IST

IST

RS485 Temperature & Humidity (SHT45-based) Sensor Probe

Version 1.0

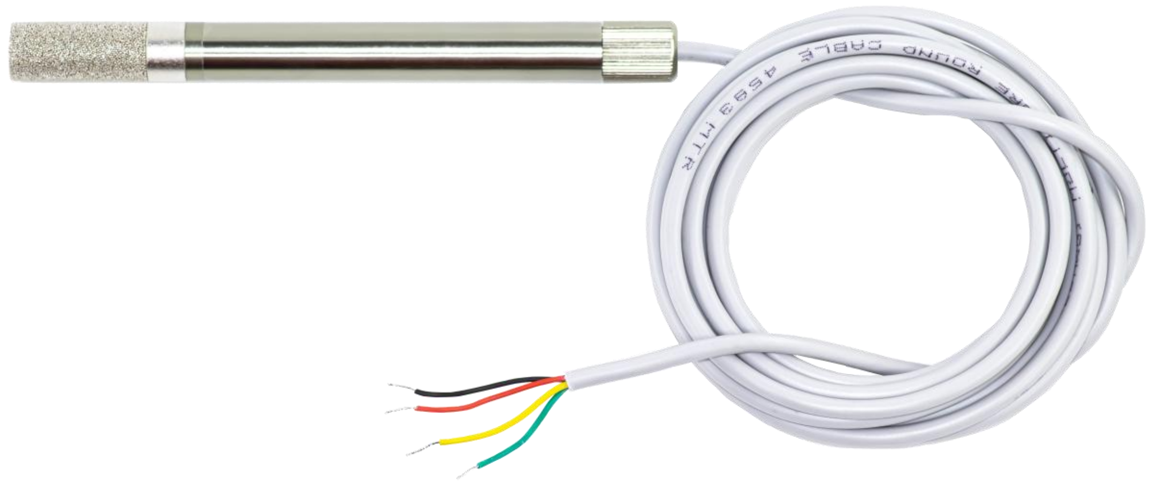
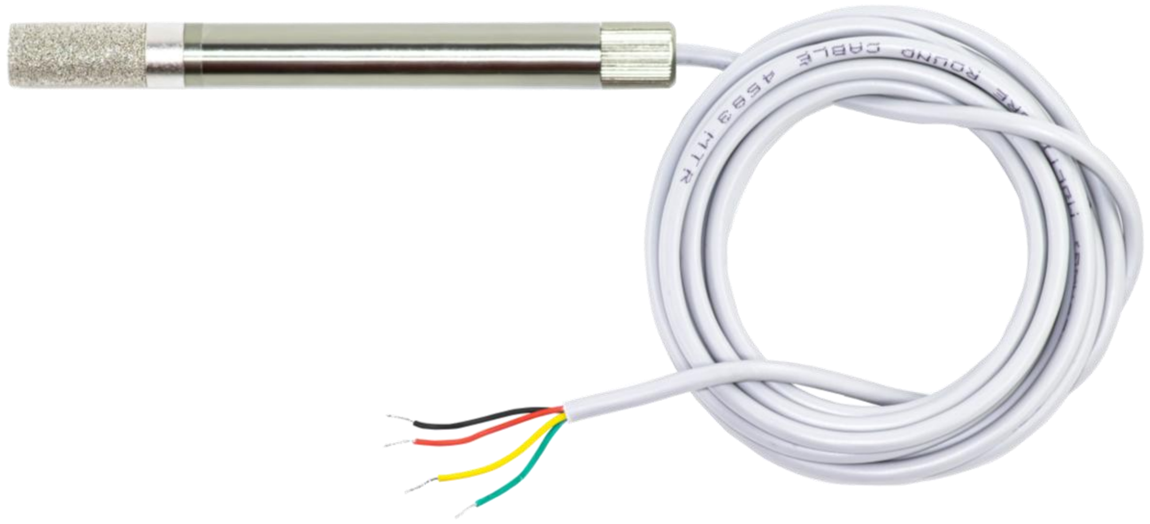


Table of Contents

1.Introduction



This sensor is a high-precision industrial-grade RS485 temperature and humidity measurement device utilizing the advanced sensor. Designed for robustness, reliability, and accurate digital signal processing, it converts ambient temperature and humidity into reliable RS485 signals for seamless integration with centralized monitoring systems.

The sensor supports temperature measurements from -40°C to 125°C and humidity from 0% to 100% RH, with high accuracy and fast response. Encased in an aluminum alloy shell with waterproof, breathable, and dust-proof features, this probe is suitable for demanding environmental conditions.Ideal for HVAC systems, industrial environments, building automation, climatology stations, museums, hotels, and closed-loop control systems.

1.1 Features

- Based on the high-accuracy SHT45 sensor
- RS485 Modbus RTU output
- Excellent long-term stability
- Wide voltage input: 9–36V DC
- DIP switch configurable Modbus address and baud rate
- Fast response and high resolution
- Waterproof and breathable sensor design

2.Technical specification

The 7Semi RS485 Temperature & Humidity Sensor is engineered to provide precise real-time environmental data in a rugged, industrial-grade form factor. Integrating the advanced SHT45 sensor, it offers high-resolution temperature and humidity readings over a robust RS485 interface using the Modbus RTU protocol. This sensor is optimized for long-term performance and reliability in demanding conditions.

Its digital output enables straightforward integration with industrial control systems, SCADA platforms, and data loggers. The onboard signal conditioning and Modbus address/baud rate configurability via DIP switch further simplify deployment in diverse applications ranging from environmental monitoring to HVAC automation and industrial process control.

