

# JUMO ecoLine O-DO/NTU

Digital sensors for turbidity and oxygen measurement



Modbus interface description



20261300T92Z003K000

V1.01/EN/00692386



<b>1</b>	<b>Safety information</b> . . . . .	<b>5</b>
1.1	Warning symbols . . . . .	5
1.2	Note signs . . . . .	5
<b>2</b>	<b>Modbus protocol description</b> . . . . .	<b>7</b>
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 . . . . .	8
2.4.2	Writing one word . . . . .	9
2.4.3	Writing n words . . . . .	10
2.5	Data types . . . . .	11
2.6	Examples of data transmission . . . . .	12
2.6.1	Integer values . . . . .	12
2.6.2	Float values . . . . .	13
2.7	Checksum (CRC16) . . . . .	14
2.8	Error messages . . . . .	15
2.8.1	Modbus error codes . . . . .	15
<b>3</b>	<b>Modbus over a serial interface</b> . . . . .	<b>17</b>
3.1	Modbus slave operation via RS485 serial interface . . . . .	17
<b>4</b>	<b>Interfaces</b> . . . . .	<b>19</b>
4.1	Interface assignment on a JUMO ecoLine O-DO/NTU . . . . .	19
<b>5</b>	<b>Configuring interfaces</b> . . . . .	<b>21</b>
5.1	Settings for the serial interface . . . . .	21
<b>6</b>	<b>Functional sequence</b> . . . . .	<b>23</b>
6.1	Measuring operation on the JUMO ecoLine O-DO (type 202613) . . . . .	24
6.2	Measuring operation on the JUMO ecoLine NTU (type 202670) . . . . .	28

---

# Contents

---

---

## 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

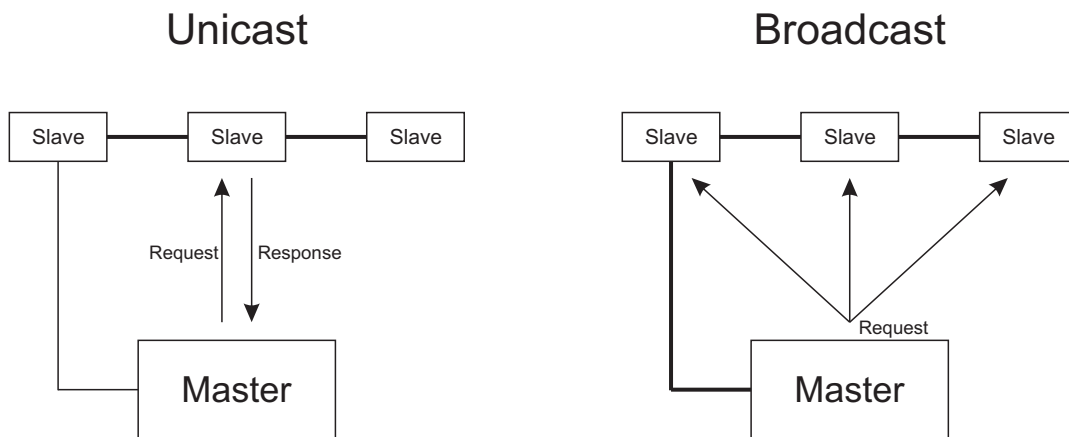
---

## 2 Modbus protocol description

### 2.1 Master-slave principle

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

⇒ chapter 3 "Modbus over a serial interface", page 17



The master controls data exchange by cyclically querying the slaves on the overall bus. The slaves (e.g. JUMO ecoLine 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 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.

# 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 (Remote Terminal Unit)** and ASCII mode (transmission of data in ASCII format). JUMO ecoLine sensors support only the **RTU mode**. Here, the data are transmitted via the serial bus in binary format (RS422/485).

