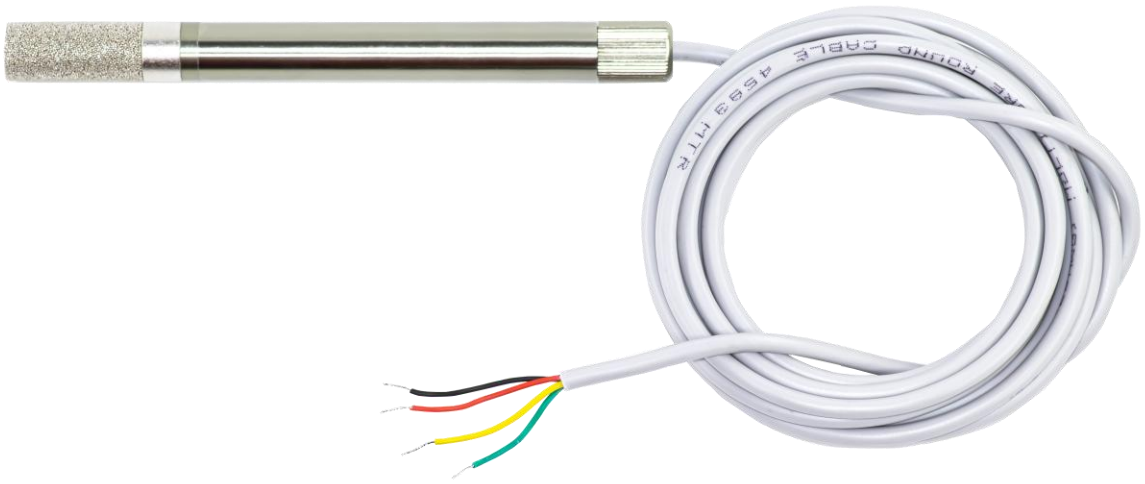
7Semi

RS485 Temperature & Humidity Sensor Probe

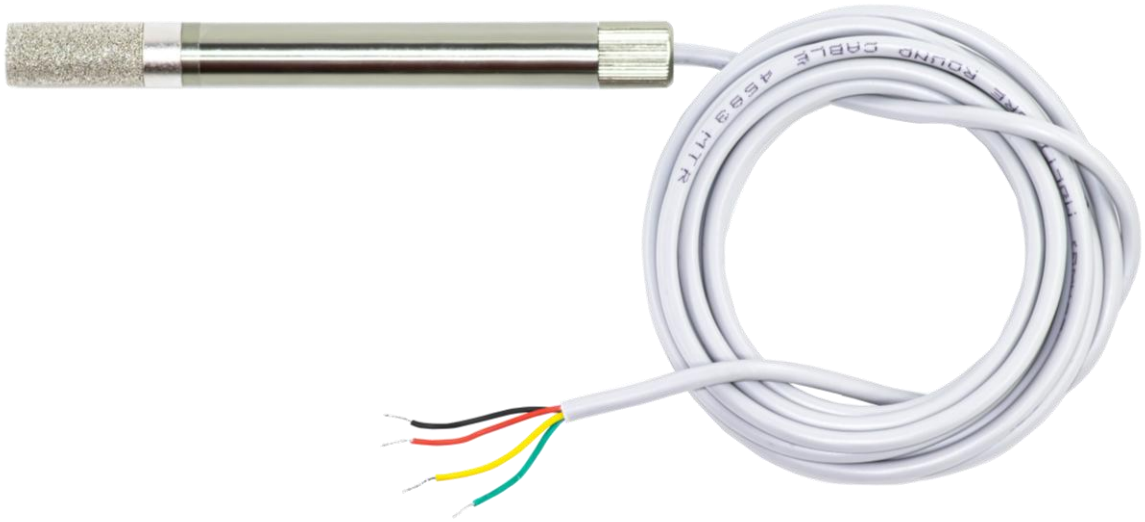
Version 2.0



7Semi +91 8655821342 info@7semi.com www.7semi.com

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 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 temperature & humidity sensor
- RS485 Modbus RTU output
- Excellent long-term stability
- Wide voltage input: 9–36V DC
- Software configurable Modbus address and baud rate
- Fast response and high resolution
- 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 advance 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 Software further simplify deployment in diverse applications ranging from environmental monitoring to HVAC automation and industrial process control.

