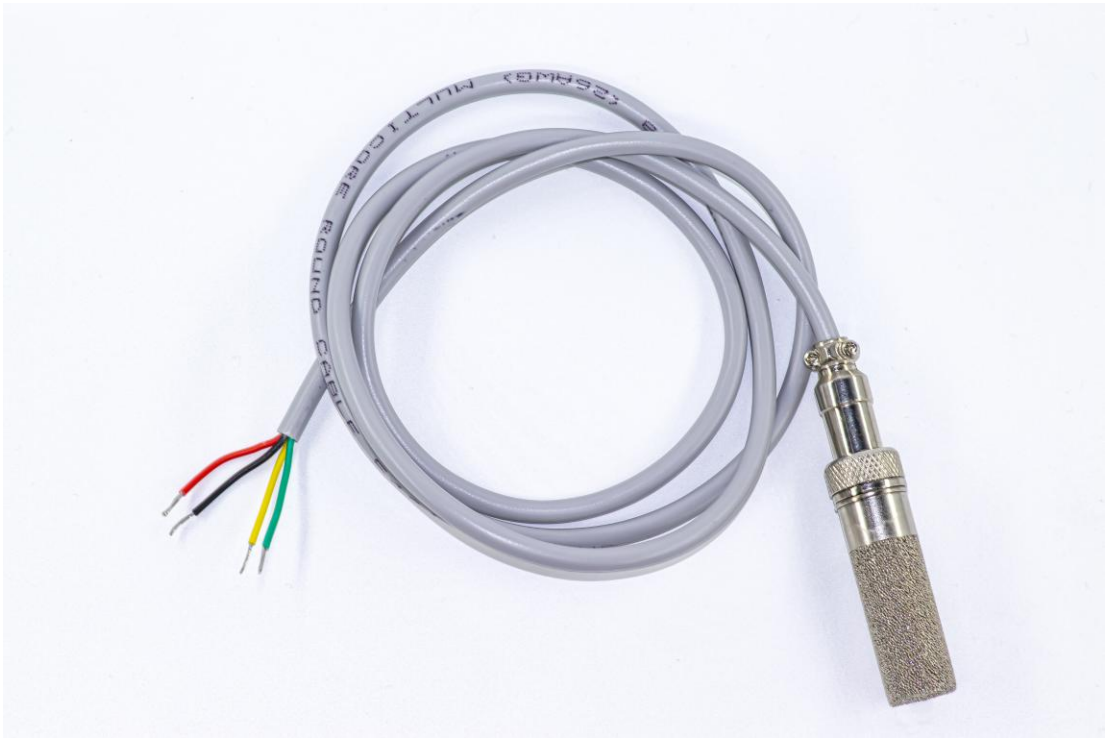
IST

IST

Industrial UART Humidity, Temperature and VOC Sensor Probe

Version 1.0



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

Table of Contents

1.Introduction

This all-in-one sensor is an excellent choice for developers needing accurate and low-power environmental measurements. Its compact design and versatile sensing capabilities make it suitable for everything from consumer wearables to industrial monitoring systems.

1.1 Features

- Digital temperature, humidity and air quality sensor probe with multicore cable.
- Compact and robust design with stainless steel body.
- Suitable for indoor and outdoor applications.
- Seamless compatibility with common development platforms.
- Ideal for HVAC, agriculture, weather stations, and smart home systems.
- Long-term stability for reliable performance over time.

2.Technical specification

This compact board is designed to provide the collection and processing of various environmental parameters, all in an ultra-low-power form factor. Its integrated design, featuring both data acquisition and on-board computation, makes it well-suited for IoT applications, and home automation projects. Flexible interfaces help ensure easy integration into existing systems, while the board's accuracy and reliability support demanding use cases.

The board outputs preformatted UART data frames containing decoded temperature, humidity, VOC values simplifying integration with external systems. Its energy-efficient design and compact form factor make it ideal for both portable and embedded applications.

Applications:-

- **Home Automation & control**
- **IoT Integration**
- **Weather Forecasting**
- **Educational Tools**
- **Indoor & outdoor monitoring systems**

This sensor board is the ideal solution for applications that demand high accuracy, low power consumption, and easy integration into existing systems.