Applications

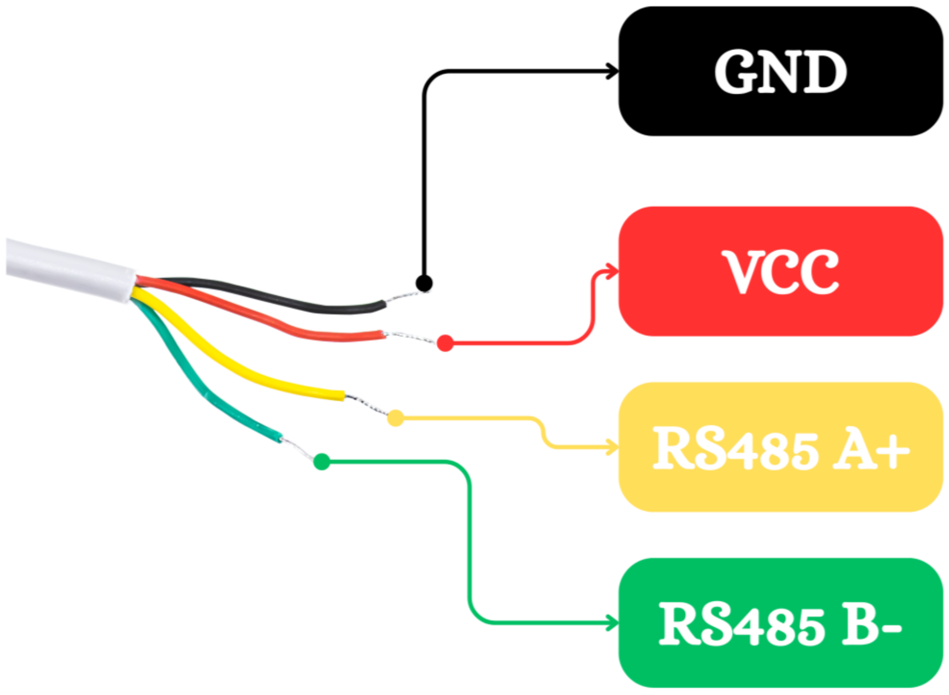
- HVAC control
- Building automation
- Warehouses and greenhouse
- Museums and archive
- Industrial process monitoring
- Weather stations and environmental studies

2.1 General specifications

- **Temperature Measurement Range:** -40 ~ 125°C
- **Humidity Measurement Range:** 0 ~ 100% RH
- **Temperature Accuracy:** ±0.1°C (25°C)
- **Humidity Accuracy:** ±1.5% RH (25°C)
- **Sampling Cycle Period:** 3 sec
- **Power Supply Voltage:** 9 ~ 36V (DC)
- **Product Size:** 200mm(L) × 15.7mm(D) / 7.87 × 0.62 inch
- **Output Signal:** RS485 signal
- **Communication Protocol:** Standard MODBUS RTU protocol
- **Baud Rate:** 9600 (default); configurable to 19200, 38400, 115200
- **Display Resolution:** Temperature: 0.1°C; Humidity: 0.1% RH
- **Sensitivity Attenuation Value:**
    - Temperature: < 0.1°C / year
    - Humidity: < 0.5% RH / year

