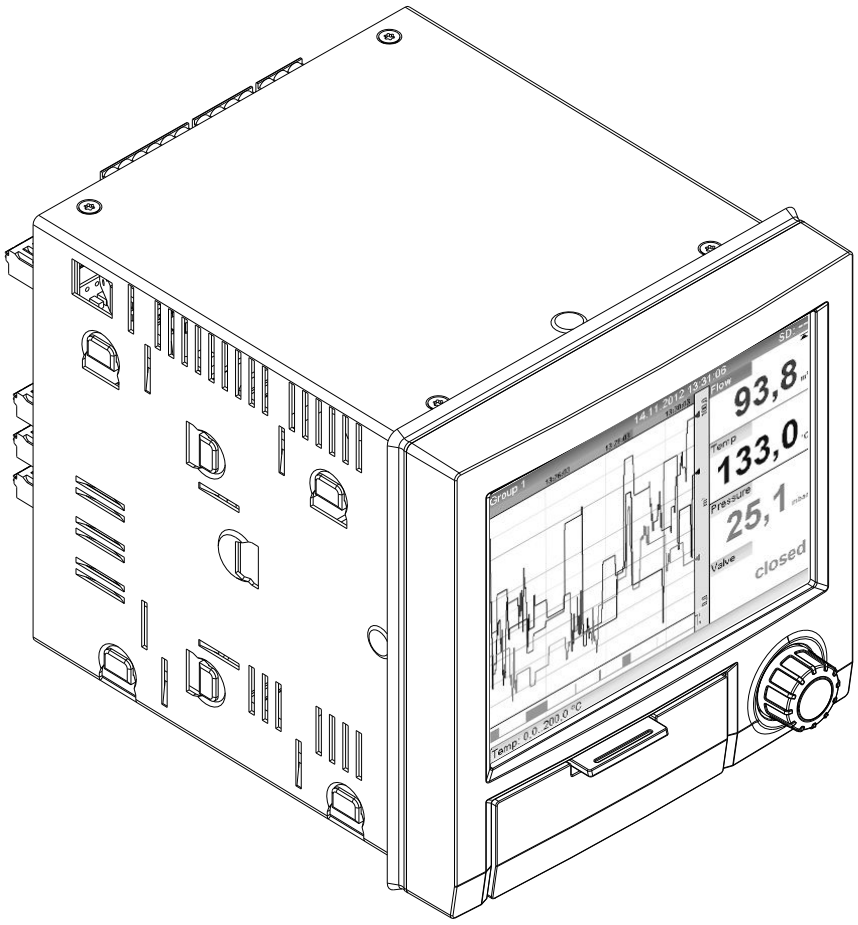
BA01259O/09/EN/03.24-00
71644471
2024-01-31

Valid as of version
ENU000A, V2.04.xx

# Operating Instructions
# ORSG35

## Universal Data Manager
## Additional instructions Modbus RTU/TCP slave



# Table of contents

# 1 General information

## 1.1 Safety symbols

**⚠ DANGER**
This symbol alerts you to a dangerous situation. Failure to avoid this situation will result in serious or fatal injury.

**⚠ WARNING**
This symbol alerts you to a dangerous situation. Failure to avoid this situation can result in serious or fatal injury.

**⚠ CAUTION**
This symbol alerts you to a dangerous situation. Failure to avoid this situation can result in minor or medium injury.

**NOTICE**
This symbol contains information on procedures and other facts which do not result in personal injury.

## 1.2 Scope of delivery

**NOTICE**

**This manual contains an additional description for a special software option.**
These additional instructions are not a substitute for the Operating Instructions pertaining to the device!

- ▶ Detailed information can be found in the Operating Instructions and the additional documentation.

## 1.3 Requirements

The "Modbus slave" option must be enabled in the device. For optional retrofitting, please refer to the Operating Instructions.

Modbus RTU via RS485 is only possible if the optional RS232/RS485 interface (on the back of the device) is present in the device, whereby only RS485 is supported. Modbus TCP is possible via the integrated Ethernet interface (on the back of the device).

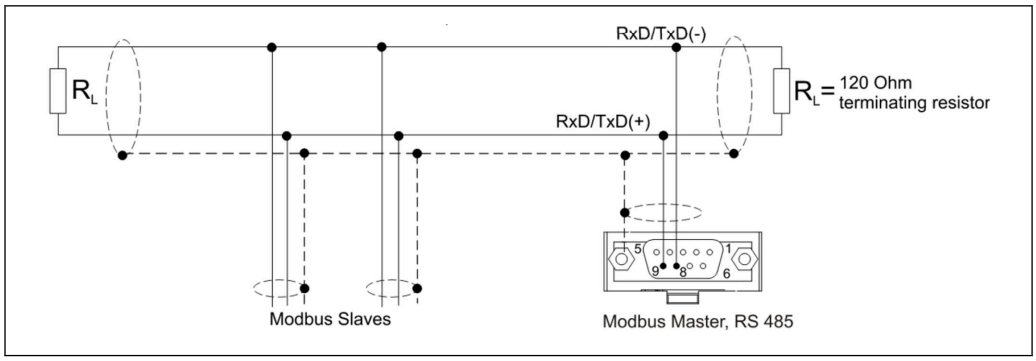
## 1.4 Firmware history

Overview of the device software history:

| Device software Version/date | Software changes | Analysis software version | Version of OPC server | Operating Instructions |
|---|---|---|---|---|
| V02.00.00 / 01.2013 | Original software | V1.3.0 and higher | V5.00.03 and higher | BA01259O/09/EN /01.13 |
| V02.00.xx / 02.2015 | Bug fixes | V1.3.0 and higher | V5.00.03 and higher | BA01259O/09/EN /02.15 |
| V2.04.06 / 10.2022 | Bug fixes | V1.6.3 and higher | V5.00.07 and higher | BA01259O/09/EN /01.24-00 |

## 1.5 Connection of Modbus RTU

The pin assignment does not comply with the standard (Modbus over serial line specification and implementation guide V1.02).




*Pin assignment of the Modbus RTU connector*

| Pin | Direction | Signal | Description |
|---|---|---|---|
| Housing | - | Functional ground | Protective ground |
| 1 | - | GND | Ground (isolated) |
| 9 | Input | RxD/TxD(+) | RS-485 B line |
| 8 | Output | RxD/TxD(-) | RS-485 A line |

## 1.6 Modbus TCP connection

The Modbus TCP interface is physically identical to the Ethernet interface.

### 1.6.1 Transfer LED

*Functional description of the status LED for Modbus TCP*

| Status LED | Indicator for |
|---|---|
| Off | No communication |
| Flashes green | Communication |

### 1.6.2 Link LED

*Functional description of the link LED for Modbus TCP*

| Status LED | Indicator for |
|---|---|
| Off | No connection |
| Flashes yellow | Activity |

## 1.7 Functional description

The Modbus RTU option enables the device to be connected to Modbus via RS485 with the functionality of a Modbus RTU slave.