Applications

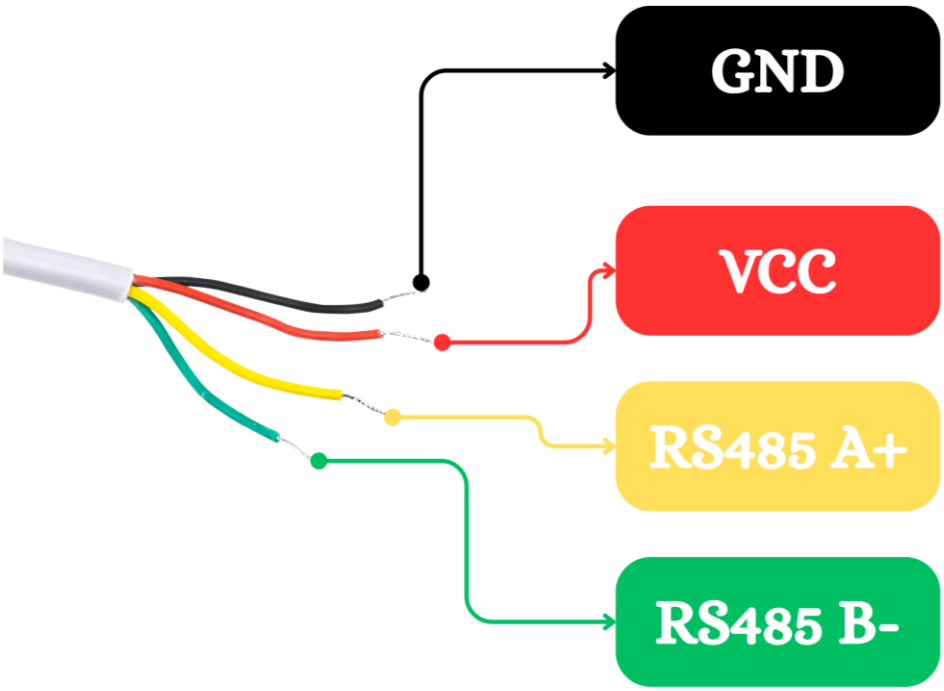
- HVAC control
- Building automation
- Warehouses and greenhouse
- Museums and archive
- Industrial process monitoring
- Weather stations and environmental studies

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% 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:** RS-485 signal
- **Communication Protocol:** Standard MODBUS RTU protocol
- **Baud Rate:** 9600 (default); configurable to 19200, 38400, 57600, 115200
- **Display Resolution:** Temperature: 0.1°C; Humidity: 0.1% RH