3. Hardware setup

3.1 Pinouts

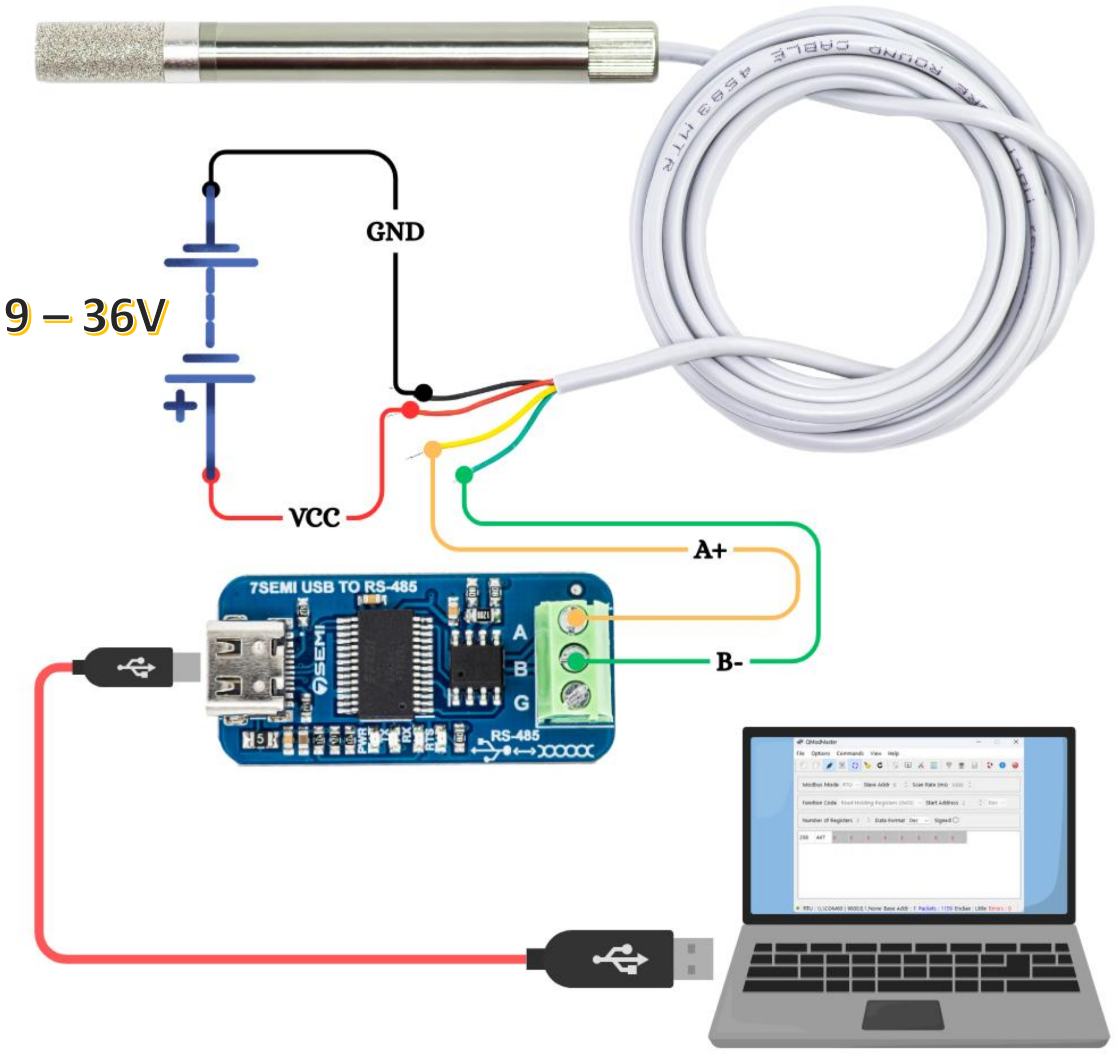


Wire Color	Function
Yellow	RS485 A+
Green	RS485 B−
Red	VCC
Black	GND

Connection Guidelines

- Power Supply: Connect the red wire to a 6–48V DC power source and the black wire to ground (GND). Ensure the power supply is stable and within the rated voltage range.
- RS485 Communication: Connect the yellow wire to RS485 A+ and the white wire to RS485 B−. Use twisted-pair shielded cable for longer transmission distances and improved noise immunity.

3.2 Hardware connection



4.RS485 Communication Protocol

The 7Semi RS485 Temperature & Humidity Sensor utilizes the Modbus RTU protocol for digital communication over the RS485 interface. Modbus RTU is a robust, standardized protocol used widely in industrial automation for reliable communication between field devices and control systems such as PLCs, HMIs, SCADA systems, and data loggers.

This sensor acts as a Modbus slave, waiting for requests from a Modbus master and responding with the requested data. All Modbus messages are transmitted in a binary format with CRC (Cyclic Redundancy Check) error detection to ensure communication integrity

Supported Modbus Function Codes

Function Code	Description
`0x03`	Read Holding Registers

- The sensor currently supports **function code `0x03`**, which allows reading holding register values (such as temperature and humidity).

Register Address Mapping

Register Address	Parameter	Data Type	Scaling	Unit
`0x0001`	Temperature	16-bit Unsigned	Value ÷ 10	Degrees Celsius
`0x0002`	Humidity	16-bit Unsigned	Value ÷ 10	Percent RH

- The sensor provides temperature and humidity data in 16-bit unsigned integer format. To convert the raw data into human-readable values, divide the received value by 10.

Example Modbus RTU Request

To read temperature and humidity (2 registers) from slave address `01`:

```
[01] [03] [00] [01] [00] [02] [CRC_L] [CRC_H]
```

Byte Position	Value	Description
1	01	Slave address
2	03	Function code: Read Holding Registers
3–4	00 01	Start register address (temperature)
5–6	00 02	Number of registers to read (2)
7–8	CRC_L, CRC_H	CRC16 checksum (Low byte first)

If temperature is 25.0°C and humidity is 60.0% RH, the response will be:

```
[01] [03] [04] [00] [FA] [02] [58] [CRC_L] [CRC_H]
```

Byte Position	Value	Description
1	01	Slave address
2	03	Function code
3	04	Byte count
4–5	00 FA	Temperature = 0x00FA = 250 → 25.0°C
6–7	02 58	Humidity = 0x0258 = 600 → 60.0% RH
8–9	CRC_L, CRC_H	CRC16 checksum (Low byte first)

CRC16 Checksum Calculation

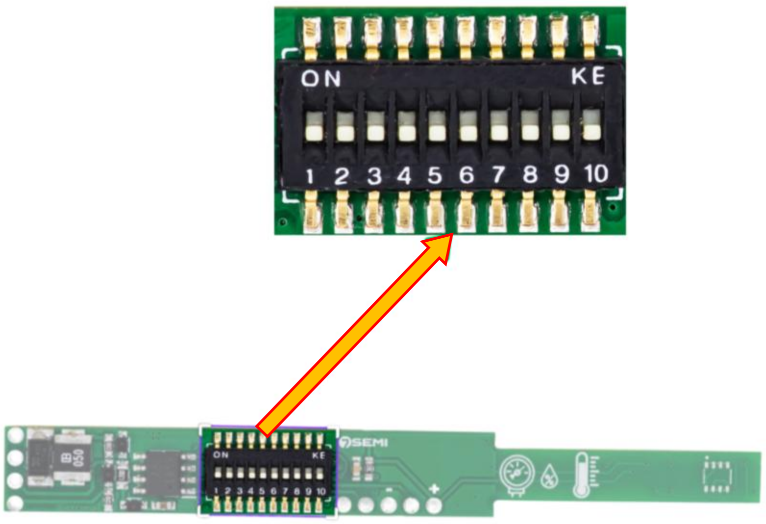
Modbus RTU uses a **CRC16 (Cyclic Redundancy Check)** algorithm for error detection. The CRC is calculated over all bytes in the message (excluding the CRC itself). The sensor uses the standard Modbus CRC16 method with the following polynomial: **`0xA001`**.

C code:

```
uint16_t CRC16(uint8_t *buf, uint8_t len) {

    uint16_t crc = 0xFFFF;
    for (uint8_t i = 0; i < len; i++) {
        crc ^= buf[i];
        for (uint8_t j = 0; j < 8; j++) {
            if (crc & 1)
                crc = (crc >> 1) ^ 0xA001;
            else
                crc >>= 1;
        }
    }
    return crc;
}
```

DIP Switch Configuration



Use the internal 8-digit DIP switch to set the Modbus address (binary representation):

- Switches 1-8 correspond to values: 1, 2, 4, 8, 16, 32, 64, 128.
- Address = Sum of switch values set to ON.