**Supported baud rates:** 9600, 19200, 38400, 57600, 115200

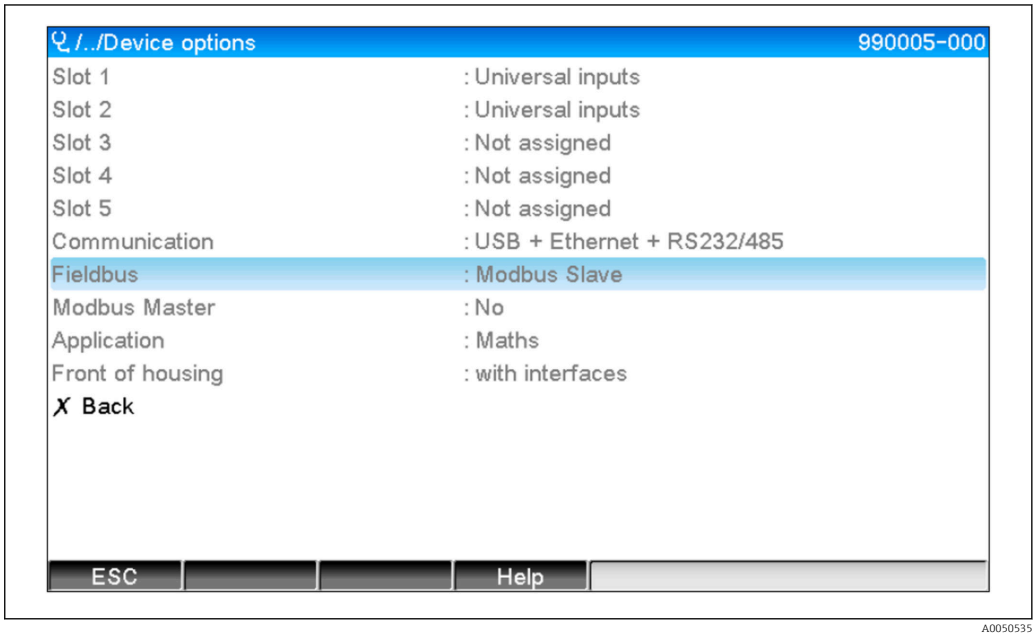
**Parity:** None, Even, Odd

The Modbus TCP option enables the device to be connected to Modbus TCP with the functionality of a Modbus TCP slave. The Ethernet connection supports 10/100 Mbit, full or half duplex.

You can choose between Modbus TCP or Modbus RTU in the settings. It is not possible to select both at the same time.

## 1.8 Checking the availability of the Modbus slave functionality

In the main menu under → **Diagnostics** → **Device information** → **Device options** or → **Setup** → **Advanced setup** → **System** → **Device options**, you can check under **Fieldbus** whether the **Modbus Slave** option is enabled. The hardware interface via which communication is possible can be determined under **Communication**:

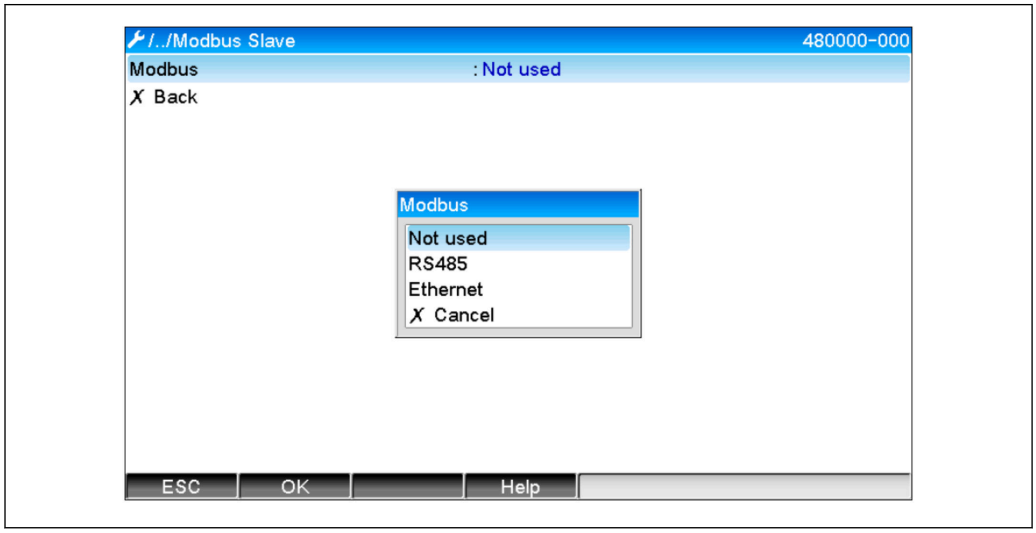



1 *Checking the availability of the Modbus slave functionality*

# 2 Settings under Setup

## 2.1 Modbus TCP, RS485

Under → **Setup** → **Advanced setup** → **Communication** → **Modbus Slave**, you can select which interface is to be used for Modbus:

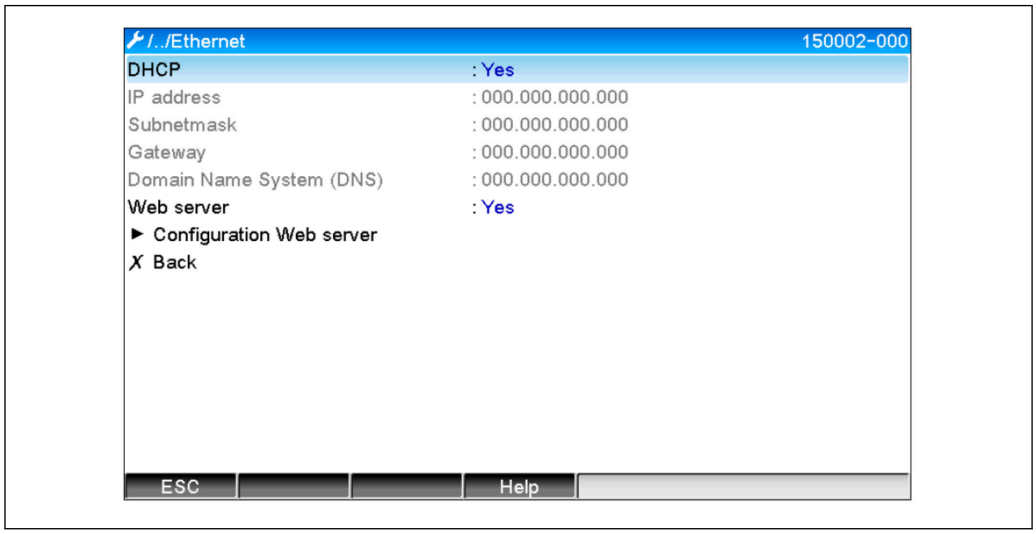



2 *Selecting the interface for Modbus*

If Modbus RTU (RS485) has been selected, the following parameters can be set:
- Device address (1 to 247)
- Baud rate (9600, 19200, 38400, 57600, 115200)
- Parity (None, Even, Odd)

If Modbus TCP (Ethernet) has been selected, the following parameter can be set:
Port TCP port (standard: 502)

When using Modbus TCP, the settings for the Ethernet interface can be made under → **Setup** → **Advanced setup** → **Communication** → **Ethernet**:




3 *Ethernet interface settings*

In addition, a time period can be set under → **Expert → Communication → Modbus Slave → Timeout**, after which the relevant channel is set to "Invalid".