3. Hardware setup

3.1 Pinouts

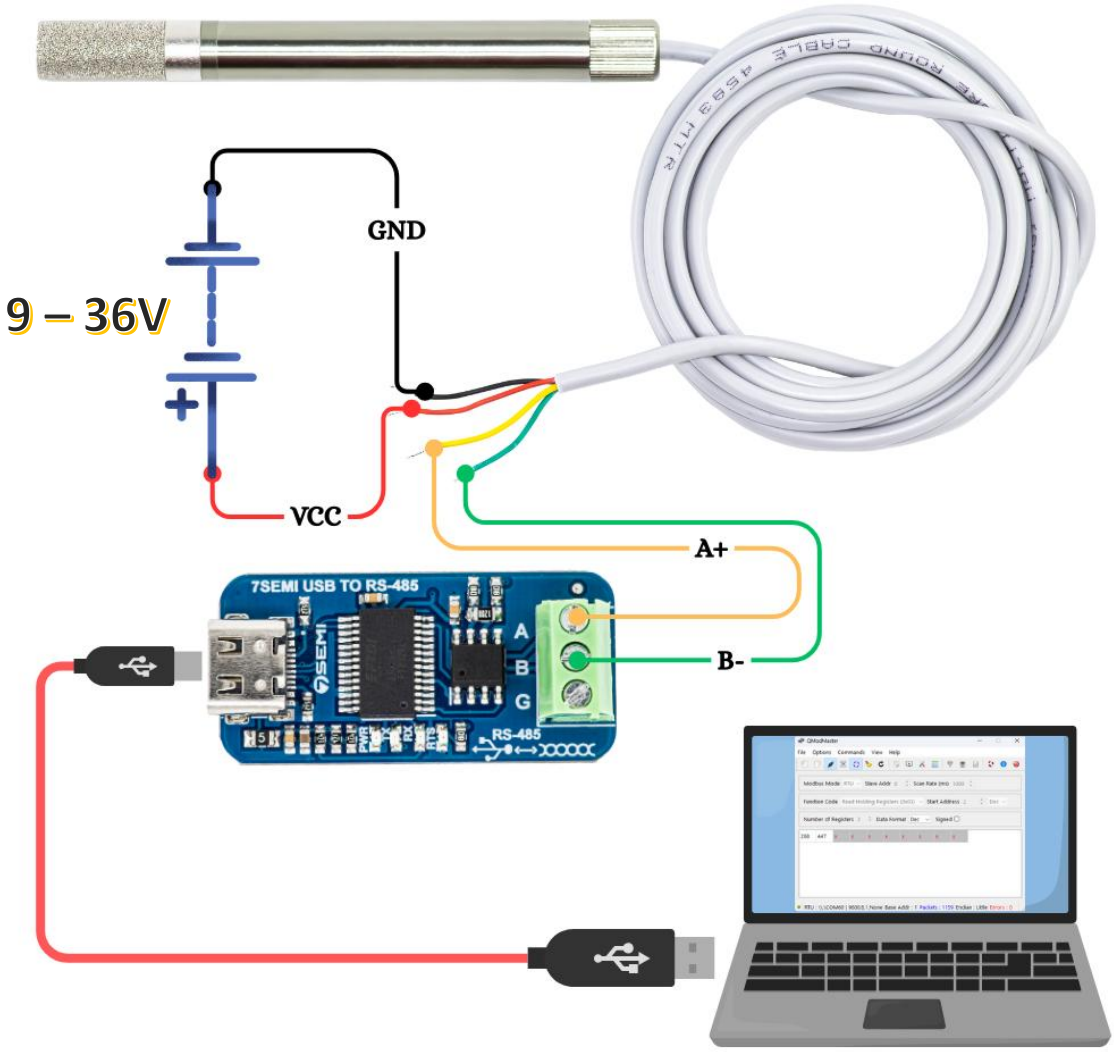


Wire Color	Function
Yellow	RS485 A+
Green	RS485 B−
Red	VCC
Black	GND

Connection Guidelines

- Power Supply: Connect the red wire to a 9–36V 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
0x04	Read Input Registers
0x06	Write Single Register
0x10	Write Multiple Registers

- The sensor supports **function code `0x03`**, **function code `0x04`**, **function code `0x06`** and **function code `0x10`**, which allows reading holding register values (such as temperature and humidity), Read Input Registers (Mirror of 0x03), Write Single Register and Write Multiple Registers

Register Address Mapping - Address quick reference (Base Addr = 1)

Register Address	Purpose	Start Address (Dec)
0x0000	Temperature (T) ×100 (int16)	1
0x0001	Humidity RH ×100 (uint16)	2
0x00E0	diag_mb_crc_err	225
0x00E1	diag_mb_except	226
0x00E4	Uptime (sec) high	229

0x00E5	Uptime (sec) low	230
0x00F0	Status (bit0=SHT OK, bit2=WARMUP)	241
0x0100	"Real" slave ID (read-only)	257
0x0101	New slave ID (write)	258
0x0200	Temp offset ×100 (int16)	513
0x0202	Measurement period (ms)	515
0x0206	RH offset ×100 (int16)	519
0x0210	Baud code (0..4)	529

- 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 100.
- Tip: for **signed** values (0x0000, 0x0200, 0x0206) tick **Data Format → Signed** in QModMaster.

➢ Testing Steps For QModMaster

1) Basic sensor read

1. **Function Code:** Read Holding Registers (0x03)
2. **Start Address: 1**
3. **Number of Registers: 2**
4. **Signed: ON**
    - Expect T and RH as integers ×100. E.g., `2579` → 25.79 °C; `6665` → 66.65 %RH.

2) Warm-up & status bits

1. Read **Start Address 241**, **Count 1** (0x00F0).
    - Immediately after boot, **bit2 (value 4)** is set (WARMUP). After ~5 s it clears.
    - **bit0 (value 1)** means SHT4x OK.

3) Uptime counter

1. Read **Start Address 229**, **Count 2** (0x00E4..0x00E5).
    - Combine: **seconds = (hi<<16) + lo**; it should increase each read.

4) Firmware version

1. Read **Start Address 255**, **Count 1** (0x00FE).
    - You should see **0x0100 = 256 decimal** for v**1.0**.

5) Real ID discovery (no matter which ID you queried)

1. Read **Start Address 257**, **Count 1** (0x0100).
    - Returns the device's **actual** slave ID.

6) Change measurement period (persisted)

1. **Function:** *Write Single Register (0x06)*
2. **Start Address: 515** (0x0202)
3. **Value:** e.g. **1000** (ms)
    - Observe new readings update every ~1 s. Power-cycle and re-read: value remains.

7) Apply offsets (persisted)

- **Temp offset** −0.50 °C: Write **Start 513** (0x0200) with **−50**.
    - If QModMaster won't accept "−50", enter **65536 − 50 = 65486**.
- **RH offset** +1.25 %RH: Write **Start 519** (0x0206) with **125**.
- Re-read T/RH (Step 1) and verify the shifts. Power-cycle → still there.