Example:

- Switch 1 ON = Address 1
- Switches 1, 3, 4 ON = Address (1 + 4 + 8) = 13

Baud Rate Setting

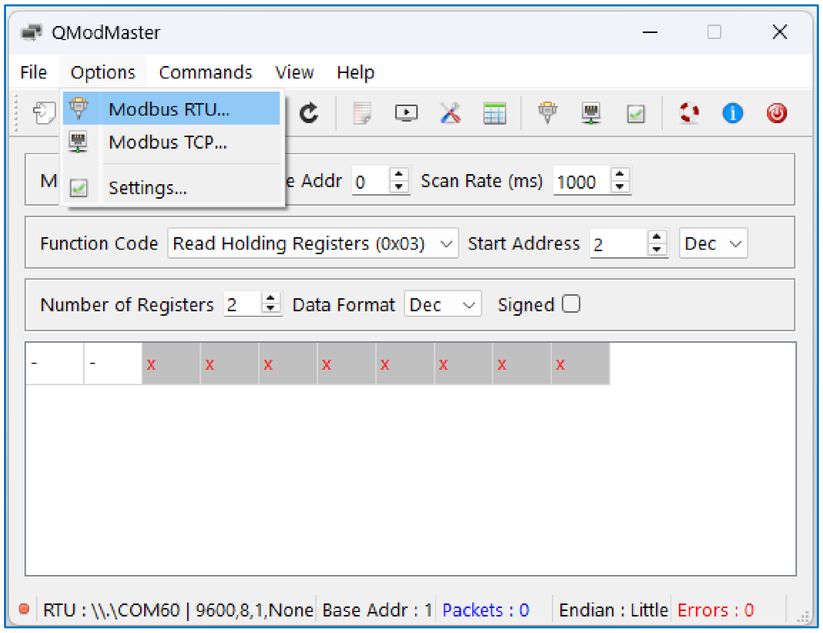
Two additional switches allow selection of the baud rate:

Binary Code	Baud Rate
00	9600 bps
01	19200 bps
10	38400 bps
11	115200 bps

4.1 Modbus output

Follow the steps below for the software setup:-

1. Open the software `(qModMaster-Win64-exe-0.5.3-beta)`
2. Go to *Options* → *Modbus RTU*



M

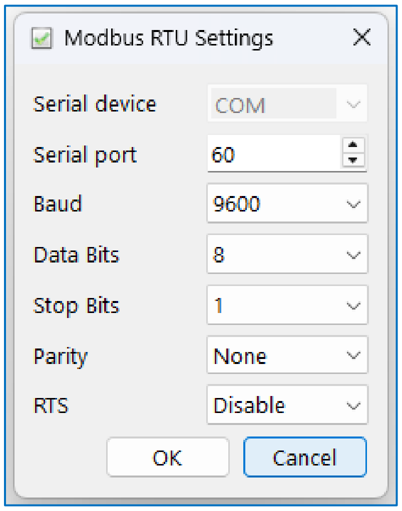
e Addr 0 Scan Rate (ms) 1000

Function Code Read Holding Registers (0x03) Start Address 2 Dec

Number of Registers 2 Data Format Dec Signed ☐

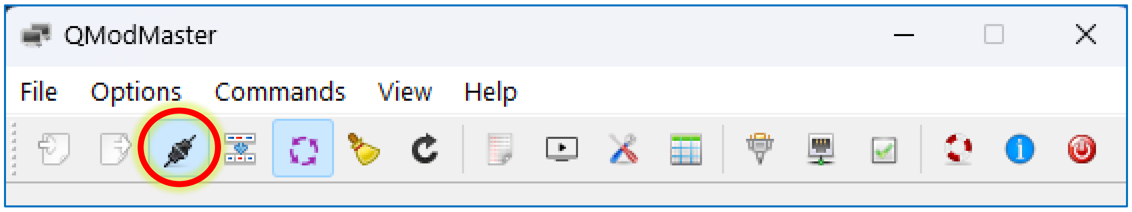
RTU : \\.\COM60 | 9600,8,1,None Base Addr : 1 Packets : 0 Endian : Little Errors : 0

3. Check your COM port in device manager and select the other settings as per the image below. Click *OK*

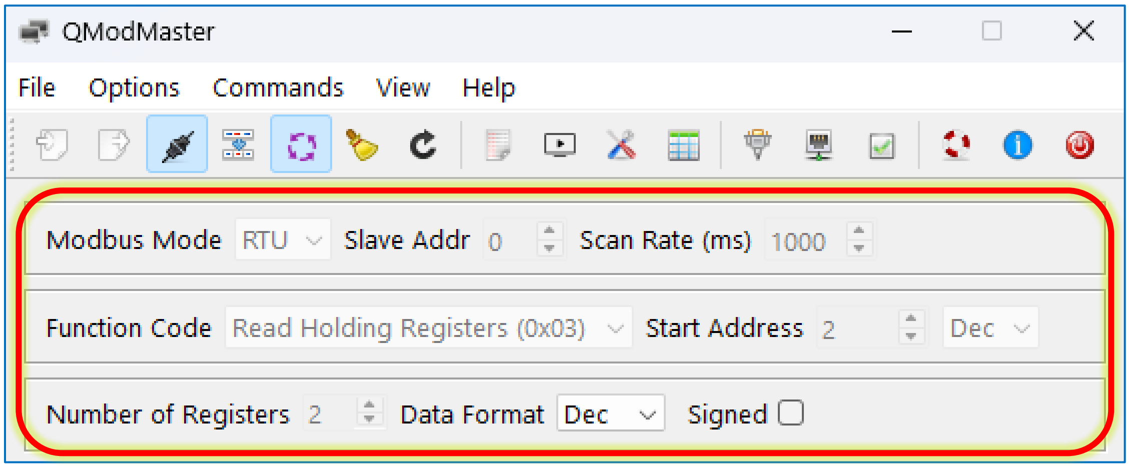


Serial device	COM
Serial port	60
Baud	9600
Data Bits	8
Stop Bits	1
Parity	None
RTS	Disable

