JUMO tecLine...

... Cl2 (202630), TC (202631), ClO2+O3 (202634), H2O2+PAA (202636), Br (202637), Cl2 OM (202681)

Modbus interface description

20263000T92Z001K000

V2.00/EN/00718260/2022-03-24

Contents

Contents

1 Safety information

1.1 Warning symbols

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.

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!

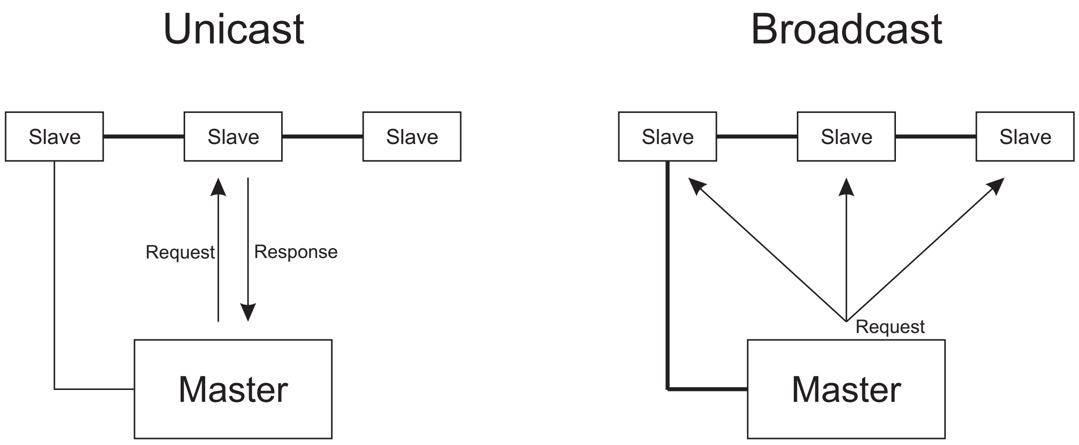
At the end of its service life, the device and any batteries present do not belong in the trash! Please ensure that they are **disposed of** properly and in an **environmentally friendly** manner.

2 Modbus protocol description

2.1 Master-slave principle

Communication between a master (e.g. a SCADA system or PLC) and a JUMO tecLine sensor as a slave via Modbus takes place on the basis of the master-slave principle in the form of a data request/ instruction – response. Slaves are identified by the device address. Master devices do not need an address.

⇨ chapter 3 "Modbus over a serial interface", Page 15



The master controls data exchange by cyclically requesting the slaves in the overall bus. The slaves (e.g. JUMO tecLine sensors) only have a response function. 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 transfer data from slave to slave, the master must extract data from one slave and then transfer it to the next.

As a rule, the master directs its requests specifically to individual slaves. For this, it must communicate with the relevant slaves by means of their unicast address. Requests can also be directed to all slaves in the bus 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 requests. Data collisions would otherwise result in serial bus systems. For this reason, use of broadcast addresses is only appropriate with function codes for writing data. Broadcasts cannot be used with function codes for reading data.

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). JUMO tecLine sensors only support the **RTU mode**. Here, the data is transferred via the serial bus in binary format (RS485).

⇨ chapter 3 "Modbus over a serial interface", Page 15

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	n bytes	2 bytes

Each 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. 125 words (250 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	Address of first word	Number of words **n**	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave address	Function 0x03	Number of bytes read	Word value(s)	Checksum CRC
1 byte	1 byte	1 byte	**n** × 2 bytes	2 bytes

Example

Reading the measured temperature value. The example deals with the measured temperature value (24.091 °C in this example), which can be extracted at the start address 0x0004 in all JUMO tecLine sensors 20263x. As a floating-point number (float), the value has a length of 4 bytes. Accordingly, two words (Modbus register) must be read (cf. "Process data measured values", Page 19).

Hex code for the data request:

01	03	00 04	00 02	85 CA
Slave	Function	Address of 1st word	Number of words	CRC

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

01	03	04	BA 2F 41 C0	DE E2
Slave	Function	Bytes read	Temperature value 24.091 °C (floating point)	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 to set the baud rate to 38400 Bd is to be sent to a JUMO tecLine CIO2 (202634). The slave address of the device is 1 here; the word address is 0x0401 (⇨ "Device data Modbus", Page 18) and the value to be written is "4" (value for setting the baud rate to 38400 Bd).