8) Baud-rate change with safe switch

1. Write **Start 529** (0x0210) with a **baud code**:

    ➢ 0=9600, 1=19200, 2=38400, 3=57600, 4=115200.

2. You'll get a normal reply at the **old** baud; then the device switches.
3. In QModMaster, **Options → Port**: change to the new baud; keep the same Slave Addr.
4. Re-read T/RH to confirm.

9) Change slave ID

1. In the main window of QmodMaster, change Function Code to Write Single Register (0x06) (sometimes shown as *Preset Single Register*).
2. Set Start Address to 258 (Dec).
3. Number of Registers: 1.

4. Data Format: Dec (so you can enter the ID as a decimal number).
5. In the value grid (first cell), type the new ID (valid range 1…247).
6. Click Read/Write button, here your slave ID change Successfully.
7. Switch to the new ID and confirm.
    - After the brief reboot, change Slave Address in QModMaster to the new ID.
    - To verify, set Function Code to Read Holding Registers (0x03).
    - Start Address = 1, Number of Registers = 2.

10) How to actually recover the ID

If you don't know the current ID, Turn off for 2 second and On it, for the next ~10 s, talk to Slave = 1 (your code's recovery alias).

1. Immediately in QModMaster:
    - Set Slave Addr = 1 (the recovery alias).
    - Function Code: 0x03 Read Holding Registers.
    - Start Address = 257, Number of Registers = 1.
    - Data format = Dec
2. Hit Read.
    - If wiring and baudrate are correct, the register at address 1 will return the *real* slave ID stored in flash.
    - Example: If it returns 7, then your device will answer only to ID = 7 after the 10-second recovery period ends.
3. After that, switch QModMaster's Slave Addr to that returned ID for normal operation.

11) Broadcast write (ID 0, no reply)

1. Set **Slave Addr = 0** (broadcast).
2. **Function:** *Write Multiple Registers (0x10)*
3. **Start Address: 515** (0x0202), **Number of Registers: 1**.
4. In the data cell type **2000** (ms) and send.
    - There is **no reply** by design; QModMaster may show a timeout—**that's expected**.
    - If you have multiple nodes, all should adopt the new period.

12) Read diagnostics

- **CRC errors**: Start **225** (0x00E0) = 0 =>OK
- **Exceptions**: Start **226** (0x00E1) = 0 =>OK
    - To bump **exceptions**, try writing to a read-only address (e.g., Write 0x06 to **Start 1**)—the firmware will reject it and increment the counter.

13) Input registers mirror (0x04)

- Repeat Step 1 but select **Function 0x04 (Read Input Registers)** and the same addresses.
    - You should get identical values (mirror of the read-only map).

14)Optional: Device Identification (MEI 0x2B/0x0E)

- QModMaster usually doesn't send 0x2B. If you have **Modbus Poll** or **modpoll**, you can query the MEI "Device ID" to read strings 7semi, **Industrial RS485 Temperature & Humidity Sensor Probe**, and version 1.0.

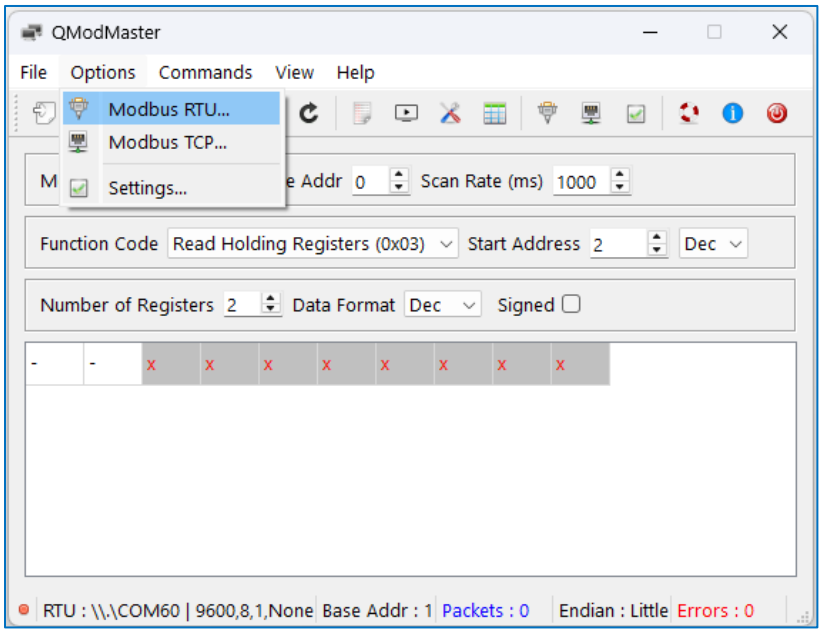
Baud Rate Setting

Decimal Code	Baud Rate
0	9600 bps
1	19200 bps
2	38400 bps
3	57600 bps
4	115200 bps