2.1 General specifications

Operating Range:

- Temperature: -40°C to +125°C
- Humidity: 0 to 100% RH

Accuracy:

- Temperature: ±0.2°C
- Relative humidity: ±1.8% RH

Operating Voltage:

- VDD (supply voltage): 1.71 V to 3.6 V
- VDDIO (interface voltage): 1.2 V to 3.6 V

Communication protocol: UART

Baud rate for UART: 9600

2.2 Sensor protocol

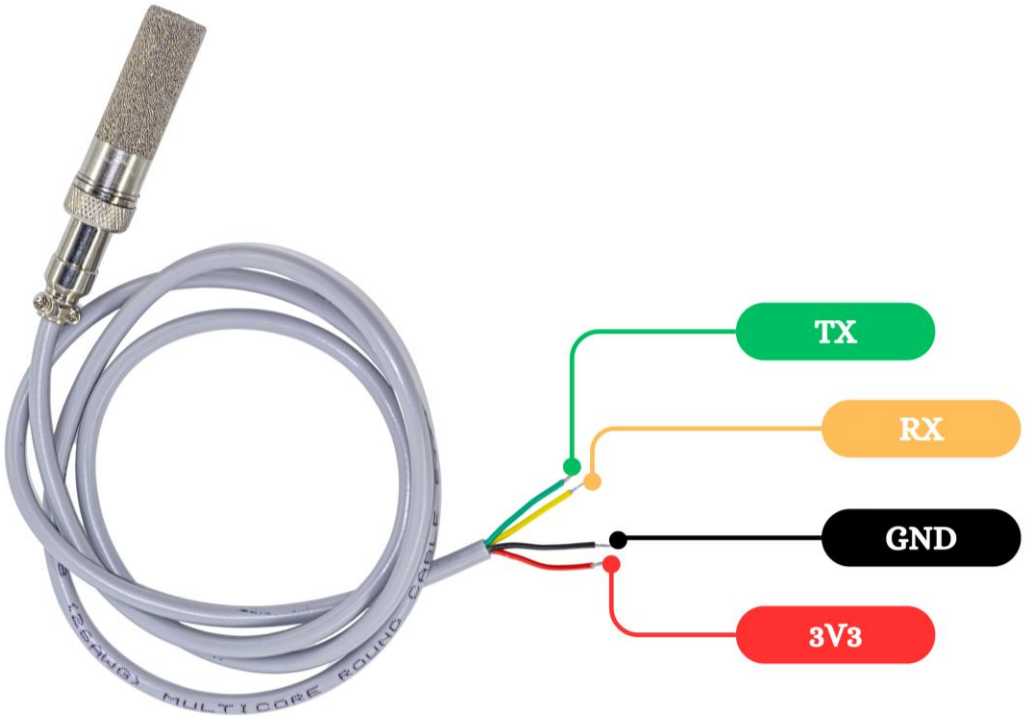
The sensor outputs the data frame using UART communication.

AA 6C D0 E3 41 9F ED 5E 42 25 00 00 00

Field	Bytes	Description
Start Byte	AA	Packet start marker
Temperature	6C D0 E3 41	4 bytes, Little Endian
Humidity	9F ED 5E 42	4 bytes, Little Endian
VOC index	25 00 00 00	4 bytes

3. Hardware setup

3.1 Pinouts



RX: UART receives pin, connect to the TX of the external device.

TX: UART transmits pin, connect to the RX of the external device.