Hex code for the instruction:

01	06	04 01	00 04	D8 F9
Slave	Function	Word address	Value	CRC

Hex code for the response:

01	06	04 01	00 04	D8 F9
Slave	Function	Word address	Value	CRC

2.4.3 Writing n words

This function is used to write n words starting at a specific address.

Instruction

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

Response

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

Example

Write the sensor slope 153 nA/ppm at the Modbus address 0x0208 in the case of a JUMO tecLine H2O2+O3 (202634).

⇨ "Process data parameter", Page 19

Hex code for the instruction:

01	10	02 08	00 02	04	00 00 43 19	1B 93
Slave	Function	Address of 1st word	Number of words	Number of bytes	Slope 153 nA/ppm (float)	CRC

Hex code for the response:

01	10	02 08	00 02	C1 B2
Slave	Function	Address of 1st word	Number of words	CRC

2.5 Data types

Data type	Description	Access	Possible function codes	Number of Modbus registers

Float	2 words as 32-bit floating-point number with coding according to IEEE 754, keeping in mind that the order in which the bytes are transferred depends on the Modbus implementation of a device. JUMO tecLine sensors do **not** (20263x) transfer the bytes in the order specified by the IEEE 754 standard coding, but rather in the typical order for float variables (byte 3, byte 4, byte 1, byte 2).
	S = Prefix sign bit
	E = Exponent (two's complement)
	M = 23-bit normalized mantissa
		read only	03, 04	2

		read/ write	03, 04, 16	

IEEE 754 standard coding

Byte 1	Byte 2	Byte 3	Byte 4
$SE_7EEEEEE_1$	$E_0M_{23}MMMMMM_{16}$	$M_{15}MMMMMMM_8$	$M_7MMMMMMM_0$

Typical modbus coding of floating-point variables

Address of the 1st Modbus register for variables		Address of the 2nd Modbus register for variables	
Byte 3	Byte 4	Byte 1	Byte 2
$M_{15}MMMMMMM_8$	$M_7MMMMMMM_0$	$SE_7EEEEEE_1$	$E_0M_{23}MMMMMM_{16}$

When creating customer-specific applications, the correct byte order in storage format needs to be checked. Many compilers use the following storage format:

Compiler coding

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

Data type	Description	Access	Possible function codes	Number of Modbus registers
int	Word (16-bit) as unsigned integer value. The higher-value byte (MSB) is transferred before the lower-value byte (LSB). Value range: 0 to 65,535	read only	03, 04	1
		read/ write	03, 04, 16	
unsigned long int	Double word (32-bit) as an unsigned integer value. MSB is transferred before LSB. Value range: 0 to 4,294,967,295	read only	03, 04	2
		read/ write	03, 04, 16	
char [...]	Character/byte (8-bit) as unsigned integer value. 2 characters are contained in 1 word. The 1st character is transferred in MSB, the 2nd character in LSB. Value range: 0 to 255	read only	03, 04	1

2.6 Examples of data transmission

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

Data request

Slave address	Function 0x03	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

Slave address	Function 0x03	Number of bytes read	Word value(s)	Checksum CRC
1 byte	1 byte	1 byte	n bytes	2 bytes

2.6.1 Integer values (16-bit)

Example

In this example, the status of the firmware of a JUMO tecLine Cl2 (202630) at address 0x0309 (⇨ "Device data", Page 18) is to be extracted. The value in this example should be 1410 (1,410), i.e. 582 in hexadecimal.

Data request:

01	03	03 09	00 01	54 4C
Slave	Function	Address of 1st word	Number of words	CRC

Response (value in the Modbus int format):

01	03	02	05 82	3B 75
Slave	Function	Bytes read	Integer value	CRC

2.6.2 Long integer values (32-bit)

Example

In this example, the time stamp of the latest calibration in the calibration memory (calibration 0) of a JUMO tecLine Cl2 (202630) is to be extracted at address 0x0214 (⇨ "Process data parameter", Page 19). The value in this example is to correspond to the date 03/08/2019, 13:10 p.m., i.e. 1903081310 in decimal and thus 716EB75E in hexadecimal.