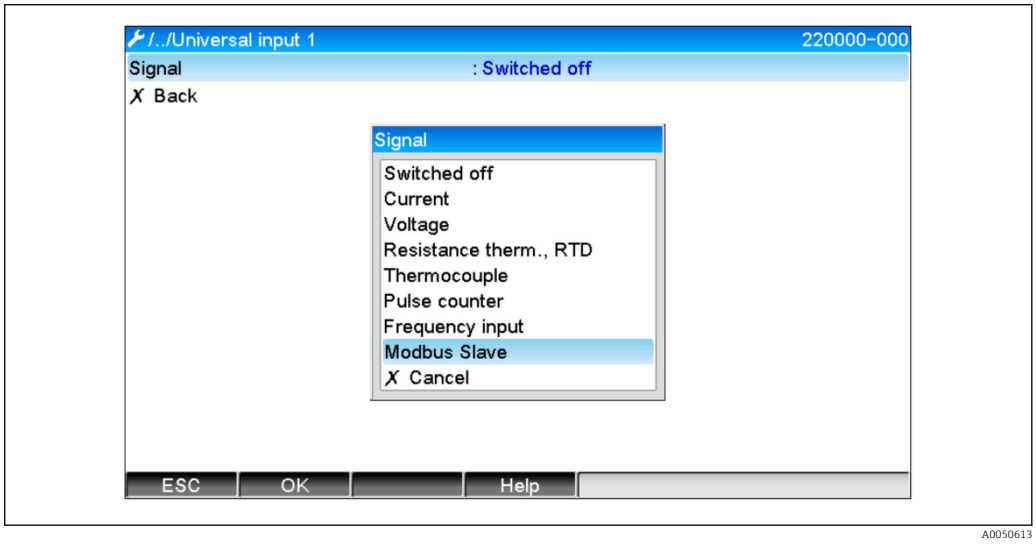
The timeout only applies to channels that receive a value from the Modbus master. Channels that are only read by the Modbus master are not affected.

## 2.2 Universal channels

All universal inputs (12) are enabled and can be used as Modbus inputs, even if they are not actually available as plug-in cards.

### 2.2.1 Data transfer: Modbus master -> device:

Under → **Setup → Advanced setup → Inputs → Universal inputs → Universal input X**, the **Signal** parameter is set to **Modbus Slave**:




4 *Setting the universal input to Modbus*

With this setting, the universal input can be written by a Modbus master as described in → 9.

### 2.2.2 Data transfer: Device → Modbus master:

The universal inputs 1 to 12 can be read by the Modbus master as described in → 12.

## 2.3 Maths channels

### 2.3.1 Data transfer: Device → Modbus master:

Optional math channels are available under → **Setup → Advanced setup → Application → Math**.

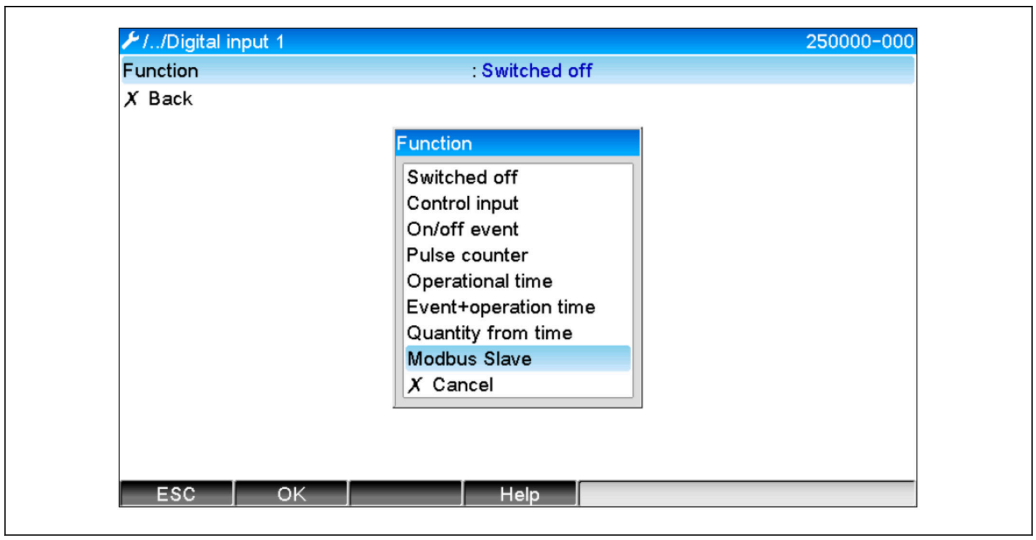
The results can be read by the Modbus master (see → 14 and → 16).

## 2.4 Digital channels

All digital inputs (6) are enabled and can be used as Modbus inputs.

### 2.4.1 Data transfer: Modbus master → device:

Under → **Setup** → **Advanced setup** → **Inputs** → **Digital inputs** → **Digital input X**, the **Function** parameter is set to **Modbus Slave**:




*5 Setting the digital channel to Modbus*

With this setting, the digital channel can be written by the Modbus master as described in → 11.

The digital status transmitted by the Modbus master has the same functionality in the device as the status of a real digital channel.

### 2.4.2 Data transfer: Device → Modbus master:

**Control input or on/off message**

The Modbus master can read out the digital status of the digital channel set in this way (see → 16).

**Pulse counter or operating time**

The Modbus master can read out the totalizer or the total operating time of the digital channel set in this way (see → 17).

**Message + operating time**

The Modbus master can read out the digital status and the totalizer of the digital channel set in this way (see → 16and → 17).

## 2.5 General information

The **03: Read Holding Register** and **16: Write Multiple Registers** functions are supported.

The following parameters can be transmitted from the **Modbus master to the device**:
- Analog values (instantaneous values)
- Digital statuses

The following parameters can be transmitted from the **device to the Modbus master**:
- Analog values (instantaneous values)
- Integrated analog values (totalizer)
- Math channels (result: Status, instantaneous value, operating time, totalizer)
- Integrated math channels (totalizer)
- Digital statuses

- Pulse counter (totalizer)
- Operating times
- Relay statuses

## 2.6 Addressing

The request/response examples refer to Modbus RTU via RS485.

The register addresses are all based on 0.

### 2.6.1 Modbus master → device: Instantaneous value of universal channels

The values of the universal channels 1–12 must be written via **16 Write Multiple Registers**. It is possible to transfer the value as a 32-bit float or 64-bit float.

*Register addresses of the universal inputs*

| Channel | | Reg. dec. | Reg. hex. | Length, bytes | | Reg. dec. | Reg. hex. | Length, bytes |
|---|---|---|---|---|---|---|---|---|
| Universal 1 | | 200 | 0C8 | 6 | | 5200 | 1450 | 10 |
| Universal 2 | | 203 | 0CB | 6 | | 5205 | 1455 | 10 |
| Universal 3 | | 206 | 0CE | 6 | | 5210 | 145A | 10 |
| Universal 4 | | 209 | 0D1 | 6 | | 5215 | 145F | 10 |
| Universal 5 | | 212 | 0D4 | 6 | | 5220 | 1464 | 10 |
| Universal 6 | | 215 | 0D7 | 6 | | 5225 | 1469 | 10 |
| Universal 7 | | 218 | 0DA | 6 | | 5230 | 146E | 10 |
| Universal 8 | | 221 | 0DD | 6 | | 5235 | 1473 | 10 |
| Universal 9 | | 224 | 0E0 | 6 | | 5240 | 1478 | 10 |
| Universal 10 | | 227 | 0E3 | 6 | | 5245 | 147D | 10 |
| Universal 11 | | 230 | 0E6 | 6 | | 5250 | 1482 | 10 |
| Universal 12 | | 233 | 0E9 | 6 | | 5255 | 1487 | 10 |

The first register contains the status of the floating point number (32-bit float) transmitted in the second and third registers (see → 25).