⇒ chapter 3 "Modbus over a serial interface", page 17

## 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:

<b>Slave address</b>	Device address of a specific slave
<b>Function code</b>	Function selection (read/write words)
<b>Data field</b>	Contains the information (depending on the function code) - word address/bit address - number of words/number of bits - word value(s)/bit value(s)
<b>Checksum</b>	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	3	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 x	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	2 x bytes	2 bytes



# 2 Modbus protocol description

## Example

Reading the measured temperature value. The example here is for the measured temperature value (25.3 °C in the example), which can be read at the start address 0x0053 in both JUMO ecoLine sensors. As a floating-point number (float), the value has a length of 4 bytes. According, two words (Modbus register) must be read (see chapter 6 "Functional sequence", page 23).

Hex code for the data query:

01	03	00 53	00 02	34 1A
Slave	Function	Address of 1st word	Number of words	CRC

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

01	03	04	41 CA 66 66	65 BB
Slave	Function	Bytes read	Temperature value 25.3 °C (float)	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	Wort 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 start the measurement is being sent to a JUMO ecoLine NTU. The slave address of the device is 1, the word address is 0x0001 (chapter 6.2 "Measuring operation on the JUMO ecoLine NTU (type 202670)", page 28) and the value to be written is "3" (start command value to start measuring the temperature and turbidity in the NTU).

Hex code for the instruction:

01	06	00 01	00 03	98 0B
Slave	Function	Word address	Value	CRC

Hex code for the response:

01	06	00 01	00 03	98 0B
Slave	Function	Word address	Value	CRC

## 2 Modbus protocol description

---

### 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 x	Number of bytes 2 x	x word val- ue(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 temperature compensation value 25.3 °C to the Modbus PDU address 0x005D on a JUMO ecoLine O-DO.

⇒ chapter 6.1 "Measuring operation on the JUMO ecoLine O-DO (type 202613)", page 24

Hex code for the instruction:

01	10	00 5D	00 02	04	41 CA 66 66	A8 82
Slave	Function	Address of 1st word	Number of words	Number of bytes	Temperature compensation value 25.3 °C (float)	CRC

Hex code for the response:

