constant CR junior	

6.5 Configuration of conductivity input CR

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
20738	5102	enum	1	r/w	Compensation source	Enumeration: 0 = manual temperature 1 = temperature input 2 = interface
20739	5103	enum	1	r/w	Relative cell constant mode	Enumeration: 0 = one cell constant for all measuring ranges 1 = separate relative cell constant for each measuring range
20741	5105	enum	1	r/w	Cable break detection	Enumeration: 0 = Off 1 = On
20742	5106	float	2	r/w	Manual temperature	
20744	5108	float	2	r/w	Reference temperature for lin. TC and TC curve	
20746	510 A	float	2	r/w	Filter time	
20748	510C	float	2	r/w	Nominal cell constant	
20750	510E	enum[4]	4	r/w	Linearization table activation	Array elements [0] to [3] for measuring ranges 1 to 4 Enumeration: 0 = No 1 = Yes

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
20754	5112	enum[4]	4	r/w	Unit for calculation	Array elements [0] to [3] for measuring ranges 1 to 4 Enumeration: 0 = µS/cm 1 = mS/cm 2 = kΩ×cm 3 = MΩ×cm
20758	5116	T_-KOMP_CR [4]	4	r/w	Compensation	The array elements [0] to [3] correspond to the enumerations for the compensation settings of measuring ranges 1 to 4. Enumeration: 0 = None 1 = TC linear 2 = TC curve 3 = natural water 4 = natural water with expanded temperature range 5 = TDS 6 = ASTM neutral 7 = ASTM acidic 8 = ASTM alkaline
20762	511 A	float[4]	8	r/w	TK	The array elements [0] to [3] correspond to the temperature coefficients of measuring ranges 1 to 4
20770	5122	float[4]	8	r/w	Offset	The array elements [0] to [3] correspond to the offsets of measuring ranges 1 to 4
20778	512 A	float[4]	8	r/w	TDS factor	The array elements [0] to [3] correspond to the TDS factors of measuring ranges 1 to 4
20786	5132	string[4][6]	12	r/w	Units text for TDS and customer-specific linearization with linearization tables	The array elements [0][x] to [3][x] correspond to the TDS units texts for measuring ranges 1 to 4 in ISO-8859-1 encoding with up to 6 characters, including "ZERO" as end identifier.

6 Modbus address tables

6.6 Configuration of conductivity input Ci

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	**Hex.**					
20994	5202	enum	1	r/w	Compensation source	Enumeration: 0 = manual temperature 1 = temperature input 2 = interface
20995	5203	enum	1	r/w	Relative cell constant mode	Enumeration: 0 = one cell constant for all measuring ranges 1 = separate relative cell constant for each measuring range
20996	5204	float	2	r/w	Manual temperature	
20998	5206	float	2	r/w	Reference temperature for lin. TC and TC curve	
21000	5208	float	2	r/w	Filter time	
21002	520 A	float	2	r/w	Nominal cell constant	
21004	520C	float	2	r/w	Install. factor	
21006	520E	enum[4]	4	r/w	Linearization table activation	Array elements [0] to [3] for measuring ranges 1 to 4 Enumeration: 0 = No 1 = Yes
21010	5212	enum[4]	4	r/w	Unit for calculation	Array elements [0] to [3] for measuring ranges 1 to 4 Enumeration: 0 = µS/cm 1 = mS/cm 2 = kΩ×cm 3 = MΩ×cm

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
21014	5216	enum[4]	4	r/w	Compensation	Array elements [0] to [3] for measuring ranges 1 to 4 Enumeration: 0 = None 1 = TC linear 2 = TC curve 3 = natural water 4 = natural water with expanded temperature range 5 = TDS 6 = NaOH 0 to 12 % 7 = NaOH 25 to 50 % 8 = HNO_3 0 to 25 % 9 = HNO_3 36 to 82 % 10 = H_2SO_4 0 to 28% 11 = H_2SO_4 36 to 85% 12 = H_2SO_4 92 to 99% 13 = HCl 0 to 18 % 14 = HCl 22 to 44 % 15 = NaCL 0 to 25 % 16 = $MgCl_2$ 0 to 17.5 % 17 = $MgCl_2$ 18.5 to 25 %
21018	521 A	float[4]	8	r/w	TK	The array elements [0] to [3] correspond to the temperature coefficients of measuring ranges 1 to 4
21026	5222	float[4]	8	r/w	Offset	The array elements [0] to [3] correspond to the offsets of measuring ranges 1 to 4
21034	522 A	float[4]	8	r/w	TDS factor	The array elements [0] to [3] correspond to the TDS factors of measuring ranges 1 to 4
21042	5232	string[4][6]	12	r/w	Units text for TDS and customer-specific linearization with linearization tables	The array elements [0][x] to [3][x] correspond to the TDS units texts for measuring ranges 1 to 4 in ISO-8859-1 encoding with up to 6 characters, including "ZERO" as end identifier.

