

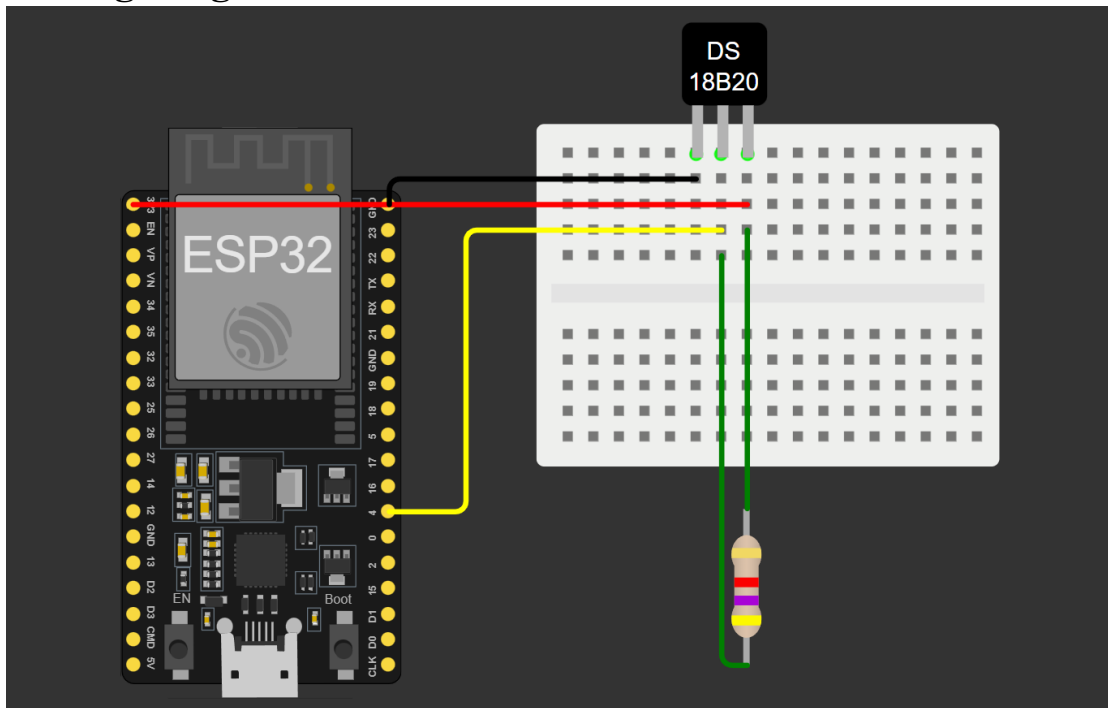
Documentation

SETUP-

- **Requirement:**

1. ESP32 Microcontroller
2. DS18B20 Temperature Sensor for One-Wire Communication
3. Resistor (4.7k Ω)
4. BreadBoard
5. Power Supply

- **Wiring Diagram:**



1. The Data pin of the sensor is connected to GPIO 4 on the ESP32.
2. The Vcc pin of the sensor is connected to 3.3V on the ESP32.
3. The Ground pin of the sensor is connected to Ground on the ESP32.
4. A 4.7k Ω pull-up resistor is placed between the Data pin of the DS18B20 and the VCC (3.3V) pin of the ESP32.

- **Library Used:**

1. `WiFi.h` : For connecting the ESP32 to a WiFi network.
2. `PubSubClient.h` : For MQTT communication.
3. `OneWire.h` : For One-Wire communication protocol.
4. `DallasTemperature.h` : Provides functions for interfacing with the DS18B20.

- **Code Comments:**

1. Library used- including all the libraries required.

```
#include <WiFi.h> // Library to enable WiFi connectivity on ESP32
#include <PubSubClient.h> // Library for MQTT communication
#include <OneWire.h> // Library to handle One-Wire communication
#include <DallasTemperature.h> // Library to interface with DS18B20
sensors
```

2. Wifi Credentials- creating variables for ssid and password for Connecting to Wifi.

```
// Wifi Credentials
const char* ssid = "****"; // your ssid
const char* password = "****"; // your password
```

3. Thingsboard Credentials- creating variables for Thingsboard URL, port and Access Token for establishing MQTT communication.

```
// MQTT broker address and credentials
const char* mqtt_server = "demo.thingsboard.io"; //
int mqtt_port = 1883; // default port for MQTT
const char* access_token = "*****"; // Access Token for your
ThingsBoard Device
```