**Example: Writing to universal channel 6 with the value 123.456 (32-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| | **00** | **80** | **42** | **F6** | **E9** | **79** |
| | | Floating point number status | Floating point number = 123.456 (32-bit float) | | | |

| Register | Value (hex) |
|---|---|
| 215 | **0080** |
| 216 | **42F6** |
| 217 | **E979** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 10 | 16: Write Multiple Registers |

| | | | |
|---|---|---|---|
| | Register | 00 D7 | Register 215 |
| | No. of registers | 00 03 | 3 registers |
| | No. of bytes | 06 | |
| | Status | 00 80 | |
| | FLP | 42 F6 E9 79 | 123.456 |
| | CRC | 28 15 | |
| **Response:** | Slave address | 01 | |
| | Function | 10 | 16: Write Multiple Registers |
| | Register | 00 D7 | Register 271 |
| | No. of registers | 00 03 | |
| | CRC | 30 30 | |

The first register contains the status (see → 📄 25) of the floating point number (64-bit float) transmitted in the second to fifth registers.

**Example: Writing to universal channel 6 with the value 123.456 (64-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| | **00** | **80** | **40** | **5E** | **DD** | **2F** | **1A** | **9F** | **BE** | **77** |
| | | Floating point number status | Floating point number = 123.456 (64-bit float) | | | | | | | |

| Register | Value (hex) |
|---|---|
| 5225 | **0080** |
| 5226 | **405E** |
| 5227 | **DD2F** |
| 5228 | **1A9F** |
| 5229 | **BE77** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 10 | 16: Write Multiple Registers |
| | Register | 14 69 | Register 5225 |
| | No. of registers | 00 05 | 5 registers |
| | No. of bytes | 0A | |
| | Status | 00 80 | |
| | FLP | 40 5E DD 2F 1A 9F BE 77 | 123.456 |
| | CRC | 67 56 | |
| **Response:** | Slave address | 01 | |
| | Function | 10 | 16: Write Multiple Registers |
| | Register | 14 69 | Register 5225 |
| | No. of registers | 00 05 | |
| | CRC | D5 E6 | |

## 2.6.2 Modbus master → device: Digital input status

### Writing all statuses simultaneously

The statuses of the digital inputs 1–6 must be written via **16 Write Multiple Registers**.

*Register addresses of the digital inputs (Modbus master → device)*

| Channel | Reg. dec. | Reg. hex. | Length, bytes |
|---|---|---|---|
| Digital 1–6 | 1240 | 4D8 | 2 |

### Example: Setting digital input 4 to high (all others to low), slave address 1

| Byte 0<br>Status (bit 15–8) | Byte 1<br>Status (bit 7–0) |
|---|---|
| 00000000 | 0000**1**000 |
| Always 0 | Bit 3 high<br>Digital 4 |

| Register | Value (hex) |
|---|---|
| 1240 | **0008** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 10 | 16: Write Multiple Registers |
| | Register | 04 D8 | Register 1240 |
| | No. of registers | 00 01 | 1 register |
| | No. of bytes | 02 | |
| | Digital status | 00 08 | Digital 4 to high |
| | CRC | F0 8E | |
| **Response:** | Slave address | 01 | |
| | Function | 10 | 16: Write Multiple Registers |
| | Register | 04 D8 | Register 1240 |
| | No. of registers | 00 01 | |
| | CRC | 80 C2 | |

### Write statuses individually

The statuses of digital inputs 1-6 must be written via **16 Write Multiple Registers**.

*Register addresses of the digital inputs (Modbus master → device)*

| Channel | Reg. dec. | Reg. hex. | Length, bytes |
|---|---|---|---|
| Digital 1 | 1200 | 4B0 | 2 |
| Digital 2 | 1201 | 4B1 | 2 |
| Digital 3 | 1202 | 4B2 | 2 |
| Digital 4 | 1203 | 4B3 | 2 |
| Digital 5 | 1204 | 4B4 | 2 |
| Digital 6 | 1205 | 4B5 | 2 |

**Example: Setting digital input 4 to high, slave address 1**

| Byte 0<br>Status (bit 15–8) | Byte 1<br>Status (bit 7–0) |
|---|---|
| 00000000 | 0000**1**000 |
| Always 0 | Bit 3 high Digital 4 |

| **Register** | **Value (hex)** |
|---|---|
| 1203 | **0001** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 10 | 16: Write Multiple Registers |
| | Register | 04 B3 | Register 1203 |
| | No. of registers | 00 01 | 1 register |
| | No. of bytes | 02 | |
| | Digital status | 00 01 | Digital 4 to high |
| | CRC | 38 53 | |
| **Response:** | Slave address | 01 | |
| | Function | 10 | 16: Write Multiple Registers |
| | Register | 04 B3 | Register 1203 |
| | No. of registers | 00 01 | |
| | CRC | F1 1E | |

### 2.6.3 Device → Modbus master: Universal channels (instantaneous value)

The universal inputs 1–12 are read out via **03 Read Holding Register (4x)**.

It is possible to transfer the value as a 32-bit float or 64-bit float.

*Register addresses of the universal inputs (device → Modbus master)*

| **Channel** | | **Reg. dec.** | **Reg. hex.** | **Length, bytes** | | **Reg. dec.** | **Reg. hex.** | **Length, bytes** |
|---|---|---|---|---|---|---|---|---|
| Universal 1 | | 200 | 0C8 | 6 | | 5200 | 1450 | 10 |
| Universal 2 | | 203 | 0CB | 6 | | 5205 | 1455 | 10 |
| Universal 3 | | 206 | 0CE | 6 | | 5210 | 145A | 10 |
| Universal 4 | | 209 | 0D1 | 6 | | 5215 | 145F | 10 |
| Universal 5 | | 212 | 0D4 | 6 | | 5220 | 1464 | 10 |
| Universal 6 | | 215 | 0D7 | 6 | | 5225 | 1469 | 10 |
| Universal 7 | | 218 | 0DA | 6 | | 5230 | 146E | 10 |
| Universal 8 | | 221 | 0DD | 6 | | 5235 | 1473 | 10 |
| Universal 9 | | 224 | 0E0 | 6 | | 5240 | 1478 | 10 |
| Universal 10 | | 227 | 0E3 | 6 | | 5245 | 147D | 10 |
| Universal 11 | | 230 | 0E6 | 6 | | 5250 | 1482 | 10 |
| Universal 12 | | 233 | 0E9 | 6 | | 5255 | 1487 | 10 |

The first register contains the status (see → 25) and the limit value violations (see → 24) of the floating point number transmitted in the second and third registers (32-bit float).