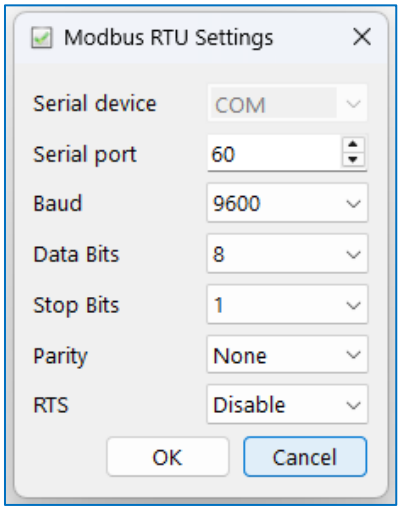
4.1 Modbus output

Follow the steps below for the software setup:-

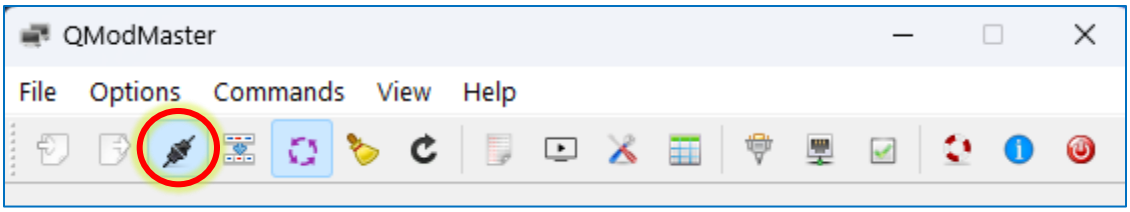
1. Open the software **(qModMaster-Win64-exe-0.5.3-beta)**
2. Go to *Options* → *Modbus RTU*



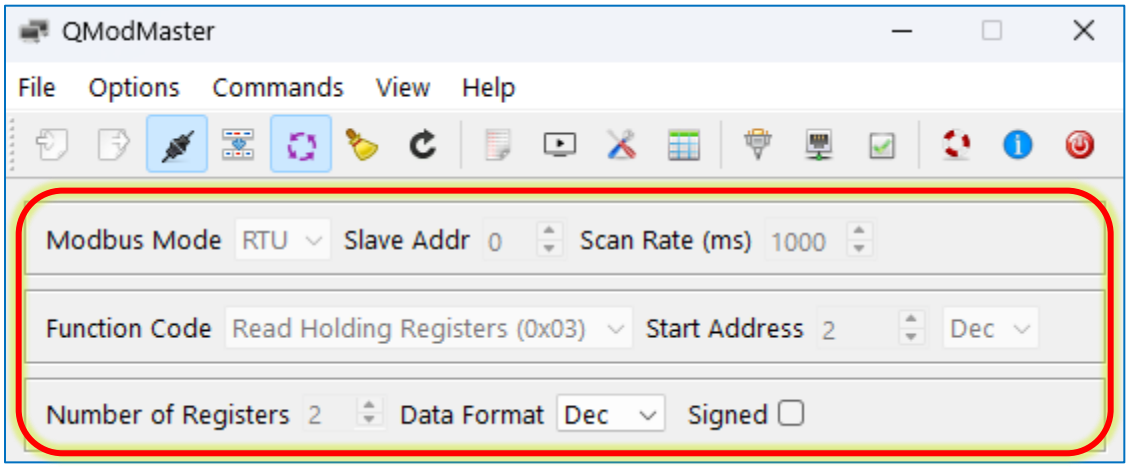
3. Check your COM port in device manager and select the other settings as per the image below. Click *OK*