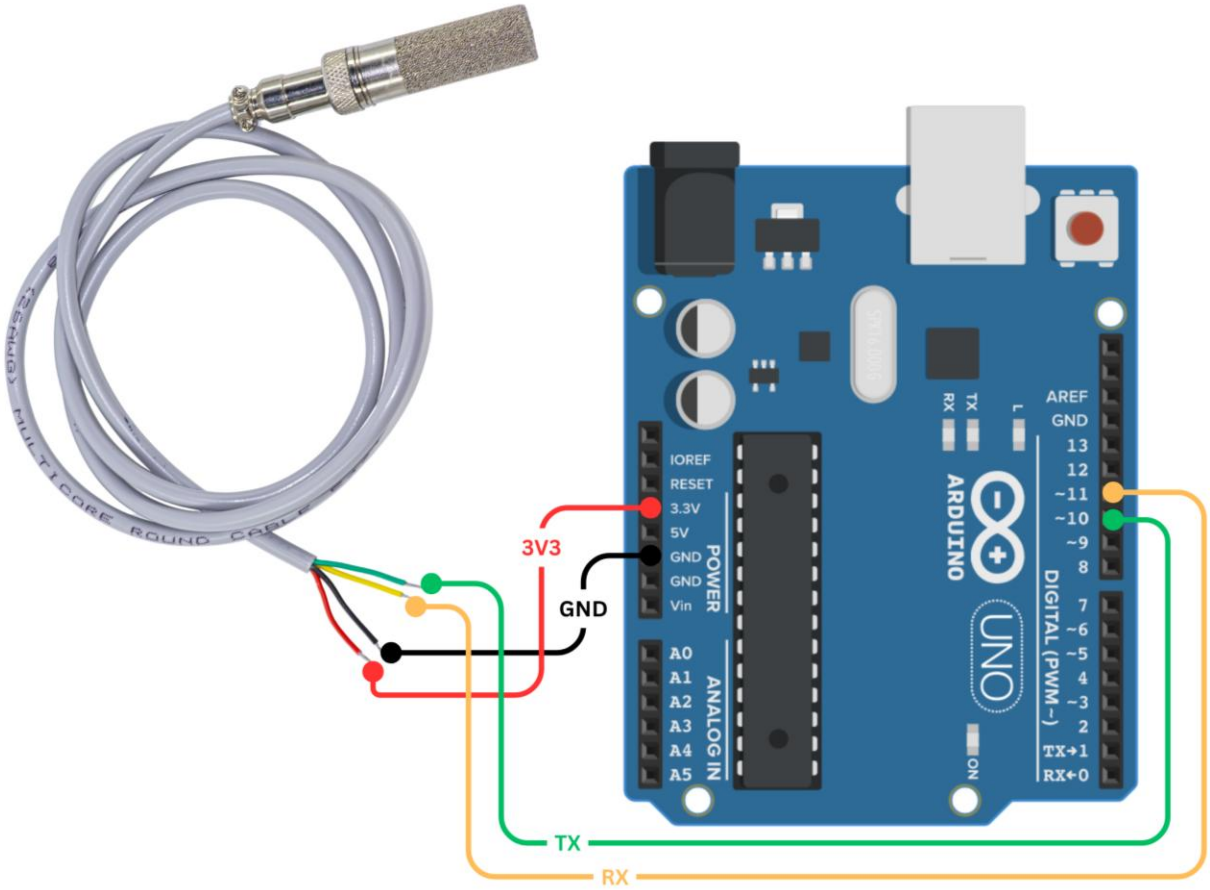
GND: Ground pin, connect to the system's ground.

3V3: Power input, connect to a 3.3V DC supply.

Connection Guidelines

- **Power Supply**: Ensure a clean and stable 3.3V DC power source is connected to the 3V3 pin, with GND tied to the power supply's ground.
- **UART Communication**: Connect the **RX** pin of the sensor board to the **TX** pin of the external device and the **TX** pin of the board to the **RX** pin of the external device.
- Properly match baud rates and settings (e.g., stop bits, parity) for UART communication.

3.2 Hardware connection



4.Sample code and explanation

4.1 Sample code

```
#include <SoftwareSerial.h>
const int frameSize = 13;
uint8_t buffer[frameSize];

// Define software serial on pins 10 (RX), 11 (TX)
SoftwareSerial mySerial(10, 11); // RX, TX

void setup() {
    mySerial.begin(9600); // Initialize Software Serial Communication
    Serial.begin(9600); // Initialize Serial Communication
}

void loop() {
    if (mySerial.available() >= frameSize)
    {
        mySerial.readBytes(buffer, frameSize);

        // Validate start byte
        if (buffer[0] != 0xAA)
        {
            Serial.println("Invalid frame start");
            return;
        }
```

```
        // Extract values
        int32_t temp_int, hum_int, voc_index;
        memcpy(&temp_int, &buffer[1], sizeof(int32_t));
        memcpy(&hum_int, &buffer[5], sizeof(int32_t));
        memcpy(&voc_index, &buffer[9], sizeof(int32_t));

        // Convert to float
        float temperature = *(float*)&temp_int;
        float humidity = *(float*)&hum_int;
        uint32_t voc= *(uint32_t*)&voc_index;

        // Ensure VOC index is positive
        if (voc_index < 0) voc_index = 0;

        // Print decoded values
        Serial.print("Temperature: ");
        Serial.print(temperature);
        Serial.print(" C, Humidity: ");
        Serial.print(humidity);
        Serial.print(" %, VOC Index: ");
        Serial.println(voc);
    }
}
```