6 Modbus address tables

6.7 Configuration of temperature input

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
21504	5400	float	2	r/w	Filter time	Unit: seconds
21506	5402	float	2	r/w	Offset	
21509	5405	enum	1	r/w	Connection type	Enumeration: 0 = 2-wire 1 = 3-wire
21510	5406	enum	1	r/w	Temperature input activation	Enumeration: 0 = Inactive 1 = Active

6.8 Configuration of calibration timers

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
22784	5900	short	1	r/w	Calibration interval	Unit: Days

6.9 Configuration of sensor stress

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
23040	5A00	enum	1	r/w	Sensor stress alarm activation	Enumeration: 0 = Inactive 1 = Active

6.10 Configuration of sensor monitoring

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
23104	5A40	float	2	r/w	CIP temperature	
23106	5A42	short	1	r/w	CIP cycle duration	Unit: Days
23107	5A43	float	2	r/w	SIP temperature	
23109	5A45	short	1	r/w	SIP cycle duration	Unit: Days
23110	5A46	enum	1	r/w	Sensor monitoring alarm	Enumeration: 0 = Active 1 = Inactive
23111	5A47	short	1	r/w	Maximum number of CIP cycles until alarm	
23112	5A48	short	1	r/w	Maximum number of SIP cycles until alarm	
23113	5A49	short	1	r/w	Maximum number of CIP cycles until pre-alarm	

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
23114	5A4A	short	1	r/w	Maximum number of SIP cycles until pre-alarm	

6.11 Configuration of display

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
23809	5D01	enum	1	r/w	Language	Enumeration: 0 = German 1 = English 2 = French 3 = Spanish
23810	5D02	short	1	r/w	Automatic logoff time	Unit: Minutes 0 = No automatic logoff
23811	5D03	enum	1	r/w	Display type	Enumeration: 0 = Standard display 1 = Large display 2 = Bargraph
23812	5D04	enum	1	r/w	Temperature Unit	Enumeration: 0 = °C 1 = °F
23813	5D05	float[4]	8	r/w	Bargraph start	The array elements [0] to [3] correspond to the initial values of the bargraph for measuring ranges 1 to 4
23821	5D0D	float[4]	8	r/w	Bargraph end	The array elements [0] to [3] correspond to the final values of the bargraph for measuring ranges 1 to 4
23829	5D15	enum[4]	4	r/w	Decimal place	Array elements [0] to [3] for measuring ranges 1 to 4 Enumeration: 0 = XXXX 1 = XXX.x 2 = XX.xx 3 = X.xxx
23835	5D1B	short	1	r/w	Display contrast	

6 Modbus address tables

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	**Hex.**					
23836	5D1C	enum	1	r/w	Signal for primary value display	Enumeration 0 = None 1 = Temperature input 2 = Compensation temperature 3 = Uncompensated conductivity 4 = Compensated conductivity
23837	5D1D	enum	1	r/w	Signal for secondary value display	Enumeration 0 = None 1 = Temperature input 2 = Compensation temperature 3 = Uncompensated conductivity 4 = Compensated conductivity

6.12 Process values

6.12.1 Inputs

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
8192	2000	float	2	r	Uncompensated conductivity measured value in micro-Siemens	
8194	2002	float	2	r	Uncompensated conductivity measured value	
8196	2004	float	2	r	Compensated conductivity measured value	
8198	2006	float	2	r	Compensation temperature value	
8200	2008	short	1	r	Current measuring range for conductivity measured value	
8202	200 A	char	1	r	Compensation range left	
8203	200B	char	1	r	Invalid conductivity	0 = No error present 1 = Invalid conductivity
8204	200C	string[6]	3	r	Units text uncomp. conductivity	ASCII encoding with up to 6 characters, including "ZERO" as end identifier. The number of bytes must be even (If necessary, append an additional ZERO character)
8207	200F	string[6]	3	r	Units text comp. conductivity	ASCII encoding with up to 6 characters, including "ZERO" as end identifier. The number of bytes must be even (If necessary, append an additional ZERO character)
8210	2012	short	1	r/w	Current measuring range for conductivity measured value via interface	0 = Measuring range 1 1 = Messbereich 2 2 = Messbereich 3 3 = Messbereich 4
8211	2013	float	2	r/w	Compensation temperature interface	Compensation temperature transmitted from JUMO digiLine master to JUMO digiLine CR/Ci
8450	2102	float	2	r	Measured value for temperature input in °C	Temperature measured value from JUMO digiLine CR/Ci in °C
8452	2104	float	2	r	Measured value for temperature input in °F	Temperature measured value from JUMO digiLine CR/Ci in °F
8454	2106	char	1	r	Temperature input alarm	0 = No error present 1 = Temperature input alarm