4. Connect your system to the correct COM port and click on this icon.

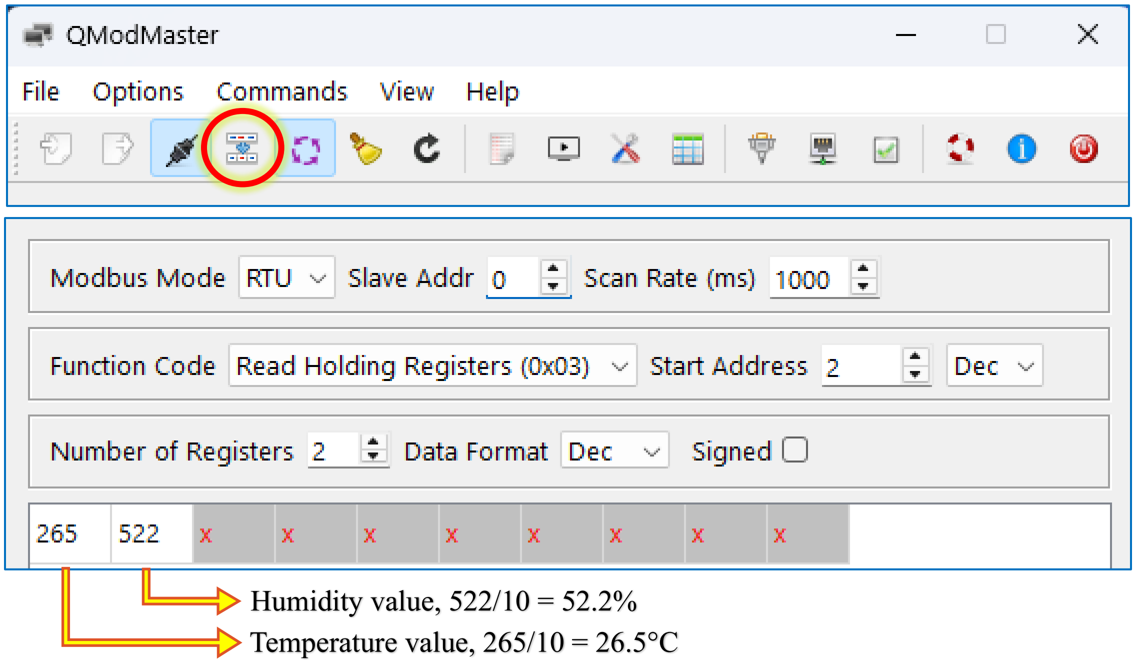


5. Also, make sure the following parameters are set properly.



Modbus Mode RTU · Slave Addr 0 · Scan Rate (ms) 1000
Function Code Read Holding Registers (0x03) · Start Address 2 · Dec
Number of Registers 2 · Data Format Dec · Signed ☐

6. Now to check the data, click on this icon.

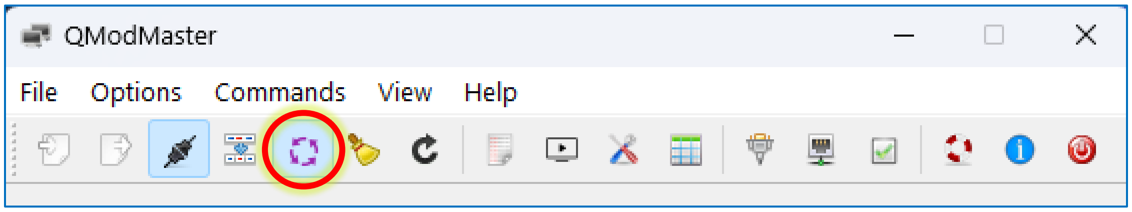


Modbus Mode RTU · Slave Addr 0 · Scan Rate (ms) 1000
Function Code Read Holding Registers (0x03) · Start Address 2 · Dec
Number of Registers 2 · Data Format Dec · Signed ☐