Data request:

01	03	02 14	00 02	85 B7
Slave	Function	Address of 1st word	Number of words	CRC

Response (value in the Modbus int format):

01	03	04	71 6E B7 5E	76 DA
Slave	Function	Bytes read	Long integer value	CRC

2.6.3 Float values

JUMO tecLine sensors (20263x) work with floating-point values with an order that is **different** from IEEE-754 standard formats
(32-bit).

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 in the case of JUMO tecLine sensors (20263x)

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

Example

In this example, the concentration value of the disinfectant is to be extracted at address 0x0000 of the device. The value should be 0.168 ppm (0x08313E2C in reverse IEEE 754 format) here.

Data request:

01	03	00 00	00 02	C4 0B
Slave	Function	Address of 1st word	Number of words	CRC

Response (values in Modbus floating-point format):

01	03	04	08 31 3E 2C	B8 21
Slave	Function	Bytes read	Floating-point value	CRC

Many compilers (e.g. 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.

2.7 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                        else
        CRC = CRC XOR 0xA001
while (not all BytesOfMessage processed);
```

NOTE!

The low byte of the checksum is transmitted first!
Example: The CRC16 checksum DB 25 is transmitted and represented in the sequence 25 DB.

Example

Query unit of the output of the value for concentration at address 0x0200:

Instruction: read a word from address 0x0200

01	03	02 00	00 01	85 B2
Slave	Function	Address	Read one word	CRC

Response (CRC16 = 0x45F8)

01	03	02	00 03	F8 45
Slave	Function	Number of bytes	Word 1	CRC

Word 1 = 3 stands for the unit "ppm".

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 JUMO tecLine (20263x) are listed in chapter 2.4 "Function codes", Page 7.
- 02 = Invalid address or too many words or bits are to be read or written
- 03 = The format of the data cannot be read.
- 255 = A communication problem exists.

Modbus error responses can be identified in that the MSB of the function code has been set to 1.

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):

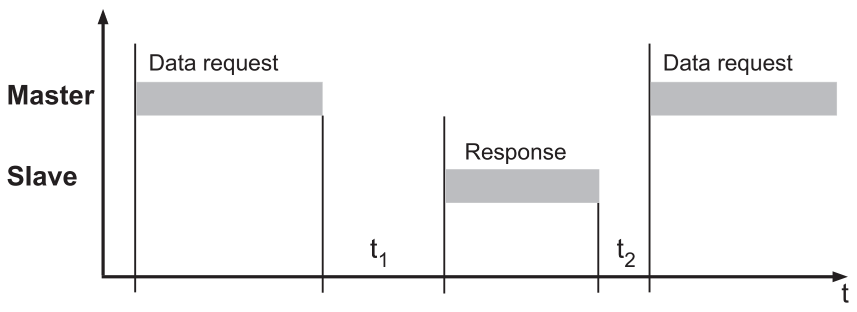
01	86	02	C3 A1
Slave	OR function	Errors	CRC

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

3.1 Modbus slave operation via RS485 serial interface

Chronological sequence of communication

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



t_1	Waiting period 1, consisting of 3.5 character transmission times, plus the time that the slave (sensor) requires for processing the data request
t_2	Waiting period 2 that the master has to observe before it starts a new data request: 3.5 character transmission times

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 end identifier after a data request or data response is 1.5 characters long. The duration of these 1.5 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 for transmission of a character) depends on the baud rate.
The following results in the case of a data format of 8 data bits, no parity bit, and one stop bit:

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

For all other data formats that can be set for the sensor, the following is the result:

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

Sample calculation for 9600 baud:

Baud rate [baud]	**Data format** [bit]	**Character transmission time**[ms]
9600	11	1.146
	10	1.042

Waiting period

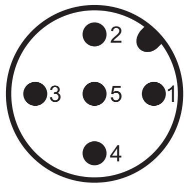
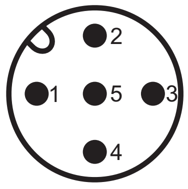
The waiting period is, as already described in "Chronological sequence of communication", always 3.5 character transmission times. It starts after the identifier for end of data request or end of response respectively.

Sample calculation for 9600 baud:

Baud rate [baud]	**Data format** [bit]	**Waiting period**[ms]
9600	11	4.011
	10	3.647

4 Interfaces

4.1 Interface assignment of the JUMO tecLine sensors (20263x)

5-pole M12 plug connector, A-coded		
Pin	**Potential**	**Symbol**
1	not connected	Connector  Socket 
2	+24 V	
3	GND	
4	RS485 B (RxD/TxD-)	
5	RS485 A (RxD/TxD+)	
