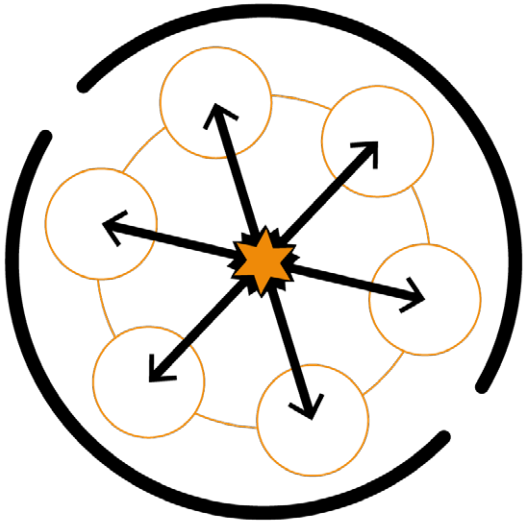
Document number: DB-KU-100249-2
Initial creation: March 2024



Modbus Quick Start Guide

Rethink Sensing

Contents

General information

Modbus specifications

For detailed information about the Modbus specifications, refer to the following documents:

- Modbus_over_serial_line_V1_02.pdf
- Modbus_Application_Protocol_V1_1b3.pdf

Modbus test software

A wide variety of Modbus test software and Modbus libraries for C++, Python or other programming languages are available on the Internet.

Additionally, you can find the TrueDyne Sensors AG tool "TDS Remote Control" including instructions for a free download from our website.

Default settings:

Baud rate	19200 BAUD
Data bits	8
Parity	Even
Byte order	1-0-3-2
Stop bits	1 bit
Modbus address	247
FlowControl	None (0)
Transmission type	Modbus RTU (protocol)

The device has neither line polarization nor line termination. An external line termination is required.

Modbus RTU function codes

Code	Name	Description
0x01	Read Coils	Read one or more coils
0x03	Read Holding Registers	Read a consecutive holding register block
0x04	Read Input Registers	Read one or more successive registers
0x05	Write Single Coil	Write one coil
0x06	Write single register	Write one single register
0x0F	Write Multiple Coils	Write multiple successive coils
0x10	Write Multiple Registers	Write multiple successive registers

For detailed information about these functions, refer to the document "Modbus_Application_Protocol_V1_1b3.pdf"

For our sensors, reading any register is carried out with either command 0x03 or 0x04. There is no difference in the handling of the information between these two commands.

The following Modbus RTU functions are not supported

- 0x02 Read Discrete Inputs
- 0x07 Read Exception Status
- 0x08 Diagnostics
- 0x0B Get Comm Event Counter
- 0x0C Get Comm Event Log

When addressing the devices, it is mandatory to ensure that no two devices have the same address. In such a case, the entire serial bus may behave abnormally, since the master is then no longer able to communicate with all existing slaves on the bus.

The following differences from the "Modbus over serial line V1.02" protocol exist

- 3.6 Cables - The cable strands are not twisted relative to each other
- 3.7 Visual Diagnostics - There is no LED display on the sensor
- No line polarization is required for the sensor, nor is provision made for one.

At least 32 sensors are supported in the bus system.

Data display

Each Modbus register contains two bytes, the data length of a command and a response is always a multiple of two bytes (one register).

ByteOrder

Description: Use this function to select the sequence in which the bytes are transmitted. The transmission sequence must be coordinated with the Modbus master.

The sequence (ByteOrder) is not standardized by the Modbus protocol. However, if the host system and the measuring device do not use the same byte order, correct data exchange is not possible.

Changing the order (ByteOrder) in the host system often requires extensive knowledge and considerable programming effort. This is why the ByteOrder parameter was introduced.

This makes it possible to use the host system's default settings and modify the byte sequence on the sensor through trial and error. If correct data exchange is not achieved, the host system byte order settings must be adjusted accordingly.