4.2 Code explanation

```
#include <SoftwareSerial.h>
```

Includes the SoftwareSerial library, which allows using different pins for serial communication (instead of just pins 0 and 1).

```
const int frameSize = 13;
uint8_t buffer[frameSize];
```

frameSize: Total number of bytes expected in one data frame. Set to 13 bytes.

buffer: Array to temporarily store incoming serial data.

```
void setup() {
    mySerial.begin(9600);
    Serial.begin(9600); // Initialize Serial Communication
}
```

Initializes both the SoftwareSerial and the Serial Monitor at 9600 baud.

```
void loop() {
    if (Serial.available() >= frameSize)
```

Serial.available(): Checks how many bytes are available in the serial buffer.

If 13 or more bytes are available, we proceed to read them into the buffer.

```
Serial.readBytes(buffer, frameSize);
```

Serial.readBytes: Reads exactly frameSize bytes (13) from the serial buffer into buffer.

```
if (buffer[0] != 0xAA)
{
    Serial.println("Invalid frame start");
    return;
}
```

The first byte is checked for a start marker (0xAA).

If the first byte is not 0xAA, the function returns early, indicating an invalid frame.

```
int32_t temp_int, hum_int, voc_index;
memcpy(&temp_int, &buffer[1], sizeof(int32_t));
memcpy(&hum_int, &buffer[5], sizeof(int32_t));
memcpy(&voc_index, &buffer[9], sizeof(int32_t));
```

Three 4-byte values are extracted using memcpy:

buffer[1] to buffer[4]: Temperature (4 bytes)

buffer[5] to buffer[8]: Humidity (4 bytes)

buffer[9] to buffer[12]: VOC Index (4 bytes)

```
float temperature = *(float*)&temp_int;
float humidity = *(float*)&hum_int;
uint32_t voc = *(uint32_t*)&voc_index;
```

These lines reinterpret the 4-byte int32_t values as:

temperature: float version of temp_int

humidity: float version of hum_int

voc: unsigned version of voc_index

```
if (voc_index < 0) voc_index = 0;
```

Ensures voc_index is not negative.

This step seems slightly redundant since voc_index is cast to uint32_t anyway, but might be retained for logical clarity or debugging.

```
Serial.print("Temperature: ");
Serial.print(temperature);
Serial.print(" C, Humidity: ");
Serial.print(humidity);
Serial.print(" %, VOC Index: ");
Serial.println(voc);
```

Prints the decoded values in a human-readable format on the serial monitor.