01	10	00 5D	00 02	D0 1A
Slave	Function	Address of 1st word	Number of words	CRC

# 2 Modbus protocol description

## 2.5 Data types

Data type	Description	Access	Possible function codes	Number of Modbus registers																																
Floating point	<p>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 transmitted depends on the Modbus implementation of a device. JUMO ecoLine sensors send the bytes in the order specified by the IEEE 754 standard coding.</p> <p>S = Prefix sign bit E = Exponent (two's complement) M = 23-bit normalized mantissa</p> <p>IEEE 754 standard coding</p> <table border="1"> <thead> <tr> <th>Byte 1</th> <th>Byte 2</th> <th>Byte 3</th> <th>Byte 4</th> </tr> </thead> <tbody> <tr> <td>SE7EEEEEE1</td> <td>E0M23MMMMMM16</td> <td>M15MMMMMMM8</td> <td>M7MMMMMMM0</td> </tr> </tbody> </table> <p>Typical modbus coding of floating-point variables <b>not for JUMO ecoLine O-DO/NTU</b></p> <table border="1"> <thead> <tr> <th colspan="2">Address of the 1st Modbus register for variables</th> <th colspan="2">Address of the 2nd Modbus register for variables</th> </tr> <tr> <th>Byte 3</th> <th>Byte 4</th> <th>Byte 1</th> <th>Byte 2</th> </tr> </thead> <tbody> <tr> <td>M15MMMMMMM8</td> <td>M7MMMMMMM0</td> <td>SE7EEEEEE1</td> <td>E0M23MMMMMM16</td> </tr> </tbody> </table> <p>When creating customer-specific applications, the correct byte sequence in storage format needs to be checked. Many compilers use the following storage format:</p> <p>Compiler coding</p> <table border="1"> <thead> <tr> <th>Byte 4</th> <th>Byte 3</th> <th>Byte 2</th> <th>Byte 1</th> </tr> </thead> <tbody> <tr> <td>MMMMMMMM</td> <td>MMMMMMMM</td> <td>EMMMMMMM</td> <td>SEEEEEEE</td> </tr> <tr> <td>Address x</td> <td>Address x+1</td> <td>Address x+2</td> <td>Address x+3</td> </tr> </tbody> </table>	Byte 1	Byte 2	Byte 3	Byte 4	SE7EEEEEE1	E0M23MMMMMM16	M15MMMMMMM8	M7MMMMMMM0	Address of the 1st Modbus register for variables		Address of the 2nd Modbus register for variables		Byte 3	Byte 4	Byte 1	Byte 2	M15MMMMMMM8	M7MMMMMMM0	SE7EEEEEE1	E0M23MMMMMM16	Byte 4	Byte 3	Byte 2	Byte 1	MMMMMMMM	MMMMMMMM	EMMMMMMM	SEEEEEEE	Address x	Address x+1	Address x+2	Address x+3	read only	03	2
		Byte 1	Byte 2	Byte 3	Byte 4																															
SE7EEEEEE1	E0M23MMMMMM16	M15MMMMMMM8	M7MMMMMMM0																																	
Address of the 1st Modbus register for variables		Address of the 2nd Modbus register for variables																																		
Byte 3	Byte 4	Byte 1	Byte 2																																	
M15MMMMMMM8	M7MMMMMMM0	SE7EEEEEE1	E0M23MMMMMM16																																	
Byte 4	Byte 3	Byte 2	Byte 1																																	
MMMMMMMM	MMMMMMMM	EMMMMMMM	SEEEEEEE																																	
Address x	Address x+1	Address x+2	Address x+3																																	
read/write	03, 16																																			
int	<p>Word (16-bit) as unsigned integer value</p> <p>According to the Modbus standard, the higher-value byte (MSB) is send before the lower-value byte (LSB).</p> <p>Value range: 0 to 65.535</p>	read only	03	2																																
		read/write	03, 16																																	

## 2 Modbus protocol description

---

### 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 transmitted over 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	x bytes	2 bytes

#### 2.6.1 Integer values

##### Example

In this example, the status of a measuring operation on a JUMO ecoLine O-DO at address 0x0052 (chapter 6.1 "Measuring operation on the JUMO ecoLine O-DO (type 202613)", page 24) will be read. The value in this example is 7 (word value 0x0007). This value represents the status "Temperature measurement not yet completed".

Data request:

01	03	00 52	00 01	25 DB
Slave	Function	Address of 1st word	Number of words	CRC

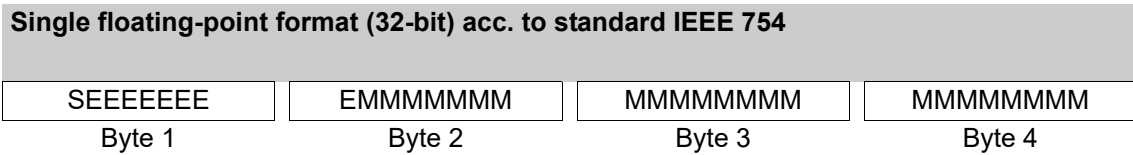
Response (values in Modbus floating-point format):

01	03	02	00 07	F9 86
Slave	Function	Bytes read	Integer value	CRC

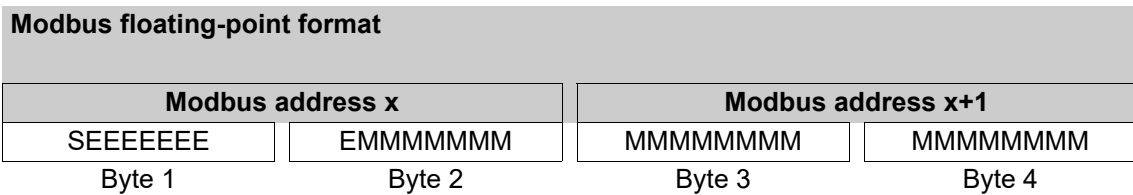
# 2 Modbus protocol description

## 2.6.2 Float values

JUMO ecoLine sensors work with floating-point values in the IEEE-754 standard format (32-bit).