ByteOrder

The byte addressing, i.e. the transmission sequence of the bytes, is set forth in the Modbus Specifications. For this reason, it is important to coordinate or adjust the addressing between master and slave during commissioning. This can be configured in the sensor using the sequence (ByteOrder) parameter (refer to the respective sensor data sheet). Depending on the selection, the bytes are transferred in the ByteOrder parameter.

For overview tables for Integer, String and Float, refer to Attachment "A".

Error handling

Transmission errors (corrupt sequences) are detected and rejected. The sensor waits for a subsequent, correct sequence. Errors on the application level receive an error message as a response. If the response consists of an error code, the leading bit (0x80) of the function code is set to signal the error condition.

Error code (hex)	Error type
0x00	No error
0x01	An invalid function code was sent to the sensor
0x02	Invalid data address (invalid register number, access denied)
0x03	Invalid data value (value outside range)
0x04	Slave device error (operation not completed successfully)

Error code 0x01 is returned if a function code other than 0x03, 0x04, 0x016 is sent to the sensor

Addressing of the Modbus registers:

For all TrueDyne sensors, the Modbus addresses are 0-based. This means that the 1st register has the address 0.

Note that there are control units and devices on the market that are 1-based. This means that the 1st register has the address 1.

User levels, password protection

The sensors have three implemented user levels, "Operator", "Maintenance" and "Service" level. Reading of registers is possible at any user level, except for some specific registers.

Writing registers of the sensor typically means changing the configuration, which also changes the behavior of the sensor. To avoid unwanted configuration changes, most write attempts are only possible from the "Maintenance" user level.

For the user level "Maintenance" you need the default password (8646). The "Service" user level is only available for TrueDyne Sensors AG. Passwords cannot be changed.

Permanent data storage

When writing to a Modbus register, in most cases, the written parameter is stored permanently in non-volatile EEPROM.

EEPROM memory chips allow approx. 100,000 write operations per parameter. If this limit is exceeded, the memory chip may be damaged, and correct function of the sensor can no longer be guaranteed as a result. A sensor with corrupt data in its memory is no longer functional.

Byte order tables

Integer

	Sequence	
Options	1.	2.
1–0–3–2* 3–2–1–0	Byte 1 (MSB)	Byte 0 (LSB)
0–1–2–3 2–3–0–1	Byte 0 (LSB)	Byte 1 (MSB)

** = factory setting, MSB = most significant byte, LSB = least significant byte*

String

	Sequence				
Options	1.	2.	...	17.	18.
1–0–3–2* 3–2–1–0	Byte 17 (MSB)	Byte 16	...	Byte 1	Byte 0 (LSB)
0–1–2–3 2–3–0–1	Byte 16	Byte 17 (MSB)	...	Byte 0 (LSB)	Byte 1

Representation using the example of a device parameter with a data length of 18 bytes
** = factory setting, MSB = most significant byte, LSB = least significant byte*

Float

	Sequence			
Options				
1-0-3-23	Byte 1 (MMMMMMMM)	Byte 0 (MMMMMMMM)	Byte 3 (SEEEEEEE)	Byte 2 (EMMMMMMM)
0-1-2-3	Byte 0 (MMMMMMMM)	Byte 1 (MMMMMMMM)	Byte 2 (EMMMMMMM)	Byte 3 (SEEEEEEE)
2-3-0-1	Byte 2 (EMMMMMMM)	Byte 3 (SEEEEEEE)	Byte 0 (MMMMMMMM)	Byte 1 (MMMMMMMM)
3-2-1-0	Byte 3 (SEEEEEEE)	Byte 2 (EMMMMMMM)	Byte 1 (MMMMMMMM)	Byte 0 (MMMMMMMM)

**default, S = sign, E = exponent, M = mantissa*

Detailed Modbus register information can be found on the specific sensor data sheets. Especially important information for ensuring communication settings includes the following: Address, Baudrate, Parity, StopBit, ByteOrder.

Phone +41 61 715 89 00 ▪ info@truedyne.com ▪ www.truedyne.com