The connection to the serial interfaces of a master device with screw or spring-cage terminals can be made using a JUMO M12 digiLine master connecting cable.		

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 tecLine sensors (20263x) can be made on a PC using the JUMO DSM software. How to use the JUMO DSM software is described in detail in the operating manual of the JUMO DSM software.

5.1 Settings for the serial interface

For all user devices on a bus to be able to communicate with one another, their interface settings must match. The following table shows the settings options for JUMO tecLine sensors (20263x).

Configuration item	Selection/settings	Description
Baud rate	2400 4800 9600 19200 **38400** 57600 115200	Transmission speed (symbol rate) of the serial interface
Data format	8 - 2 - none 8 - 1 - even 8 - 1 - odd **8 - 1 - none**	Format of the data word Useful bit - stop bit - parity
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.

6 Modbus address tables

NOTE!

The addresses specified in the tables below can, depending on access, be read, or read and described. No programming of the sensor is required for theses accesses.

Modbus functions

Address	Function	Parameter
0x03	Read Holding Registers (16-bit)	Use for all parameters
0x04	Read Input Register (16-bit)	
0x06	Write Single Register	
0x10	Write Multiple Registers	

Device data

Address	Access	Data type	Parameter	Example
0x0308	R/O	int	Hardware	1130 (1,130)
0x0309	R/O	int	Firmware	1503 (1,503)
0x030A	R/O	float	Nominal slope	7.5
0x030C	R/O	char [20]	F no.	01047146010191200001
0x0317	R/O	char [10]	Part no.	00705172

Device data Modbus

Address	Access	Data type	Parameter	Value range		Default	
0x0400	R/W	int	Slave address	1 to 247		20	202630 (Cl2)
						30	202631 (TC)
						50	202634 (O3)
						80	202634 (ClO2)
						60	202636 (H2O2)
						70	202636 (PAA)
						90	202637 (Br)
						100	202681 (Cl2 OM)
0x0401	R/W	int	Baud rate	0	2400	4	
				1	4800		
				2	9600		
				3	19200		
				4	**38400**		
				5	57600		
				6	115200		
0x0402	R/W	int	Parity/stop bit	0	none/2	3	
				1	even/1		
				2	odd/1		
				3	**none/1**		

6 Modbus address tables

Address	Access	Data type	Parameter	Value range		Default
0x0200	R/O	int	Unit	0	%	
				1	‰	
				2	g/l	
				3	ppm	
				4	mg/l	
				5	ppb	
				6	µg/l	
0x0201	R/O	int	Decimal places	0	0000	
				1	000.0	
				2	00.00	
				3	0.000	
0x0206	R/W	float	X_Zero			
0x0208	R/W	float	X_Span			
0x020A	R/W	unsigned long int	DateTime	yymmddhhmm		
History						
0x0210	R/O	float	X_Zero (0)			
0x0212	R/O	float	X_Span (0)			
0x0214	R/O	unsigned long int	DateTime (0)	yymmddhhmm		
0x0216	R/O	float	X_Zero (1)			
0x0218	R/O	float	X_Span (1)			
0x021A	R/O	unsigned long int	DateTime (1)	yymmddhhmm		
0x021C	R/O	float	X_Zero (2)			
0x021E	R/O	float	X_Span (2)			
0x0220	R/O	unsigned long int	DateTime (2)	yymmddhhmm		
0x0212	R/O	float	X_Zero (3)			
0x0214	R/O	float	X_Span (3)			
0x0226	R/O	unsigned long int	DateTime (3)	yymmddhhmm		
0x0228	R/O	float	X_Zero (4)			
0x022A	R/O	float	X_Span (4)			
0x022C	R/O	unsigned long int	DateTime (4)	yymmddhhmm		
0x022E	R/O	float	Measuring range			20

Process data measured values

Address	Access	Data type	Parameter
0x0000	R/O	float	Concentration (in selected unit)
0x0002	R/O	float	Cell current (in nA at 25 °C)
0x0004	R/O	float	Temperature (in °C)

7 Annex

7.1 Supplementary information

Definitions	• **X_Zero** (nA)	Current if no disinfectant is present in the medium to be measured. The start address for this value is 0x0206.
	• **X_Span** (nA/unit)	Current value of the current calibration in relation to the concentration of the disinfectant with the current calibration. The start address for this value is 0x0208. You obtain the unit by extracting the address 0x0200.
	• **Concentration**	The software calculates the concentration as follows: $\frac{\text{present current - X\_Zero}}{\text{X\_Span}}$
Date/time stamp	• Data type: unsigned long • Largest representable number: 2^{32} = 42|94|96|72|96, correspondingly (theoretical) Year: (20)42, Month: 94, Day: 96, Hour: 72, Minute: 96 • Last possible date: 12/31/2042, 23:59 p.m. (42|12|31|23|59)	