S - Prefix sign bit  
 E - Exponent (two's complement)  
 M - 23-bit normalized mantissa



### Example

In this example, the measured temperature value at address 0x0053 in the device will be read. The value here should be 25.3 °C (0x41CA6666 in IEEE-754 format).

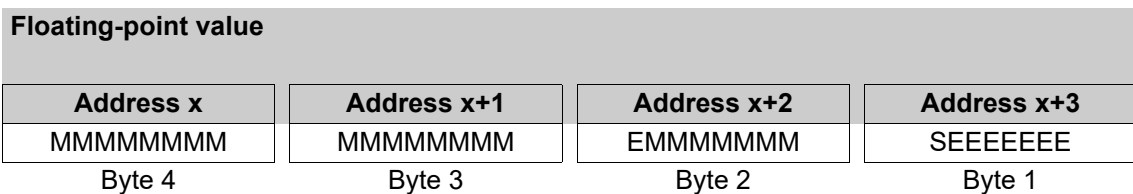
Data request:

01	03	00 53	00 02	34 1A
Slave	Function	Address 1st word	Number of words	CRC

Response (values in Modbus floating-point format):

01	03	04	41 CA	66 66	65 BB
Slave	Function	Bytes read	Floating-point value		CRC

Many compilers (for example, Microsoft Visual C++) store the floating-point values in the following sequence:



### 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 interchanged in the interface program accordingly.

## 2 Modbus protocol description

---

### 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 DB 25 is transmitted and represented in the sequence 25 DB.

#### Example

Requesting the status value of the measurement at address 0x0052:

Instruction: read a word from address 0x0052

01	03	00 52	00 01	25 DB
Slave	Function	Address	Read one word	CRC

Response (CRC16 = 0x86F9)

01	03	02	00 07	F9 86
Slave	Function	Number of bytes	Word 1	CRC

Word 1 = 7 represents the status "Temperature measurement not yet completed".

# 2 Modbus protocol description

## 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 the JUMO ecoLine O-DO/NTU are listed in chapter 2.4 "Function codes", page 8.
- 02 = Invalid address or too many words or bits should be read or written
- 03 = The format of the data cannot be read.
- 255 = A communication problem exists.

Modbus error responses can be recognized by the fixed length of 5 bytes. All other Modbus telegrams are longer.

#### 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 Modbus protocol description

---

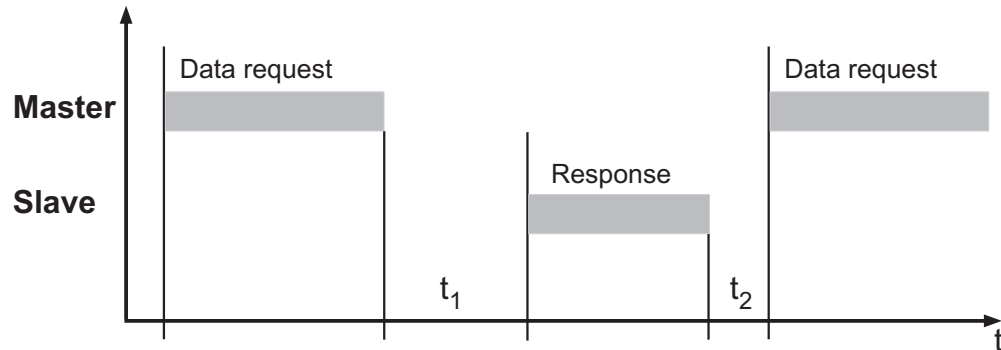


# 3 Modbus over a serial interface

## 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 <sub>1</sub>	Internal waiting period of the device before checking the data request and the internal processing time:  min.: 3.5 byte time + longest possible sensor response time waiting period required for JUMO ecoLine O-DO/NTU sensors: 300 ms
t <sub>2</sub>	The master has to observe this waiting period before starting a new data request: 3.5 characters



#### NOTE!

During t<sub>1</sub> and t<sub>2</sub> and during the response time of the slave, no data queries may be generated by the master. Queries during t<sub>1</sub> and t<sub>2</sub> 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.