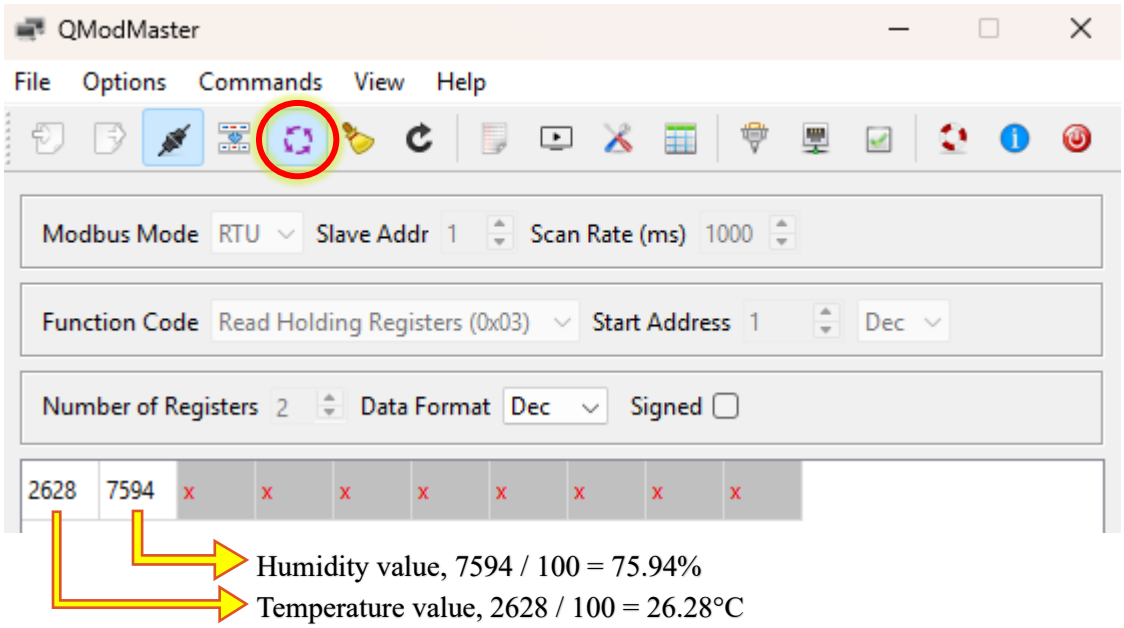
4. Connect your system to the correct COM port and click on this icon.



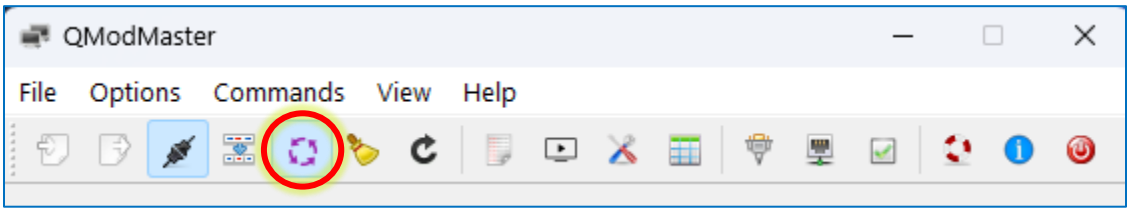
5. Also, make sure the following parameters are set properly.



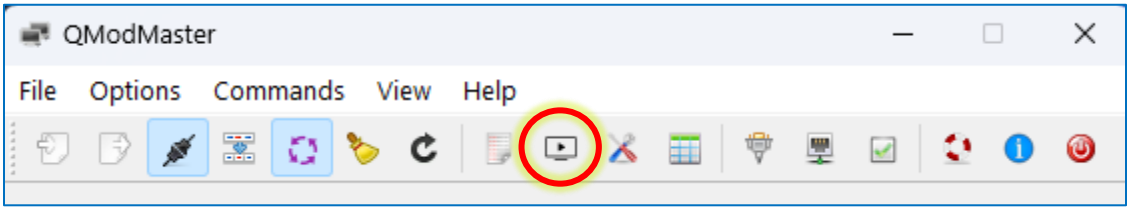
6. Now to check the data, click on this icon.



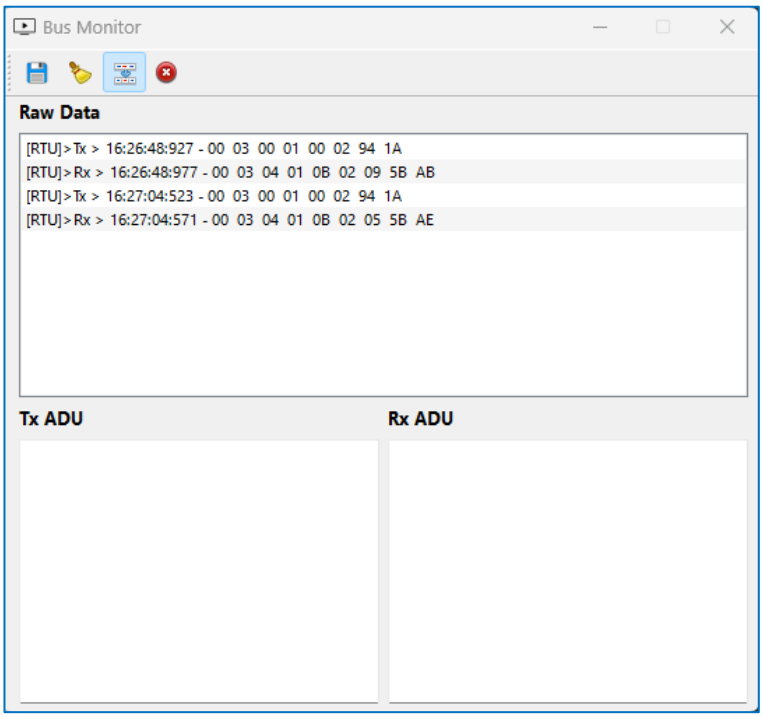
7. If you want to display the values continuously based upon the scan rate then click on this icon.