7.2 Operating principle of the history

As can be seen in the Modbus address tables of the sensor (⇨page 18), the **current calibration data** is stored at the addresses 0x0206 to 0x020A.

The storage system of the history works according to the following rules:

- A calibration that has been carried out becomes active when the "date/time" (date stamp) is written.
- If the current calibration has a larger (later) date stamp than the last calibration, this is stored at the addresses for the current calibration (0x0206 to 0x020A).
  - The data from the **current calibration** is also stored at the addresses 0x0210 to 0x0214 (X_Zero (0), X_Span (0) und Date/Time (0)), **i.e. at position 0 of the history**.
  - The data from the **old calibration** is shifted **from the previous position 0** of the history **to position 1**; **all other data** is also shifted on **by one position** respectively.
  - The previous **oldest** data (position 4) cannot be shifted on, rather it is **deleted**.
- If the current calibration has a date stamp that is **smaller or the same as the last calibration**, the data from the last calibration (at the addresses 0x0206 to 0x020A) and the data at position 0 of the history (at the addresses 0x0210 to 0x0214) are **overwritten.**
  The data from positions 1 to 4 in the history remain unaffected in this case!

NOTE!
Since the data stamp has the minute as the smallest unit, the time between 2 calibrations should be at least 1 minute.

7.3 Calibration of the sensor slope

7.3.1 Checking of the sensor slope at the measuring point

NOTE!

Observe the calibration information in the section on calibration from the operating manual for your sensor.

1. Ensure that disinfectant is present at the measuring point and its concentration is stable.
2. Wait until the signal at the sensor is stable.

NOTE!

Bear in mind the settling time specified in the data sheet for your sensor.

3. Extract the cell current (in nA) at the address 0x0002. Note the value.
4. Take a sample of the measurement water at a position close to the installation location of the sensor.
5. Measure the concentration of the disinfectant as fast as possible using a reliable and precise procedure, depending on the sensor type, e.g. with a photometer or using titration.

NOTE!

Do ***not*** use a colorimeter to measure the concentration of the disinfectant.
A colorimeter is only an indicator. It is not precise enough for calibration!

6. Calculate the slope of the sensor by dividing the previously noted cell current by the measured concentration of disinfectant.

NOTE!

If a current that deviates from zero is determined for the sensor zero point, it must be subtracted before dividing the noted cell current.

7.3.2 Save the calibration data in the sensor

1. Write the value "0" at the address 0x0206. In the event of a zero point calibration, write the current value in the case of disinfectant-free process water determined in chapter 7.4 "Calibration of the sensor zero point", Page 23 in the address 0x0206.
2. Write the value for the slope calculated in chapter 7.3.1 "Checking of the sensor slope at the measuring point", Page 21 at the address 0x0208.
3. Write the value for "time/date" at the address 0x020A.

   The structure for this variable is "YYMMDDHHMM". An example for "date/time" is 1904101430 (see also chapter 7.1 "Supplementary information", Page 20).

   By writing "date/time", the calibration is active.

7.3.3 Checking the calibration

Software checking

Check the values for the zero point and slope by extracting the values in the addresses 0x0210 (X_Zero (0)) and 0x0212 (X_Span (0)).

The value "0" (or the value from the zero point calibration from chapter 7.4 "Calibration of the sensor zero point", Page 23, if this has been carried out) should be at address 0x0210.

The value from the slope calculation from chapter 7.3.1 "Checking of the sensor slope at the measuring point", Page 21 should be at address 0x0212.

Chemical check

The concentration measured by the sensor and the analytically determined concentration should be broadly identical.