4. One Wire Communication- Creating a one-wire object for communication with the DS18B20 sensor and passing the object to the DallasTemperature Library.

```
// GPIO where the DS18B20 is connected
#define ONE_WIRE_BUS 4
OneWire oneWire(ONE_WIRE_BUS); // Create a OneWire object to
communicate with the DS18B20 sensor
DallasTemperature sensors(&oneWire); // Pass the oneWire reference to
DallasTemperature library
```

5. Initializing the WiFiClient for connection to the network and PubSubClient for MQTT Communication as well as publishing data to Thingsboard.

```
WiFiClient Client; // WiFiClient object to handle the connection to
the network
PubSubClient client(Client); // PubSubClient object for MQTT
communication
```

6. Wifi Connection- This is a function for establishing Wifi Connectivity

```
void WifiConnect() {
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password); // Connect to Wifi
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected");
}
```

7. Thingsboard Connection- This is a function for establishing connection with Thingsboard using the Access Token for MQTT communication.

```
void MQTTconnect() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        if (client.connect("ESP32", access_token, NULL)) {
            Serial.println("connected");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println("try again in 5 seconds");
            delay(5000);
        }
    }
}
```

8. Setup Function-

- Setting the baud rate and
- Calling function for wifi connectivity
- Configure the client to connect to MQTT broker(demo.thingsboard.io)
- Initialize the DallasTemperature object sensors, which is required to begin communication with DS18B20 sensor on One-Wire.

```
void setup() {  
    Serial.begin(115200);  
    WifiConnect();  
    client.setServer(mqtt_server, mqtt_port);  
    sensors.begin();  
}
```

9. Loop Function-

- If client is not connected then ,calling function 'MQTTconnect()' to establish connection to Thingsboard for MQTT communication.
- Requesting Temperature reading from the sensor.
- Retrieving the Temperature reading in Celsius and string it as Float.
- Printing the Temperature reading on Serial Monitor.
- Creating a json payload and publishing it to Thingsboard

```
void loop() {  
    if (!client.connected()) {  
        MQTTconnect(); // Establish connection to Thingsboard for MQTT  
communication  
    }  
    client.loop();  
    sensors.requestTemperatures(); // Request Temperature reading  
from the sensor  
    float temperature = sensors.getTempCByIndex(0); // Retrieve  
Temperature reading in Celsius  
    Serial.print("Temperature: ");  
    Serial.print(temperature);  
    Serial.println(" °C");  
  
    //Create a JSON payload  
    String message = "{";  
    message += "\"temperature\":";  
    message += temperature;  
    message += "}";  
  
    // Publish temperature to ThingsBoard  
    client.publish("v1/devices/me/telemetry", message.c_str());  
    delay(30000); // Delay between readings  
}
```

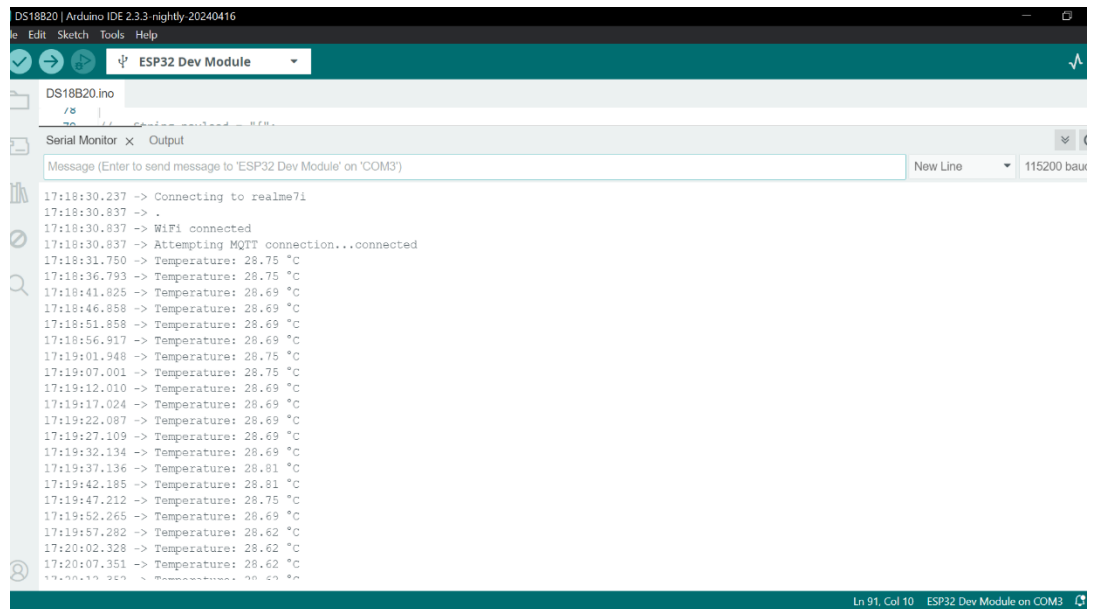
- **Approach:**

This project involves-

1. Connecting the ESP32, DS18B20 Sensor and 4.7k Ω resistor over the BreadBoard as per the Wiring Diagram.
2. Installing all the required libraries.
3. Setting up a Wifi Connection.
4. Connecting to ThingsBoard for MQTT communication using the Access Token for the particular Device.
5. Retrieving the Temperature Data from the sensor.
6. Publishing the retrieved data as json over to Thingsboard.