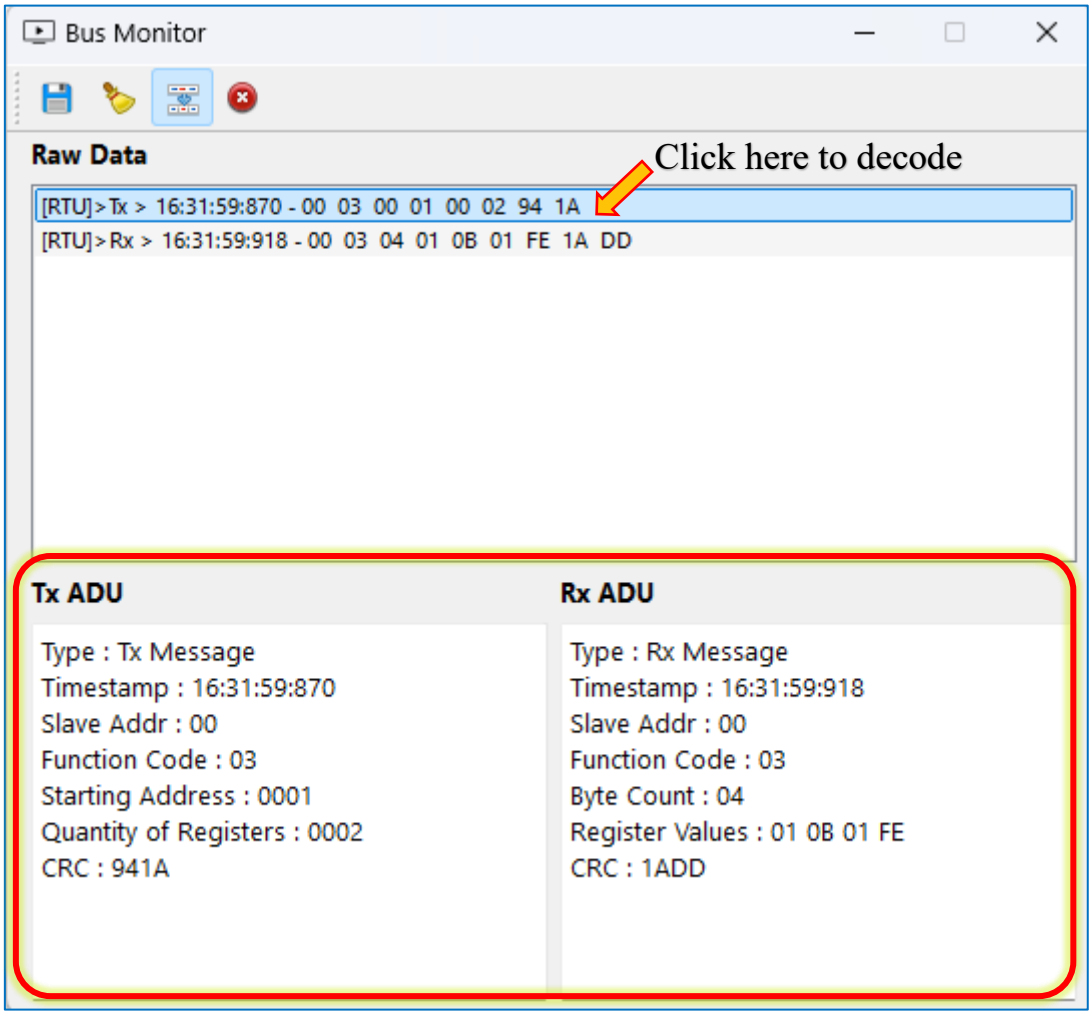


If you want to see the values in more detail you can click on this icon.