7.3.4 Regular check

Check the calibration regularly, ideally weekly, by carrying out the steps from chapter 7.3.1 "Checking of the sensor slope at the measuring point", Page 21.

If you identify that the currently determined slope is lower than 30 % of the nominal slope (value at address 0x030A), follow the maintenance instructions in the operating manual for your sensor.

NOTE!

Use the calibration logbook (history) in order to view values for the last 5 calibrations.
Be aware that the 6th calibration overwrites the previous oldest calibration in the calibration logbook.

7.4 Calibration of the sensor zero point

NOTE!

Calibration of the sensor zero point is generally not required and is not recommended by us!

Nevertheless, after a long deployment time, it may be necessary to calibrate the zero point of the sensor under certain circumstances.

7.4.1 Measurement of the cell current where disinfectant is absent

Measurement of the cell current with a sensor in the measuring circuit

If the measuring circuit can be operated without disinfectant, removing the sensor from the measuring circuit is not necessary.

NOTE!

Remember that bacteria and germs can multiple if disinfectant is not present! Keep the operation time without disinfectant as short as possible.

1. Ensure that the sensor is in a completely disinfectant-free medium.
2. Wait until the sensor signal is stable.
3. Read the cell current (in nA) at address 0x0002 and note it.
4. Carry out a slope calibration as described from chapter 7.3.1 "Checking of the sensor slope at the measuring point", Page 21.

Measurement of the cell current with the sensor removed

If the measuring circuit **cannot** be operated without disinfectant, remove the sensor and keep it in a beaker filled with water.

NOTE!

Do not completely submerge the sensor under water!
Place the sensor without the membrane on the base of the beaker!
Move the sensor during the measurement in the beaker or use a magnetic stirrer.

1. Ensure that the sensor is in completely disinfectant-free water.
2. Ensure that the water in the beaker is the same temperature as the water contained in the measuring circuit.
3. Wait until the sensor signal is stable.
4. Read the cell current (in nA) at address 0x0002 and note it.
5. Carry out a slope calibration as described from chapter 7.3.1 "Checking of the sensor slope at the measuring point", Page 21.

7.5 Restoring the nominal slope

7.5.1 Setting the standard values for nominal slope and zero point

NOTE!

The sensor is delivered ex works uncalibrated. If the sensor is new and has not yet been calibrated, the nominal slope setting described below is not required.

1. Extract the nominal slope at the address 0x030A.

   Examples of nominal slopes are e.g. 22 nA/ppm in the case of a sensor for free chlorine with the measuring range 0 to 200 ppm (202630/53-45), or 7.5 nA/ppm in the case of a sensor for total chlorine with the measuring range 0 to 20 ppm (202631/52-37).
2. Write the nominal slope at the address 0x0208.

   With this step, the replacement of the slope previously determined through calibration by the nominal slope and the storage of the last calibration in the history is prepared.
3. Write the value "0" at the address 0x0206. This is the standard value for the zero point.
4. Write the date stamp at the address 0x020A.

   The structure of the date stamp is explained in chapter 7.1 "Supplementary information", Page 20.

   By writing the date stamp, the setting for the nominal slope and the storage of the last calibration in the history is completed.

7.5.2 Checking of the transferred value for the sensor slope

NOTE!

If the sensor is new and has not yet been calibrated, the nominal slope check described below is not required.

1. Check the values at the addresses 0x0210 (X_Zero (0)) and 0x0212 (X_Span (0)). The value at address 0x0210 should be "0", while the value for the nominal slope should be at 0x0212. In this example, the value "22" for the sensor 202630/53-45.

JUMO

OneTemp pty ltd
measure | control | record
1300 768 887
www.onetemp.com.au