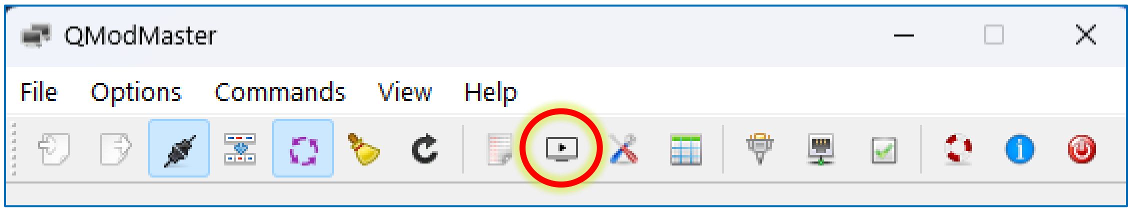
265	522	x	x	x	x	x	x	x	x

Humidity value, 522/10 = 52.2%
Temperature value, 265/10 = 26.5°C

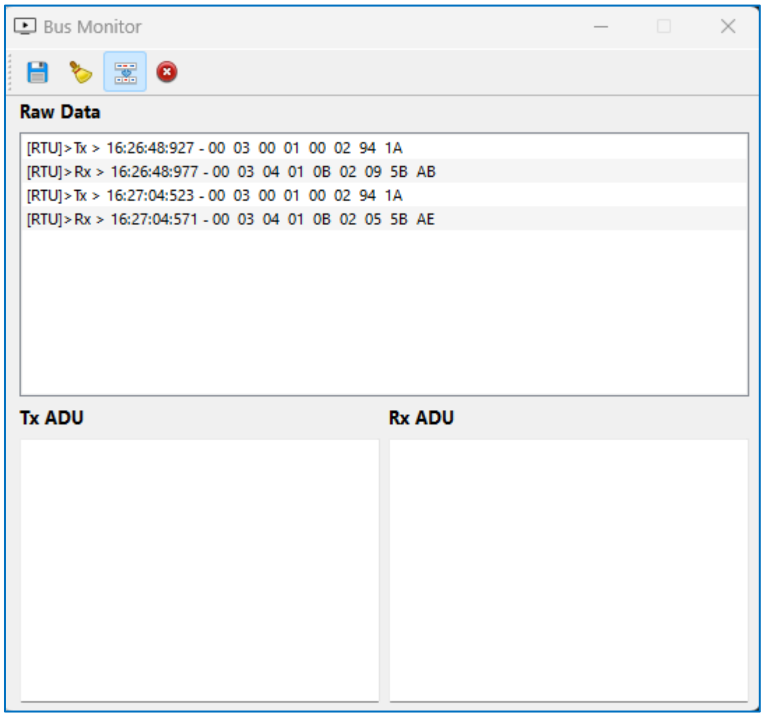
7. If you want to display the values continuously based upon the scan rate then click on this icon.



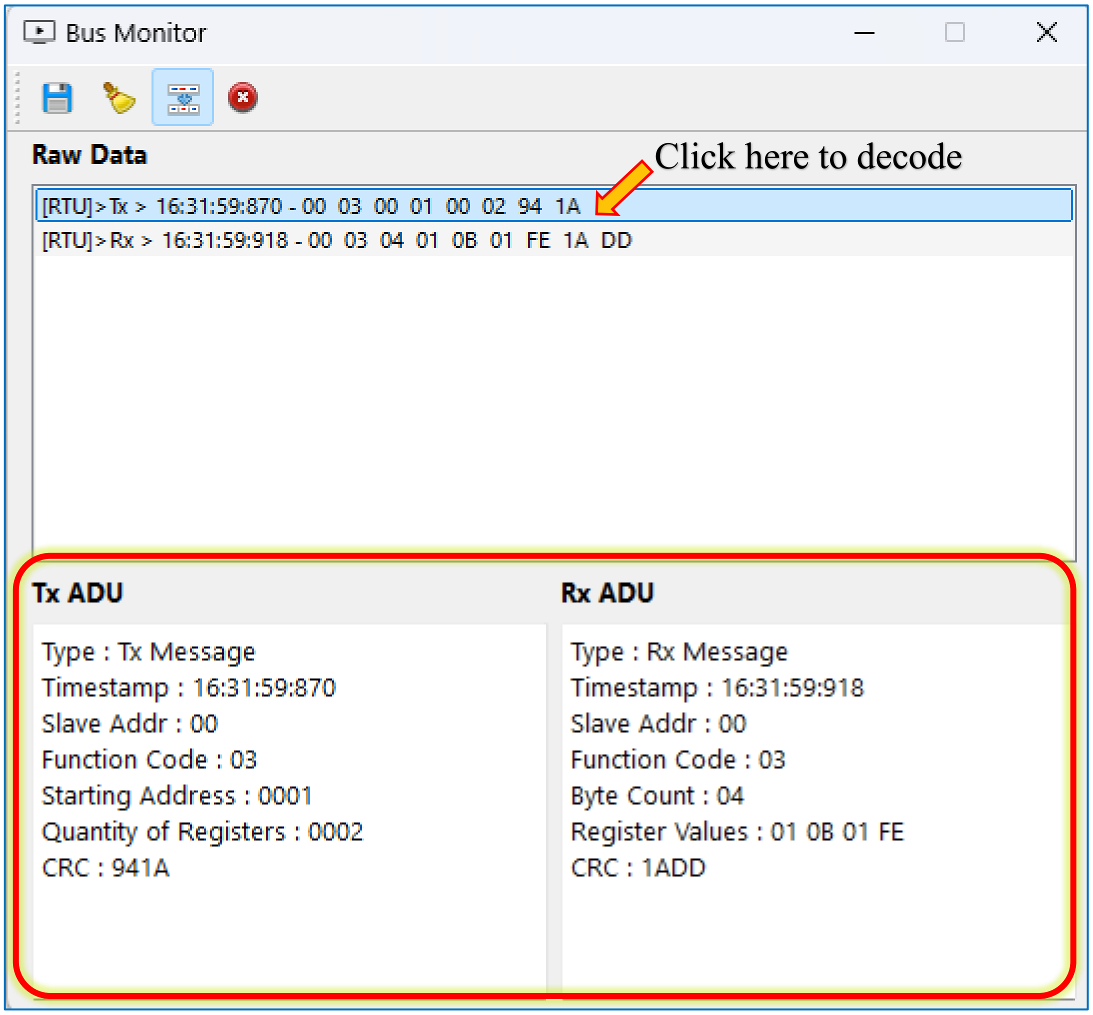
If you want to see the values in more detail you can click on this icon.



A new window will open to display the readings. To display data, follow step 6.