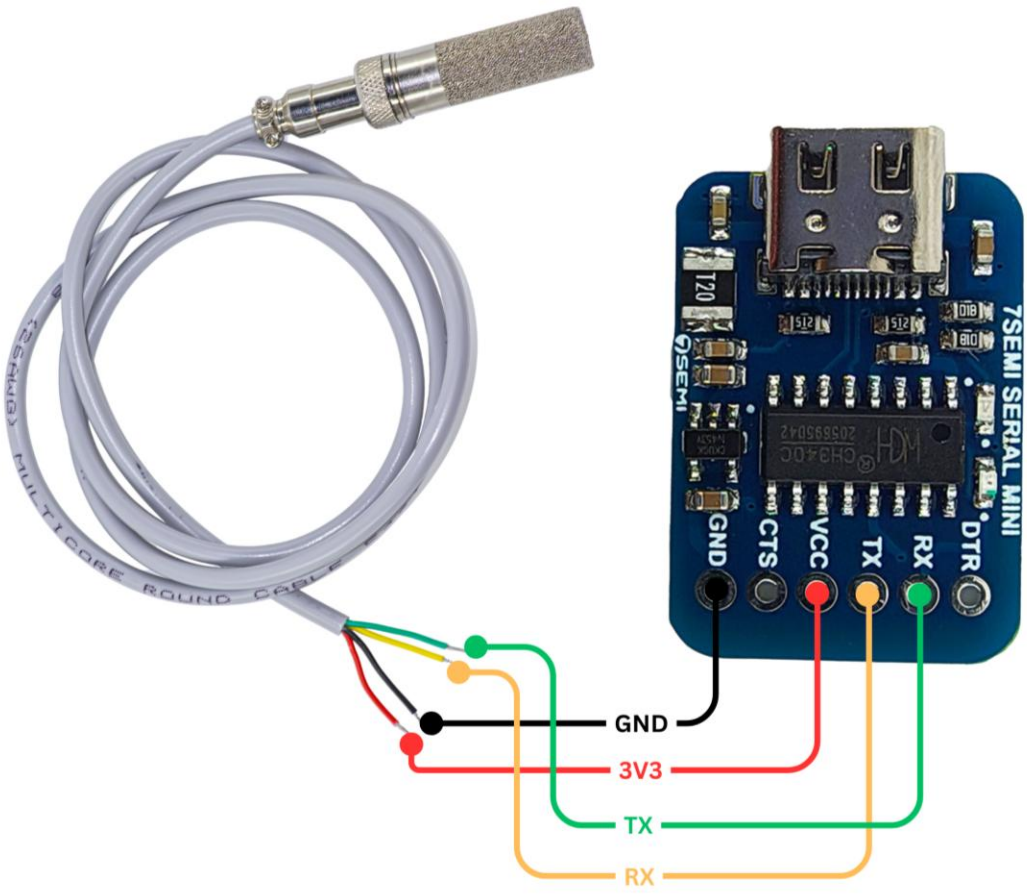
4.3 Serial output

```
Temperature: 28.54 C, Humidity: 54.43 %, VOC Index: 0
Temperature: 28.55 C, Humidity: 54.41 %, VOC Index: 1
Temperature: 28.59 C, Humidity: 54.34 %, VOC Index: 1
Temperature: 28.61 C, Humidity: 54.33 %, VOC Index: 3
Temperature: 28.61 C, Humidity: 54.31 %, VOC Index: 6
Temperature: 28.61 C, Humidity: 54.29 %, VOC Index: 9
Temperature: 28.64 C, Humidity: 54.29 %, VOC Index: 13
Temperature: 28.64 C, Humidity: 54.27 %, VOC Index: 17
Temperature: 28.67 C, Humidity: 54.24 %, VOC Index: 20
Temperature: 28.65 C, Humidity: 54.19 %, VOC Index: 24
Temperature: 28.68 C, Humidity: 54.20 %, VOC Index: 28
Temperature: 28.69 C, Humidity: 54.21 %, VOC Index: 31
Temperature: 28.71 C, Humidity: 54.35 %, VOC Index: 35
Temperature: 28.70 C, Humidity: 54.46 %, VOC Index: 38
Temperature: 28.73 C, Humidity: 54.47 %, VOC Index: 41
Temperature: 28.73 C, Humidity: 54.40 %, VOC Index: 44
Temperature: 28.75 C, Humidity: 54.36 %, VOC Index: 47
Temperature: 28.75 C, Humidity: 54.31 %, VOC Index: 50
Temperature: 28.77 C, Humidity: 54.26 %, VOC Index: 52
Temperature: 28.77 C, Humidity: 54.18 %, VOC Index: 55
Temperature: 28.79 C, Humidity: 54.11 %, VOC Index: 57
Temperature: 28.79 C, Humidity: 54.00 %, VOC Index: 59
Temperature: 28.82 C, Humidity: 53.92 %, VOC Index: 61
Temperature: 28.82 C, Humidity: 53.86 %, VOC Index: 63
Temperature: 28.83 C, Humidity: 53.82 %, VOC Index: 64
Temperature: 28.81 C, Humidity: 53.74 %, VOC Index: 66
Temperature: 28.86 C, Humidity: 53.73 %, VOC Index: 68
Temperature: 28.82 C, Humidity: 53.69 %, VOC Index: 69
Temperature: 28.84 C, Humidity: 53.71 %, VOC Index: 70
```

4.4 Sample code link

Sample output Image of USB to TTL

```
AA 6C D0 E3 41 9F ED 5E 42 25 00 00 00
```



Sample code link:- [ES-50103-U4040]

5.Mechanical specification