A data format of 8 data bits, no parity bit, and one stop bit results in the following:

$$\text{Character transmission time [ms]} = 1000 \times 10 \text{ bits} \div \text{baud rate}$$

For other data formats, the following is the result:

$$\text{Character transmission time [ms]} = 1000 \times 11 \text{ bits} \div \text{baud rate}$$

### Example

Identifier for end of data request or end of response

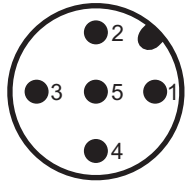
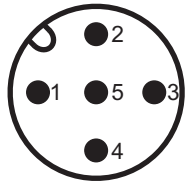
Waiting period = 3.5 characters \* 1000 \* 11 bits ÷ baud rate

Baud rate [Baud]	Data format [Bit]	Character transmission time[ms]
9600	11	1.146
	10	1.042

## 3 Modbus over a serial interface

---

## 4.1 Interface assignment on a JUMO ecoLine O-DO/NTU

Variant with 5-pole M12 plug connector, A-coded		
Pin	Potential	Symbol
1	+5 V	<p>Connector</p> 
2	not connected	
3	GND	
4	RS 485 B (RxD/TxD-)	
5	RS 485 A (RxD/TxD+)	
<p>The connection to the serial interface of a master with screw or spring-cage terminals is made with the aid of the JUMO M12 digiLine master connection cable.</p>		<p>Socket</p> 

Connection cable variant with ferrules	
Wire color	Potential
Red	+5 V
Black	GND
Green	RS 485 B (RxD/TxD-)
White	RS 485 A (RxD/TxD+)

## 4 Interfaces

---

## Important information



### CAUTION!

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

This can disrupt processes 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 technical personnel.



### NOTE!

Changes to the configuration settings described in this chapter for JUMO ecoLine sensors 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 ecoLine sensors.

Configuration item	Selection/settings	Description
Baud rate	9600	Transmission speed (symbol rate) of the serial interface
Data format	8 - 1 - no parity	Format of the data word Information bit - stop bit - parity
Device address	1 to 247	A bus user's unique ID  0 = Broadcast address <sup>a</sup> 1 to 247 = Unicast addresses <sup>b</sup>

<sup>a</sup> 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.

<sup>b</sup> Unicast addresses are intended for use as slave addresses. They are used to specify the slave devices uniquely so that the master can communicate with them explicitly.

## 5 Configuring interfaces

---

## 6 Functional sequence

---

Extracting measurement data from JUMO ecoLine sensors requires implementation of communication routines in the Modbus master. The communication routines for measuring are described in below.



**NOTE!**

Changes to the configuration settings described in this chapter for JUMO ecoLine sensors 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.

---

# 6 Functional sequence

## 6.1 Measuring operation on the JUMO ecoLine O-DO (type 202613)

The measuring operation on a JUMO ecoLine O-DO has 4 phases:

- Starting the measurement
- Waiting for the measurement to be completed
- Checking the status of the measurement and possibly extending the waiting period
- Reading the measured value

It is necessary to ensure prior to the actual measuring operation that correct compensation values are saved in the sensor. Compensation values are not saved permanently in a JUMO ecoLine sensor and should be transmitted by the master prior to measurement. The following table describes the entire measuring operation with compensation in detail.