**Example: Reading analog 1 with the value 82.47239685 (32-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| | **00** | **80** | **42** | **A4** | **F1** | **DE** |
| | Limit value violation | Floating point number status | Floating point number = 82.47239685 | | | |

| Register | Value (hex) |
|---|---|
| 200 | **0080** |
| 201 | **42A4** |
| 202 | **F1DE** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 00 C8 | Register 200 |
| | No. of registers | 00 03 | 3 registers |
| | CRC | 84 35 | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | No. of bytes | 06 | 6 bytes |
| | Status | 00 80 | |
| | FLP | 42 A4 F1 DE | 82.47239685 |
| | CRC | B0 F8 | |

The first register contains the status (see → 25) and the limit value violations (see → 24) of the floating point number transmitted in the second to fifth registers (64-bit float).

**Example: Reading universal channel 1 with the value 82.4723968506 (64-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| | **00** | **80** | **40** | **54** | **9E** | **3B** | **C0** | **00** | **00** | **00** |
| | Limit value violations | Floating point number status | Floating point number = 82.4723968506 (64-bit float) | | | | | | | |

| Register | Value (hex) |
|---|---|
| 5200 | **0080** |
| 5201 | **4054** |
| 5202 | **9E3B** |
| 5203 | **C000** |
| 5204 | **0000** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 14 50 | Register 5200 |
| | No. of registers | 00 05 | 5 registers |
| | CRC | 80 28 | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | No. of bytes | 0A | 10 bytes |
| | Status | 00 80 | |
| | FLP | 40 54 9E 3B C0 00 00 00 | 82.4723968506 |
| | CRC | 91 3E290 | |

### 2.6.4 Device → Modbus master: Math channels (result)

The results of the math channels 1–4 are read out via **03 Read Holding Register (4x)**. It is possible to transfer the value as a 32-bit float or 64-bit float.

*Register addresses of the math channels (device → Modbus master)*

| **Channel** | | **Reg. dec.** | **Reg. hex.** | **Length, bytes** | | **Reg. dec.** | **Reg. hex.** | **Length, bytes** |
|---|---|---|---|---|---|---|---|---|
| Math 1 | | 1500 | 5DC | 6 | | 6500 | 1964 | 10 |
| Math 2 | | 1503 | 5DF | 6 | | 6505 | 1969 | 10 |
| Math 3 | | 1506 | 5E2 | 6 | | 6510 | 196E | 10 |
| Math 4 | | 1509 | 5E5 | 6 | | 6515 | 1973 | 10 |

The first register contains the status (see → 📄 25) and the limit value violations (see → 📄 24) of the floating point number transmitted in the second and third registers (32-bit float).

**Example: Reading math 1 (result of instantaneous value) (32-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| | **00** | **80** | **46** | **40** | **E6** | **B7** |
| | Limit value violations | Floating point number status | Floating point number = 12345.67871 | | | |

| **Register** | **Value (hex)** |
|---|---|
| 1500 | **0080** |
| 1501 | **4640** |
| 1502 | **E6B7** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 05 DC | Register 1500 |
| | No. of registers | 00 03 | 3 registers |
| | CRC | C4 FD | |

| | | | |
|---|---|---|---|
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | No. of bytes | 06 | 6 bytes |
| | Status | 00 80 | |
| | FLP | 46 40 E6 B7 | 12345.67871 |
| | CRC | 3E 21 | |

The first register contains the status (see → 25) and the limit value violations (see → 24) of the floating point number transmitted in the second to fifth registers (64-bit float).

**Example: Reading math 1 (result of instantaneous value) (64-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| | **00** | **80** | **40** | **C8** | **1C** | **D6** | **E6** | **31** | **F8** | **A1** |
| | Limit value violations | Floating point number status | Floating point number = 12345.6789 (64-bit float) | | | | | | | |

| **Register** | **Value (hex)** |
|---|---|
| 6500 | **0080** |
| 6501 | **40C8** |
| 6502 | **1CD6** |
| 6503 | **E631** |
| 6504 | **F8A1** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 19 64 | Register 6500 |
| | No. of registers | 00 05 | 5 registers |
| | CRC | C3 4A | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | No. of bytes | 0A | 10 bytes |
| | Status | 00 80 | |
| | FLP | 40 C8 1C D6 E6 31 F8 A1 | 12345.6789 |
| | CRC | A7 FD | |

**Example: Reading math 1–4 (result of status), slave address 1**

The statuses of math channels 1–4 are read out via **03 Read Holding Register (4x)**.

*Register address of the statuses of the math channels (device → Modbus master)*

| **Channel** | **Reg. dec.** | **Reg. hex.** | **Length, bytes** |
|---|---|---|---|
| Math 1–4 | 1800 | 708 | 2 |

| Byte 0 | Byte 1<br>Status (bit 5–0) |
|---|---|
| 00000000 | 00000**11** |
| Always 0 | Bit 0 and 1 high<br>Math 1 and 2 |

| **Register** | **Value (hex)** |
|---|---|
| 1800 | **0003** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 07 08 | Register 1800 |
| | No. of registers | 00 01 | 1 register |
| | CRC | 04 BC | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 16: Write Multiple Registers |
| | Number | 02 | 2 bytes |
| | Statuses | 00 03 | Math 1 and 2 status high |
| | CRC | F8 45 | |

## 2.6.5 Device → Modbus master: Digital channels (status)

### Reading out all statuses simultaneously

The statuses of digital inputs 1–6 are read out via **03 Read Holding Register (4x)**.

*Register addresses of all digital inputs (device → Modbus master)*

| **Channel** | **Reg. dec.** | **Reg. hex.** | **Length, bytes** |
|---|---|---|---|
| Digital 1–6 | 1240 | 4D8 | 2 |

### Example: Reading the statuses of the digital inputs 1–6, slave address 1

| Byte 0<br>Status (bit 15–8) | Byte 1<br>Status (bit 7–0) |
|---|---|
| 00000000 | 00**1**00**1**00 |
| Always 0 | Bit 2 and 5 high<br>Digital 3 and 6 |

| **Register** | **Value (hex)** |
|---|---|
| 1240 | **0024** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 04 D8 | Register 1240 |
| | No. of registers | 00 01 | 1 register |
| | CRC | 05 01 | |

| | | | |
|---|---|---|---|
| **Response:** | Slave address | 01 | |
| | Function | 03 | 16: Write Multiple Registers |
| | Number | 02 | 2 bytes |
| | Statuses | 00 24 | Digital 3 and 6 high |
| | CRC | B8 5F | |

### Reading out statuses individually

The statuses of the digital inputs 1–6 are read out via **03 Read Holding Register (4x)**.

*Register addresses of the digital inputs (device → Modbus master)*

| Channel | Reg. dec. | Reg. hex. | Length, bytes |
|---|---|---|---|
| Digital 1 | 1200 | 4B0 | 2 |
| Digital 2 | 1201 | 4B1 | 2 |
| Digital 3 | 1202 | 4B2 | 2 |
| Digital 4 | 1203 | 4B3 | 2 |
| Digital 5 | 1204 | 4B4 | 2 |
| Digital 6 | 1205 | 4B5 | 2 |

### Example: Reading digital input 6, slave address 1

| Byte 0 | Byte 1<br>Status bit 0 |
|---|---|
| 00000000 | 0000000**1** |
| Always 0 | Bit 0 high<br>Digital 6 |

| Register | Value (hex) |
|---|---|
| 1205 | **0001** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 04 B5 | Register 1205 |
| | No. of registers | 00 01 | 1 register |
| | CRC | 94 DC | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Number | 02 | 2 bytes |
| | Statuses | 00 01 | Digital 6 to high |
| | CRC | 79 84 | |

## 2.6.6 Device → Modbus master: Digital channels (totalizers)

The totalizers of the digital inputs 1–6 are read out via **03 Read Holding Register (4x)**.