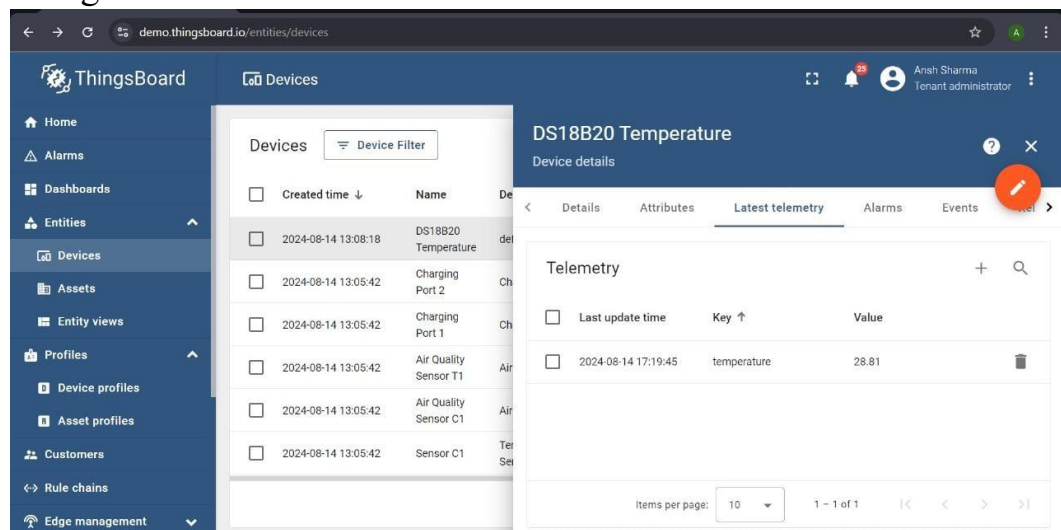
- **Output:**

1. Serial Monitor-



```
DS18B20 | Arduino IDE 2.3.3-nightly-20240416
File Edit Sketch Tools Help
ESP32 Dev Module
DS18B20.ino
Serial Monitor x Output
Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')
New Line 115200 baud
17:18:30.237 -> Connecting to realme71
17:18:30.837 -> .
17:18:30.837 -> WiFi connected
17:18:30.837 -> Attempting MQTT connection...connected
17:18:31.750 -> Temperature: 28.75 °C
17:18:36.793 -> Temperature: 28.75 °C
17:18:41.825 -> Temperature: 28.69 °C
17:18:46.858 -> Temperature: 28.69 °C
17:18:51.858 -> Temperature: 28.69 °C
17:18:56.917 -> Temperature: 28.69 °C
17:19:01.948 -> Temperature: 28.75 °C
17:19:07.001 -> Temperature: 28.75 °C
17:19:12.010 -> Temperature: 28.69 °C
17:19:17.024 -> Temperature: 28.69 °C
17:19:22.087 -> Temperature: 28.69 °C
17:19:27.109 -> Temperature: 28.69 °C
17:19:32.134 -> Temperature: 28.69 °C
17:19:37.136 -> Temperature: 28.81 °C
17:19:42.185 -> Temperature: 28.81 °C
17:19:47.212 -> Temperature: 28.75 °C
17:19:52.265 -> Temperature: 28.69 °C
17:19:57.282 -> Temperature: 28.62 °C
17:20:02.328 -> Temperature: 28.62 °C
17:20:07.351 -> Temperature: 28.62 °C
17:20:12.359 -> Temperature: 28.62 °C
Ln 91, Col 10 ESP32 Dev Module on COM3
```

2. Thingsboard-



ThingsBoard

demo.thingsboard.io/entities/devices

Ansh Sharma
Tenant administrator

DS18B20 Temperature
Device details

Details Attributes Latest telemetry Alarms Events

Telemetry

| Last update time | Key | Value |
|---------------------|-------------|-------|
| 2024-08-14 17:19:45 | temperature | 28.81 |

Items per page: 10 1 - 1 of 1

3. Thingsboard Dashboard-