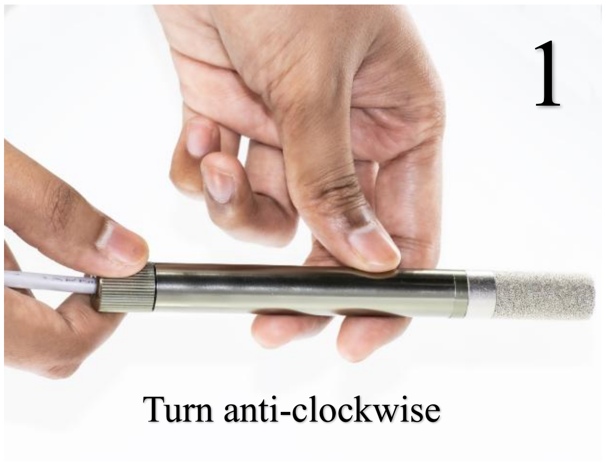
```
[RTU]>Tx > 16:26:48:927 - 00  03  00  01  00  02  94  1A
[RTU]>Rx > 16:26:48:977 - 00  03  04  01  0B  02  09  5B  AB
[RTU]>Tx > 16:27:04:523 - 00  03  00  01  00  02  94  1A
[RTU]>Rx > 16:27:04:571 - 00  03  04  01  0B  02  05  5B  AE
```

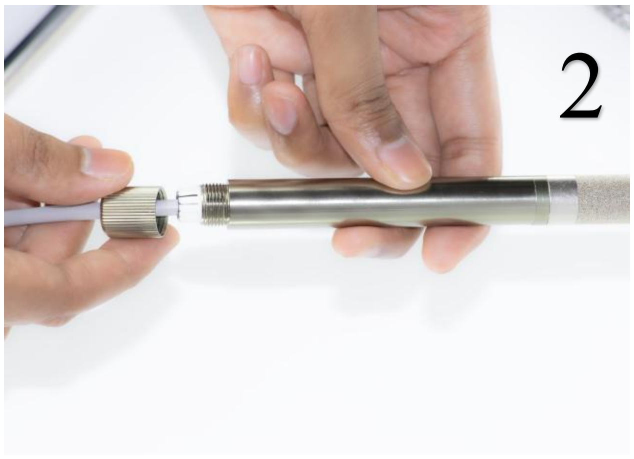


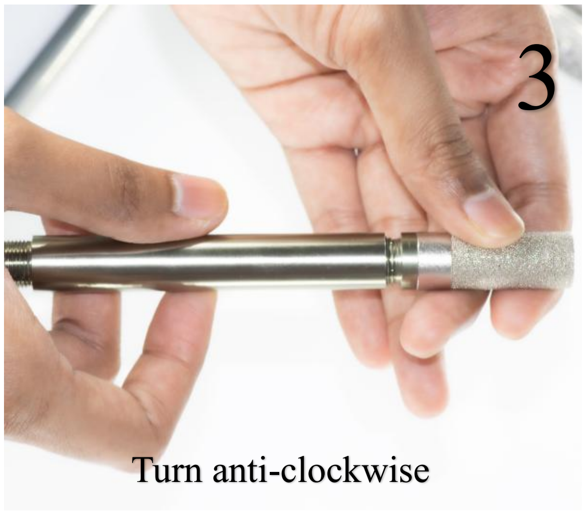
```
[RTU]>Tx > 16:31:59:870 - 00  03  00  01  00  02  94  1A
[RTU]>Rx > 16:31:59:918 - 00  03  04  01  0B  01  FE  1A  DD
```

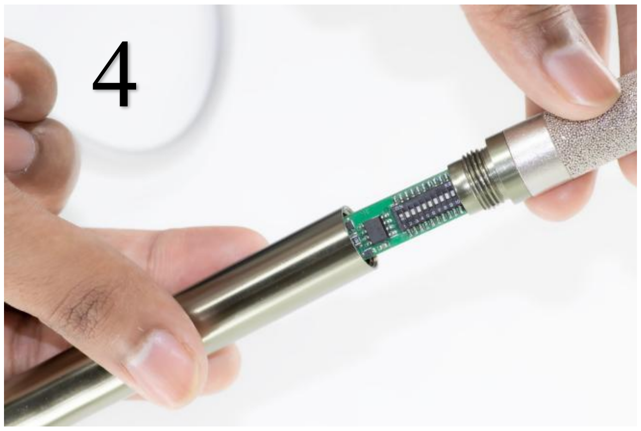
Tx ADU	Rx ADU
Type : Tx Message	Type : Rx Message
Timestamp : 16:31:59:870	Timestamp : 16:31:59:918
Slave Addr : 00	Slave Addr : 00
Function Code : 03	Function Code : 03
Starting Address : 0001	Byte Count : 04
Quantity of Registers : 0002	Register Values : 01 0B 01 FE
CRC : 941A	CRC : 1ADD