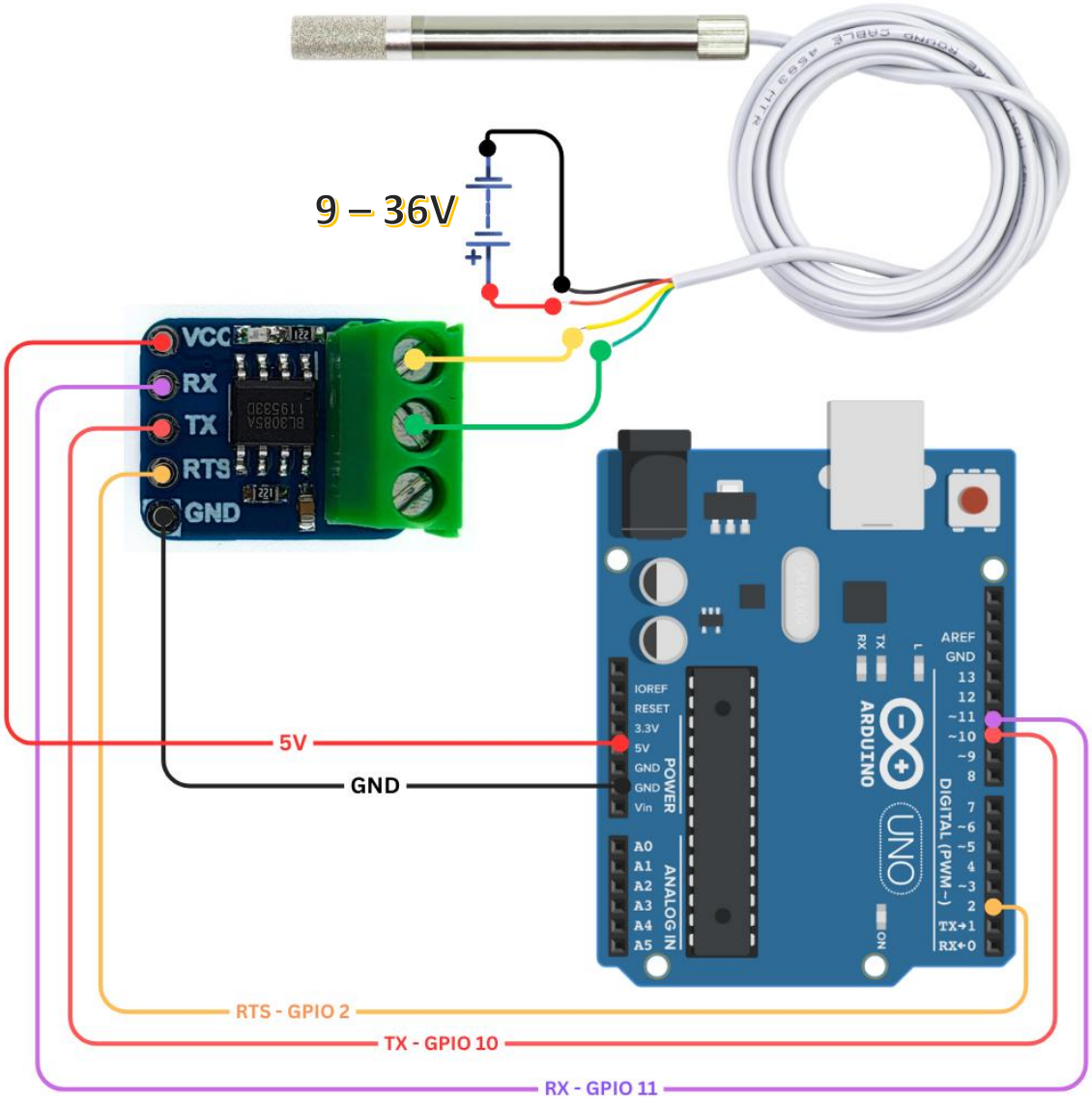
A new window will open to display the readings. To display data, follow step 6.




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

4.2 How to open the probe?

4.3 Connection and sample code for Arduino



How to Install 7Semi Arduino Library

- **Install the Arduino IDE** on your computer.
- Once installed, **open the Arduino IDE**.
- Go to **Library Manager** and search for **"7Semi-RS485-Temperature-Humidity-Probe-Arduino-Library."**
- **Install** the library from the search results.
- After installation, navigate to **File → Examples → Basic → Example Usage**, and open the **"Basic.ino"** example.
- **Upload** the example code to your Arduino board.
- Set the **baud rate to 115200** in the Serial Monitor.
- You should now **see the output** displayed on the Serial Monitor.

Sample code link for manual download:- "*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