It is possible to transfer the value as a 32-bit float or 64-bit float.

*Register addresses of the digital input totalizers (device → Modbus master)*

| Channel | | Reg. dec. | Reg. hex. | Length, bytes | | Reg. dec. | Reg. hex. | Length, bytes |
|---|---|---|---|---|---|---|---|---|
| Digital 1 | | 1300 | 514 | 6 | | 6300 | 189C | 10 |
| Digital 2 | | 1303 | 517 | 6 | | 6305 | 18A1 | 10 |
| Digital 3 | | 1306 | 51A | 6 | | 6310 | 18A6 | 10 |
| Digital 4 | | 1309 | 51D | 6 | | 6315 | 18AB | 10 |
| Digital 5 | | 1312 | 520 | 6 | | 6320 | 18B0 | 10 |
| Digital 6 | | 1315 | 523 | 6 | | 6325 | 18B5 | 10 |

The first register (low-byte) contains the status (see → 25) and the limit value violations (see → 24) of the floating point number transmitted in the second and third registers (32-bit float).

**Example: Reading the totalizer of digital input 6 (32-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| | **00** | **80** | **40** | **C9** | **99** | **9A** |
| | Limit value violations | Floating point number status | Floating point number = 65552.0 | | | |

| Register | Value (hex) |
|---|---|
| 1315 | **0080** |
| 1316 | **40C9** |
| 1317 | **999A** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 05 23 | Register 1315 |
| | No. of registers | 00 03 | 3 registers |
| | CRC | F4 CD | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Number | 06 | 6 bytes |
| | Digital status | 00 80 40 C9 99 9A | 6.3 |
| | CRC | 0F 6E | |

The first register (low-byte) contains the status (see → 25) and the limit value violations (see → 24) of the floating point number transmitted in the second to fifth registers (64-bit float).

**Example: Reading the totalizer of digital input 6 (64-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| | **00** | **80** | **40** | **19** | **33** | **33** | **39** | **80** | **00** | **00** |
| | Limit value violations | Floating point number status | Floating point number = 6.3 (64-bit float) | | | | | | | |

| **Register** | **Value (hex)** |
|---|---|
| 6325 | **0080** |
| 6326 | **4019** |
| 6327 | **3333** |
| 6328 | **3980** |
| 6329 | **0000** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 18 B5 | Register 6325 |
| | No. of registers | 00 05 | 5 registers |
| | CRC | 92 8F | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | No. of bytes | 0A | 10 bytes |
| | Status | 0080 | |
| | FLP | 40 19 33 33 39 80 00 00 | 6.3 |
| | CRC | C5 32 | |

## 2.6.7 Device → Modbus master: Integrated universal channels (totalizers)

The totalizers of the universal inputs 1–12 are read out via **03 Read Holding Register (4x)**.

It is possible to transfer the value as a 32-bit float or 64-bit float.

*Register addresses of the universal input totalizers (device → Modbus master)*

| **Channel** | | **Reg. dec.** | **Reg. hex.** | **Length, bytes** | | **Reg. dec.** | **Reg. hex.** | **Length, bytes** |
|---|---|---|---|---|---|---|---|---|
| Universal 1 | | 800 | 320 | 6 | | 5800 | 16A8 | 10 |
| Universal 2 | | 803 | 323 | 6 | | 5805 | 16AD | 10 |
| Universal 3 | | 806 | 326 | 6 | | 5810 | 16B2 | 10 |
| Universal 4 | | 809 | 329 | 6 | | 5815 | 16B7 | 10 |
| Universal 5 | | 812 | 32C | 6 | | 5820 | 16BC | 10 |
| Universal 6 | | 815 | 32F | 6 | | 5825 | 16C1 | 10 |
| Universal 7 | | 818 | 332 | 6 | | 5830 | 16C6 | 10 |
| Universal 8 | | 821 | 335 | 6 | | 5835 | 16CB | 10 |
| Universal 9 | | 824 | 338 | 6 | | 5840 | 16D0 | 10 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Universal 10 | | 827 | 33B | 6 | | 5845 | 16D5 | 10 |
| Universal 11 | | 830 | 33E | 6 | | 5850 | 16DA | 10 |
| Universal 12 | | 833 | 341 | 6 | | 5855 | 16DF | 10 |

The first register contains the status (see → 25) and the limit value violations (see → 24) of the floating point number transmitted in the second and third registers (32-bit float).

**Example: Reading universal channel 1 totalizer with the value 26557.48633 (32-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| | **00** | **80** | **46** | **CF** | **7A** | **E6** |
| | Limit value violations | Floating point number status | Floating point number = 26557.48633 | | | |

| **Register** | **Value (hex)** |
|---|---|
| 800 | **0080** |
| 801 | **46CF** |
| 802 | **7AE6** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 03 20 | Register 800 |
| | No. of registers | 00 03 | 3 registers |
| | CRC | 04 45 | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | No. of bytes | 06 | 6 bytes |
| | Status | 00 80 | |
| | FLP | 46 CF 7A E6 | 26557.48633 |
| | CRC | E6 FE | |

The first register contains the status (see → 25) and the limit value violations (see → 24) of the floating point number transmitted in the second to fifth registers (64-bit float).

**Example: Reading universal channel 1 totalizer with the value 33174.3672951 (64-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| | **00** | **80** | **40** | **E0** | **32** | **CB** | **C0** | **E1** | **99** | **A9** |
| | Limit value violations | Floating point number status | Floating point number = 33174.3672951 (64-bit float) | | | | | | | |

| **Register** | **Value (hex)** |
|---|---|
| 5800 | **0080** |
| 5801 | **40E0** |

| | |
|---|---|
| 5802 | **32CB** |
| 5803 | **C0E1** |
| 5804 | **99A9** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 16 A8 | Register 5800 |
| | No. of registers | 00 05 | 5 registers |
| | CRC | 00 61 | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | No. of bytes | 0A | 10 bytes |
| | Status | 00 80 | |
| | FLP | 40 E0 32 CB C0 E1 99 A9 | 33174.3672951 |
| | CRC | C7 54 | |

## 2.6.8 Device → Modbus master: Integrated math channels (totalizers)

The totalizers of the math channels are read out via **03 Read Holding Register (4x)**. It is possible to transfer the value as a 32-bit float or 64-bit float.

*Register addresses of the math channels (totalizers) (device → Modbus master)*

| **Channel** | | **Reg. dec.** | **Reg. hex.** | **Length, bytes** | | **Reg. dec.** | **Reg. hex.** | **Length, bytes** |
|---|---|---|---|---|---|---|---|---|
| Math 1 | | 1700 | 6A4 | 6 | | 6700 | 1A2C | 10 |
| Math 2 | | 1703 | 6A7 | 6 | | 6705 | 1A31 | 10 |
| Math 3 | | 1706 | 6AA | 6 | | 6710 | 1A36 | 10 |
| Math 4 | | 1709 | 6AD | 6 | | 6715 | 1A3B | 10 |

The first register contains the status (see → 25) of the floating point number (32-bit float) transmitted in the second and third registers.