4.2 How to open the probe?

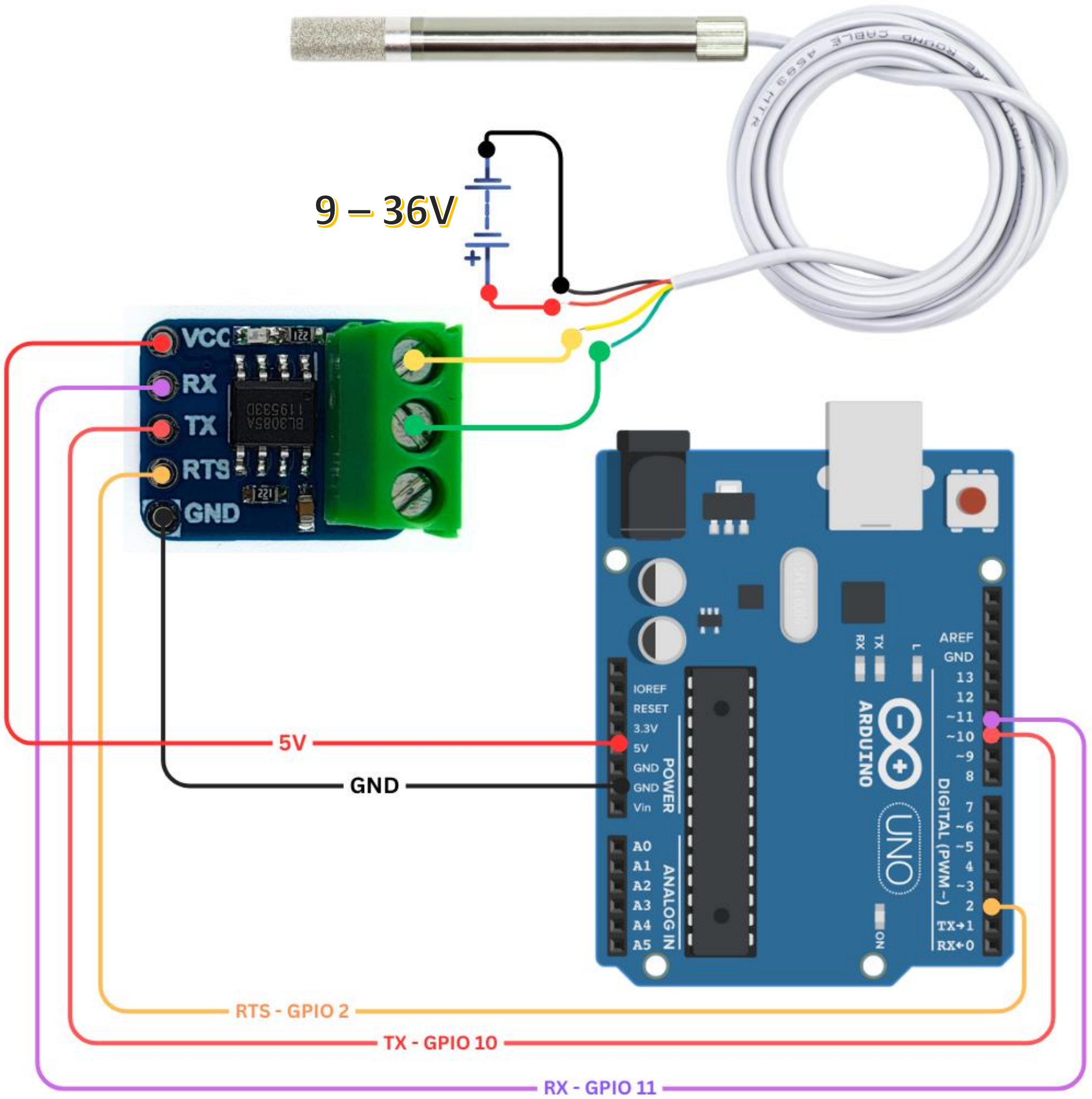






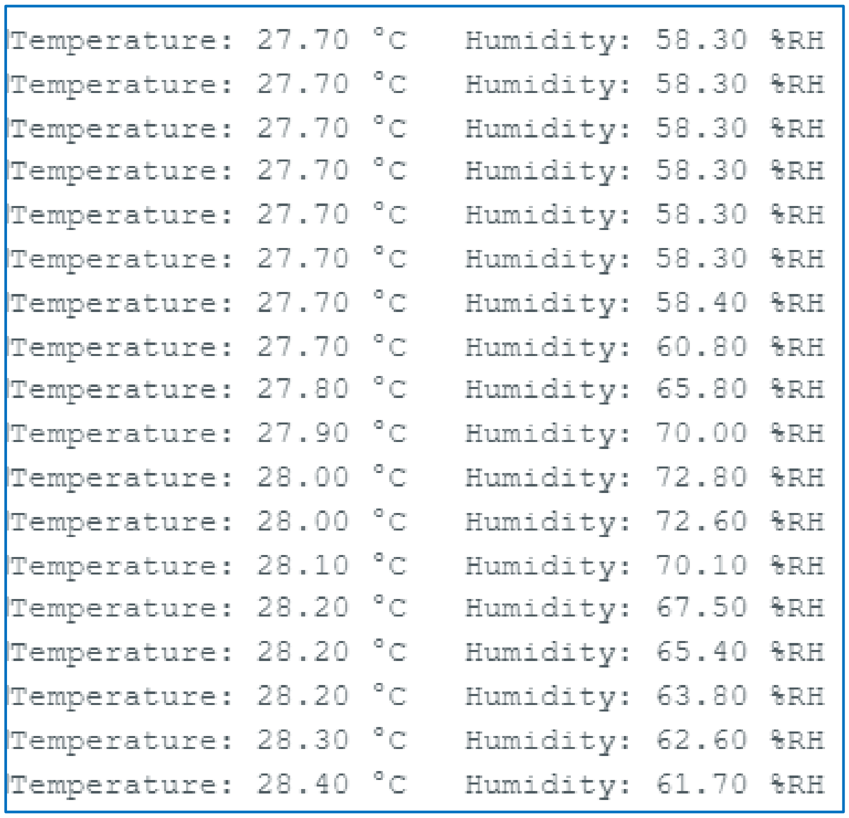


4.3 Connection and sample code for Arduino



Sample code link:- RS485-sensor-probe

Sample output image:-



```
Temperature: 27.70 °C   Humidity: 58.30 %RH
Temperature: 27.70 °C   Humidity: 58.30 %RH
Temperature: 27.70 °C   Humidity: 58.30 %RH
Temperature: 27.70 °C   Humidity: 58.30 %RH
Temperature: 27.70 °C   Humidity: 58.30 %RH
Temperature: 27.70 °C   Humidity: 58.30 %RH
Temperature: 27.70 °C   Humidity: 58.40 %RH
Temperature: 27.70 °C   Humidity: 60.80 %RH
Temperature: 27.80 °C   Humidity: 65.80 %RH
Temperature: 27.90 °C   Humidity: 70.00 %RH
Temperature: 28.00 °C   Humidity: 72.80 %RH
Temperature: 28.00 °C   Humidity: 72.60 %RH
Temperature: 28.10 °C   Humidity: 70.10 %RH
Temperature: 28.20 °C   Humidity: 67.50 %RH
Temperature: 28.20 °C   Humidity: 65.40 %RH
Temperature: 28.20 °C   Humidity: 63.80 %RH
Temperature: 28.30 °C   Humidity: 62.60 %RH
Temperature: 28.40 °C   Humidity: 61.70 %RH
```

5.Mechanical specification