6.12.2 Calibration alarm

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
9472	2500	char	1	r	Calibration alarm	

6.12.3 Sensor stress

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
9728	2600	char	1	r	Sensor stress alarm state	The sensor stress level indicates the thermal and chemical stress on the sensor in the form of an index as a function of temperature and conductivity measured values. You can find more on this in the operating manual for the JUMO digiLine CR/Ci.
9729	2601	char	1	r	Sensor stress pre-alarm state	
9730	2602	short	1	r	Current sensor stress level	

6.12.4 Sensor monitoring

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
9792	2640	char	1	r	CIP/SIP alarm	
9793	2641	char	1	r	CIP/SIP pre-alarm	

6.12.5 Date and time for time stamp

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
10304	2840	short	1	r/w	Year	
10305	2841	char	1	r/w	Month	
10306	2842	char	1	r/w	day	
10307	2843	char	1	r/w	Hour	
10308	2844	char	1	r/w	Minute	
10309	2845	char	1	r/w	Second	
10310	2846	short	1	r/w	Time zone	Offset for UTC in minutes

6.12.6 Summary of important process values

NOTE!

The variables from the summarized process values are set to invalid during the boot phase to prevent unstable measured values from creating problems in masters that access these measured values. In contrast, the original variables are always represented transparently (i.e. fluctuating).

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	Hex.					
32768	8000	float	2	r	Uncompensated conductivity measured value	
32770	8002	float	2	r	Compensated conductivity measured value	
32772	8004	float	2	r	Temperature in °C	
32774	8006	short	1	r	Current parameter block	
32775	8007	short	1	r	Status word	The status word is a bit field consisting of 16 status bits. The structure of the status word is shown in the following table.
32776	8008	short	1	r	Sensor stress level	The sensor stress level indicates the thermal and chemical stress on the sensor in the form of an index as a function of temperature and conductivity measured values. You can find more on this in the operating manual for the JUMO digiLine CR/Ci.
32777	8009	long	2	r	CIP cycle counter	
32779	800B	long	2	r	SIP cycle counter	

6 Modbus address tables

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Encoding
Dec.	**Hex.**					
32781	800D	long	2	r	Total overall duration of the lower temperature limit value violation	Unit: seconds
32783	800F	long	2	r	Total overall duration of the upper temperature limit value violation	
32785	8011	long	2	r	Maximum total overall duration of the upper conductivity limit value violation	
32787	8013	long	2	r	Number of lower temperature value violations	
32789	8015	long	2	r	Number of upper temperature limit value violations	
32791	8017	long	2	r	Number of upper conductivity limit value violations	

Structure of the status word and meaning of its 16 status bits

Bit number	#16 (msb)	#15	#14	#13	#12	#11	#10	#9
Function	Calibration data error	Configuration data error	-	-	-	-	-	Operating data acquisition alarm

#8	#7	#6	#5	#4	#3	#2	#1 (lsb)
Calibration timer expired	Sensor stress alarm	Sensor stress pre-alarm	Max. CIP/ SIP cycles alarm	Max. CIP/ SIP cycles pre-alarm	Compensation temperature range left	Invalid temperature measured value alarm	Invalid conductivity measured value alarm

JUMO GmbH & Co. KG
Street address:
Moritz-Juchheim-Straße 1
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
Email: mail@jumo.net
Internet: www.jumo.net

JUMO Instrument Co. Ltd.
JUMO House
Temple Bank, Riverway
Harlow, Essex, CM20 2DY, UK
Phone: +44 1279 63 55 33
Fax: +44 1279 62 50 29
Email: sales@jumo.co.uk
Internet: www.jumo.co.uk

JUMO Process Control, Inc.
6733 Myers Road
East Syracuse, NY 13057, USA
Phone: +1 315 437 5866
Fax: +1 315 437 5860
Email: info.us@jumo.net
Internet: www.jumousa.com