**Example: Reading the totalizer of digital input 1 (32-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| | **00** | **80** | **4B** | **29** | **85** | **F4** |
| | Limit value violations | Floating point number status | Floating point number = 33174.3672951 | | | |

| **Register** | **Value (hex)** |
|---|---|
| 1700 | **0080** |
| 1701 | **4B29** |
| 1702 | **85F4** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 06 A4 | Register 1700 |
| | No. of registers | 00 03 | 3 registers |
| | CRC | 44 A0 | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | No. of bytes | 06 | 6 bytes |
| | Status | 00 80 | |
| | FLP | 4B 29 85 F4 | 33174.3672951 |
| | CRC | 85 90 | |

The first register contains the status (see → 📄 25) of the floating point number (64-bit float) transmitted in the second to fifth registers.

**Example: Reading the totalizer of math 1 (64-bit float), slave address 1**

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| | **00** | **80** | **41** | **68** | **5F** | **26** | **35** | **2A** | **FC** | **7E** |
| | Limit value violations | Floating point number status | Floating point number = 33174.3672951 (64-bit float) | | | | | | | |

| Register | Value (hex) |
|---|---|
| 6700 | **0080** |
| 6701 | **4168** |
| 6702 | **5F26** |
| 6703 | **352A** |
| 6704 | **FC7E** |

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 1A 2C | Register 6700 |
| | No. of registers | 00 05 | 5 registers |
| | CRC | 43 18 | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | No. of bytes | 0A | 10 bytes |
| | Status | 00 80 | |
| | FLP | 41 68 5F 26 35 2A FC 7E | 33174.3672951 |
| | CRC | 83 06 | |

### 2.6.9 Device → Modbus master: Reading relay statuses

The statuses of the relays are read out via **03 Read Holding Register (4x)**.

Bit 0 corresponds to relay 1.

**Example: Relay 5 in active state**

| | | | |
|---|---|---|---|
| **Request:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | Register | 0C 50 | Register 3152 |
| | No. of registers | 00 01 | 1 register |
| | CRC | 87 4B | |
| **Response:** | Slave address | 01 | |
| | Function | 03 | 03: Read Holding Register |
| | No. of bytes | 02 | 2 bytes |
| | Data | 00 10 | |
| | CRC | B9 88 | |

| Byte 0<br>Status (bit 15–8) | Byte 1<br>Status (bit 7–0) |
|---|---|
| 00000000 | 000**1**0001 |
| Always 0 | Bit 4 high<br>Relay 5 |

| **Register** | **Value (hex)** |
|---|---|
| 3152 | **0010** |

The relay status is determined from the two data bytes as follows:
Byte 1:
- Bit 0 = Relay 1 status
- Bit 1 = Relay 2 status
- Bit 2 = Relay 3 status
- Bit 3 = Relay 4 status
- Bit 4 = Relay 5 status
- Bit 5 = Relay 6 status

1 = active, 0 = inactive

## 2.6.10 Structure of the process values

### 32-bit floating point number (IEEE-754)

| Octet | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|
| 0 | Sign | (E) $2^{7}$ | (E) $2^{6}$ | | | | | (E) $2^{1}$ |
| 1 | (E) $2^{0}$ | (M) $2^{-1}$ | (M) $2^{-2}$ | | | | | (M) $2^{-7}$ |
| 2 | (M) $2^{-8}$ | | | | | | | (M) $2^{-15}$ |
| 3 | (M) $2^{-16}$ | | | | | | | (M) $2^{-23}$ |

Sign = 0: Positive number
Sign = 1: Negative number

$$Value = -1^{VZ} \cdot (1 + M) \cdot 2^{E-127}$$
$$Value = -1^{VZ} \cdot \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i}\right) \cdot 2^{E-127}$$

E = exponent 8 bit, M = mantissa 23 bit

Example: 40 F0 00 00 h = 0**100 0000 1**111 0000 0000 0000 0000 0000 b

Value = $-1^0$ x $2^{129-127}$ x (1 + $2^{-1}$ + $2^{-2}$ + $2^{-3}$)

= 1 x $2^2$ x (1 + 0.5 + 0.25 + 0.125)

= 1 x 4 x 1.875 = 7.5

| Byte | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| | **00** | **80** | **40** | **F0** | **00** | **00** |
| | Limit value violations | Floating point number status | Floating point number = 7.5 | | | |

### 64-bit floating point number (IEEE-754)

| Octet | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|
| 0 | Sign | (E) 210 | (E) 29 | | | | | (E) 24 |
| 1 | (E) 23 | (E) 22 | (E) $2^1$ | (E) $2^0$ | (M) $2^{-1}$ | (M) $2^{-2}$ | (M) $2^{-3}$ | (M) $2^{-4}$ |
| 2 | (M) $2^{-5}$ | | | | | | | (M) $2^{-12}$ |
| 3 | (M) $2^{-13}$ | | | | | | | (M) $2^{-20}$ |
| 4 | (M) $2^{-21}$ | | | | | | | (M) $2^{-28}$ |
| 5 | (M) $2^{-29}$ | | | | | | | (M) $2^{-36}$ |
| 6 | (M) $2^{-37}$ | | | | | | | (M) $2^{-44}$ |
| 7 | (M) $2^{-45}$ | | | | | | | (M) $2^{-52}$ |

Sign = 0: Positive number

Sign = 1: Negative number

$$Value = -1^{VZ} \cdot (1 + M) \cdot 2^{E-1023}$$
$$Value = -1^{VZ} \cdot \left(1 + \sum_{i=1}^{52} b_{52-i} 2^{-i}\right) \cdot 2^{E-1023}$$

E = exponent 11 bit, M = mantissa 52 bit

Example: 40 1E 00 00 00 00 00 00 h

= 0**100 0000 0001** 1110 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 b

Value = $-1^0$ x $21025^{-1023}$ x (1 + $2^{-1}$ + $2^{-2}$ + $2^{-3}$)

= 1 x $2^2$ x (1 + 0.5 + 0.25 + 0.125)

= 1 x 4 x 1.875 = 7.5

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| | **00** | **80** | **40** | **1E** | **00** | **00** | **00** | **00** | **0** | 0 |
| | | Floating point number status | Floating point number = 7.5 | | | | | | | |

### Limit value violations

### Device → Modbus master

The statuses of the first eight limit values assigned to the channel are entered here.