Step	Action of the Modbus master / Activity
1	<p>The master device must transmit the required, current compensation values (see address tables that follow) to the JUMO ecoLine O-DO if they differ from the default values.</p> <p>The temperature compensation for the JUMO ecoLine O-DO must be set correctly when configuring the sensor on the PC using the JUMO DSM software. The source for the compensation temperature can be selected as either the internal temperature sensor or the compensation values transmitted via Modbus.</p> <p>Under certain circumstances, step 1 can be skipped, because either the default values for the JUMO ecoLine O-DO for temperature, air pressure, and salt content (25 °C, 1023 hPa, 0 g/kg) match the process conditions or the internal temperature sensor is being set as source for the compensation temperature and the default values for the JUMO ecoLine O-DO for air pressure and salt content (1023 hPa, 0 g/kg) match the process conditions.</p>
2	<p>The master device must send the command to start the measuring operation to the Modbus PDU address 0x0001 on the JUMO ecoLine O-DO. There are 3 different start commands. Each of the commands starts measurement of the temperature and the oxygen value in a unit specific to the respective start command (see address tables that follow).</p> <p>The sampling rate should not be set higher than necessary. The membrane of the JUMO ecoLine O-DO has an operating life of about 1 million measurements.</p>
3	<p>The master device must now wait for 300 ms. In order not to disrupt the measurement routine, no Modbus commands can be sent to the JUMO ecoLine O-DO during this time.</p>
4	<p>The master device must request the status of the measurements. A corresponding status word that must be requested and evaluated by the master device exists at the Modbus PDU address 0x0052 on the JUMO ecoLine O-DO. The status word consists of bit fields that represent the status of the individual measurements of the JUMO ecoLine O-DO (see address table that follows). If a bit field contains the value 7, the particular measurement is not yet complete and the waiting period must be extended. After the waiting period has expired, the status must be checked again. If necessary, this procedure must be continued (polling) until all measurements have been completed (status values 0 to 3). In the case of failed measurements (values 4 to 6), the current measurement must be canceled.</p>
5	<p>Once the measurement is complete, the master device can retrieve the measurement data for the measurements started in step 2. Only the requested measurements contain currently valid measured values. Measured values that were not requested must be ignored. The Modbus PDU addresses of the measurement data are listed in the address tables that follow.</p>



## 6 Functional sequence

---

### Compensation values for JUMO ecoLine O-DO

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Unit
Dec.	Hex.					Default value
93	005D	float	2	r/w	Temperature	°C 25 °C
95	005F	float	2	r/w	Air pressure	hPa 1023 hPa
97	0061	float	2	r/w	Salt content	g/kg 0 g/kg

## 6 Functional sequence

### Control data for measuring operation of JUMO ecoLine O-DO

Modbus PDU address		Data type	Number of Modbus registers	Access	Data
Dec.	Hex.				
1	0001	int	1	w	<p>Start measuring operation</p> <p><b>Coding for "Start measuring operation":</b> Possible start commands and their meaning: 3 = Start measuring temperature and oxygen in %Sat 7 = Start measuring temperature and oxygen in %Sat and mg/l 11 = Start measuring temperature and oxygen in %Sat and ppm</p>
82	0052	int	1	r	<p>Measuring operation status</p> <p><b>Coding for the "Measuring operation" status control word</b> Bit field assignment: Bits 0 to 2: Status of the temperature measurement Bits 3 to 5: Status of the oxygen measurement in %Sat Bits 6 to 8: Status of the oxygen measurement in mg/l Bits 9 to 11: Status of the oxygen measurement in ppm</p> <p>Status values: 0 = Measurement performed without error 1 = Measurement OK, but outside the specifications of the sensor 2 = Measurement OK, but disrupted by Modbus communication and thus reduced accuracy 3 = Measurement OK (message reserved for future extensions) 4 = Measurement NOK, outside the specifications of the sensor 5 = Measurement NOK, membrane cap missing from sensor or is damaged 6 = Measurement NOK (message reserved for future extensions) 7 = Measurement not yet complete</p>