Bit 0: 1st assigned limit value
...
Bit 7: 8th assigned limit value
Bit x = 1: Limit value violated
= 0: Limit value not violated

Example:

If a limit value for instantaneous value and a limit value for analysis 1 are assigned to universal input 1, the two limit value statuses are indicated in bit 0 and bit 1 in the measured value of universal input 1 (register 200) and the integrated universal input 1 (register 800).

| Byte | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| | **02** | **80** | **40** | **F0** | **00** | **00** |
| | Limit value violations | Floating point number status | Floating point number = 7.5 | | | |

Bit 0.0 = 0: 1st assigned limit value not violated, here limit value set to instantaneous value
Bit 0.1 = 1: 2nd assigned limit value violated, here limit value set to integrated value

### Status of the floating point number

*Device → Modbus master*

0x01 Open circuit
0x02 Input signal too high
0x03 Input signal too low
0x04 Invalid measured value
0x06 Error value
0x07 Sensor/input error
0x08 No value available (e.g. during initialization of the measurement)
0x40 Value is uncertain (error value), no limit value violated
0x41 Value is uncertain (error value), lower limit value violated or gradient decreasing
0x42 Value is uncertain (error value), upper limit value violated or gradient increasing
0x80 Value is OK, no limit value violated
0x81 Value is OK, lower limit value violated or gradient decreasing
0x82 Value is OK, upper limit value violated or gradient increasing

*Modbus master → device*

0x00..0x3F: Value invalid
0x40..0x7F: Value uncertain
0x80..0xFF: Value OK

# 3 Register overview

The register addresses are all based on 0, i.e. they correspond to the value that is transmitted in the Modbus protocol.

| Register | Value | Format | Access |
|---|---|---|---|
| 200 | Universal 1 | Status + 32-bit float | R/W |
| 203 | Universal 2 | Status + 32-bit float | R/W |
| 206 | Universal 3 | Status + 32-bit float | R/W |
| 209 | Universal 4 | Status + 32-bit float | R/W |
| 212 | Universal 5 | Status + 32-bit float | R/W |
| 215 | Universal 6 | Status + 32-bit float | R/W |
| 218 | Universal 7 | Status + 32-bit float | R/W |
| 221 | Universal 8 | Status + 32-bit float | R/W |
| 224 | Universal 9 | Status + 32-bit float | R/W |
| 227 | Universal 10 | Status + 32-bit float | R/W |
| 230 | Universal 11 | Status + 32-bit float | R/W |
| 233 | Universal 12 | Status + 32-bit float | R/W |
| 800 | Universal 1 totalizer | Status + 32-bit float | R |
| 803 | Universal 2 totalizer | Status + 32-bit float | R |
| 806 | Universal 3 totalizer | Status + 32-bit float | R |
| 809 | Universal 4 totalizer | Status + 32-bit float | R |
| 812 | Universal 5 totalizer | Status + 32-bit float | R |
| 815 | Universal 6 totalizer | Status + 32-bit float | R |
| 818 | Universal 7 totalizer | Status + 32-bit float | R |
| 821 | Universal 8 totalizer | Status + 32-bit float | R |
| 824 | Universal 9 totalizer | Status + 32-bit float | R |
| 827 | Universal 10 totalizer | Status + 32-bit float | R |
| 830 | Universal 11 totalizer | Status + 32-bit float | R |
| 833 | Universal 12 totalizer | Status + 32-bit float | R |
| 1200 | Digital 1 status | 2 bytes | R/W |
| 1201 | Digital 2 status | 2 bytes | R/W |
| 1202 | Digital 3 status | 2 bytes | R/W |
| 1203 | Digital 4 status | 2 bytes | R/W |
| 1204 | Digital 5 status | 2 bytes | R/W |
| 1205 | Digital 6 status | 2 bytes | R/W |
| 1240 | Digital 1–6 statuses | 2 bytes | R/W |
| 1300 | Digital 1 totalizer | Status + 32-bit float | R |
| 1303 | Digital 2 totalizer | Status + 32-bit float | R |
| 1306 | Digital 3 totalizer | Status + 32-bit float | R |
| 1309 | Digital 4 totalizer | Status + 32-bit float | R |
| 1312 | Digital 5 totalizer | Status + 32-bit float | R |
| 1315 | Digital 6 totalizer | Status + 32-bit float | R |
| 1500 | Math 1 | Status + 32-bit float | R |
| 1503 | Math 2 | Status + 32-bit float | R |

| Register | Value | Format | Access |
|---|---|---|---|
| 1506 | Math 3 | Status + 32-bit float | R |
| 1509 | Math 4 | Status + 32-bit float | R |
| 1700 | Math 1 totalizer | Status + 32-bit float | R |
| 1703 | Math 2 totalizer | Status + 32-bit float | R |
| 1706 | Math 3 totalizer | Status + 32-bit float | R |
| 1709 | Math 4 totalizer | Status + 32-bit float | R |
| 1800 | Math 1-4 statuses | 2 bytes | R |
| 3152 | Relay statuses | 2 bytes | R |
| 5200 | Universal 1 | Status + 64-bit float | R/W |
| 5205 | Universal 2 | Status + 64-bit float | R/W |
| 5210 | Universal 3 | Status + 64-bit float | R/W |
| 5215 | Universal 4 | Status + 64-bit float | R/W |
| 5220 | Universal 5 | Status + 64-bit float | R/W |
| 5225 | Universal 6 | Status + 64-bit float | R/W |
| 5230 | Universal 7 | Status + 64-bit float | R/W |
| 5235 | Universal 8 | Status + 64-bit float | R/W |
| 5240 | Universal 9 | Status + 64-bit float | R/W |
| 5245 | Universal 10 | Status + 64-bit float | R/W |
| 5250 | Universal 11 | Status + 64-bit float | R/W |
| 5255 | Universal 12 | Status + 64-bit float | R/W |
| 5800 | Universal 1 totalizer | Status + 64-bit float | R |
| 5805 | Universal 2 totalizer | Status + 64-bit float | R |
| 5810 | Universal 3 totalizer | Status + 64-bit float | R |
| 5815 | Universal 4 totalizer | Status + 64-bit float | R |
| 5820 | Universal 5 totalizer | Status + 64-bit float | R |
| 5825 | Universal 6 totalizer | Status + 64-bit float | R |
| 5830 | Universal 7 totalizer | Status + 64-bit float | R |
| 5835 | Universal 8 totalizer | Status + 64-bit float | R |
| 5840 | Universal 9 totalizer | Status + 64-bit float | R |
| 5845 | Universal 10 totalizer | Status + 64-bit float | R |
| 5850 | Universal 11 totalizer | Status + 64-bit float | R |
| 5855 | Universal 12 totalizer | Status + 64-bit float | R |
| 6300 | Digital 1 totalizer | Status + 64-bit float | R |
| 6305 | Digital 2 totalizer | Status + 64-bit float | R |
| 6310 | Digital 3 totalizer | Status + 64-bit float | R |
| 6315 | Digital 4 totalizer | Status + 64-bit float | R |
| 6320 | Digital 5 totalizer | Status + 64-bit float | R |
| 6325 | Digital 6 totalizer | Status + 64-bit float | R |
| 6700 | Math 1 totalizer | Status + 64-bit float | R |
| 6705 | Math 2 totalizer | Status + 64-bit float | R |
| 6710 | Math 3 totalizer | Status + 64-bit float | R |
| 6715 | Math 4 totalizer | Status + 64-bit float | R |

# 4 Troubleshooting

## 4.1 Troubleshooting for Modbus TCP

- Is the Ethernet connection between the device and master correct?
- Does the IP address sent by the master match the address configured on the device?
- Do the port configured on the master and the port configured on the device match?

## 4.2 Troubleshooting for Modbus RTU

- Do the device and master have the same baudrate and parity?
- Is the interface correctly wired?
- Does the device address sent by the master match the configured address of the device?
- Do all the slaves on the Modbus have different device addresses?

# 5 List of abbreviations/definition of terms

Modbus Master: All instruments such as a PLC, PC plug-in cards etc. that perform a Modbus Master function.

# Index


71644471