## 6 Functional sequence

---

### Measured values from JUMO ecoLine O-DO

Modbus PDU address		Data type	No. of Modbus registers	Access	Data	Unit
Dec.	Hex.					
83	0053	float	2	r	Temperature	°C
85	0055	float	2	r	Oxygen saturation	%Sat
87	0057	float	2	r	Oxygen concentration	mg/l
89	0059	float	2	r	Oxygen concentration	ppm

## 6 Functional sequence

---

### 6.2 Measuring operation on the JUMO ecoLine NTU (type 202670)

The measuring operation on a JUMO ecoLine NTU has 4 phases:

- Starting the measurement
- Waiting for the measurement to be completed
- Checking the status of the measurement and possibly extending the waiting period
- Reading the measured value

The following table describes the entire measuring operation in detail.

Step	Action of the Modbus master / Activity
1	The master device must send the command to start the measuring operation (value 31) to the Modbus PDU address 0x0001 on the JUMO ecoLine NTU.
2	The master device must now wait for 300 ms. In order not to disrupt the measurement routine, no Modbus commands can be sent to the JUMO ecoLine NTU during this time.
3	The master device must request the status of the measurements. A corresponding status word that must be requested and evaluated by the master device exists at the Modbus PDU address 0x0052 on the JUMO ecoLine NTU. The status word consists of bit fields that represent the status of the individual measurements of the JUMO ecoLine NTU (see address table that follows). If a bit field contains the value 7, the particular measurement is not yet complete and the waiting period must be extended. After the waiting period has expired, the status must be checked again. If necessary, this procedure must be continued (polling) until all measurements have been completed (status values 0 to 3). In the case of failed measurements (values 4 to 6), the current measurement must be canceled.
4	Once the measurement is complete, the master device can retrieve the measurement data. The Modbus PDU addresses of the measurement data are listed in the address tables that follow.

## 6 Functional sequence

### Control data for measuring operation of JUMO ecoLine NTU

Modbus PDU address		Data type	Number of Modbus registers	Access	Data
Dec.	Hex.				
1	0001	int	1	w	<p>Start measuring operation</p> <p><b>Coding for "Start measuring operation":</b>            Value 1 = Start measuring the temperature            Value 3 = Start measuring the temperature and turbidity in <b>NTU</b>            Value 5 = Start measuring the temperature and turbidity in <b>FNU</b></p>
82	0052	int	1	r	<p>Status of measuring operation</p> <p><b>Coding for the "Measuring operation" status control word</b></p> <p>Bit field assignment:            Bits 0 to 2: Status of the temperature measurement            Bits 3 to 5: Status of the turbidity measurement on NTU            Bits 6 to 8: Status of turbidity measurement on FTU</p> <p>Status values:            0 = Measurement performed without error            1 = Measurement OK, but outside the specifications of the sensor            2 = Measurement OK, but reduced accuracy due to excessive extraneous light            3 = Measurement OK (message reserved for future extensions)            4 = Measurement NOK, outside the specifications of the sensor            5 = Measurement NOK due to excessive extraneous light            6 = Measurement NOK (message reserved for future extensions)            7 = Measurement not yet complete</p>

## 6 Functional sequence

---

### Measured values from JUMO ecoLine NTU

Modbus PDU address		Data type	Number of Modbus registers	Access	Data	Unit
Dec.	Hex.					
83	0053	float	2	r	Temperature	°C
85	0055	float	2	r	Turbidity NTU	NTU
87	0057	float	2	r	Turbidity FTU	FNU





**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](mailto:mail@jumo.net)  
Internet: [www.jumo.net](http://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](mailto:sales@jumo.co.uk)  
Internet: [www.jumo.co.uk](http://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](mailto:info.us@jumo.net)  
Internet: [www.jumousa.com](http://www.jumousa.com)

