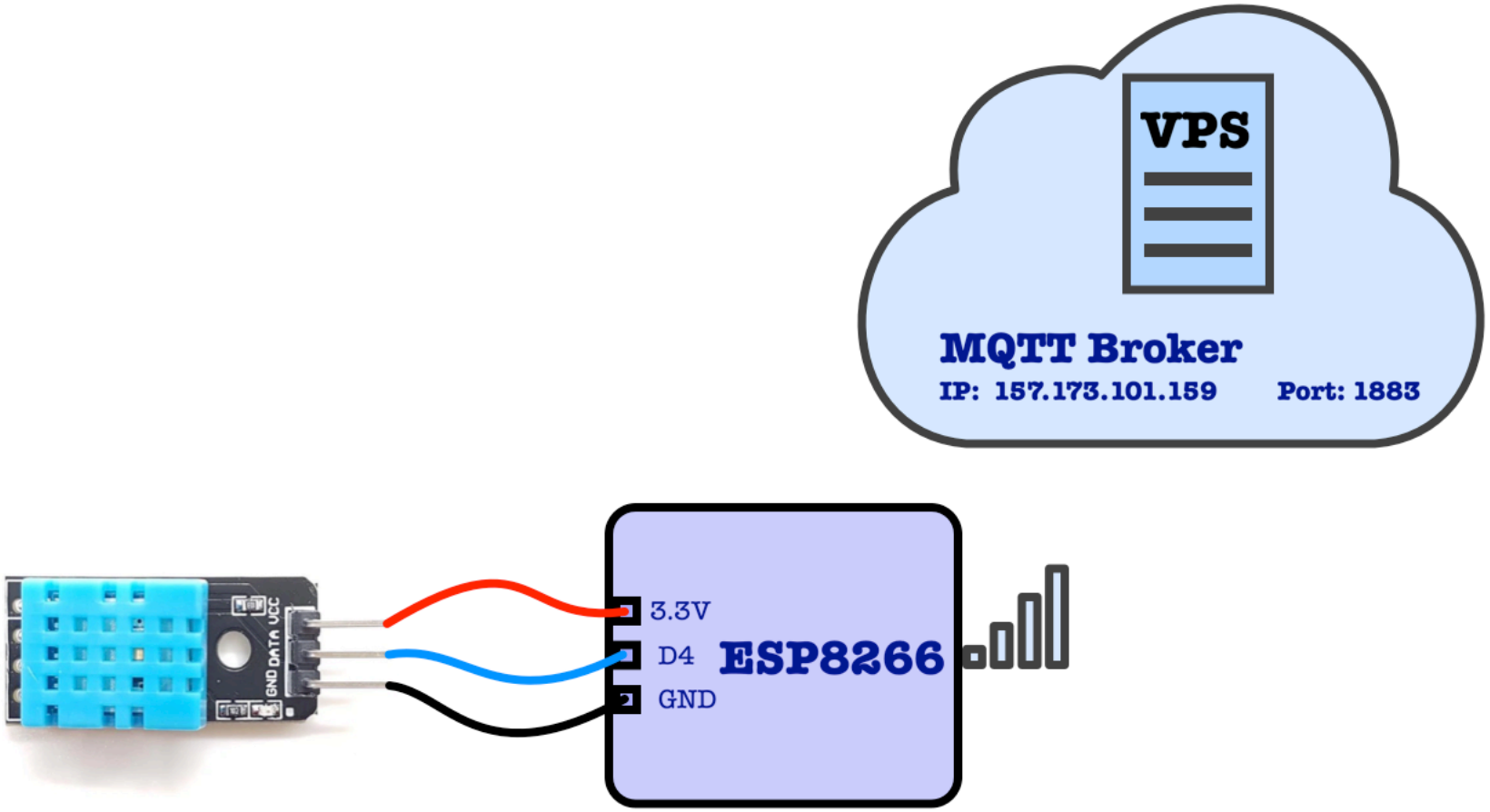
Weather Station Over MQTT



Requirements

1️⃣ Hardware Components

- DHT11 (for temperature & humidity)
- ESP8266 (NodeMCU) (for WiFi & MQTT)
- Jumper Wires (for connections)

2️⃣ Software

- Arduino IDE (with ESP8266 & DHT sensor libraries)
- Mosquitto MQTT Broker (We have a VPS at 157.173.101.159)
- WiFi Connection

Code | Function 1

```
21  // Function to Connect to WiFi
22  void setup_wifi() {
23      Serial.print("Connecting to WiFi...");
24      WiFi.begin(ssid, password);
25      while (WiFi.status() != WL_CONNECTED) {
26          delay(500);
27          Serial.print(".");
28      }
29      Serial.println("\nConnected to WiFi!");
30  }
```

Code | Function 2

```
32  // Function to Connect to MQTT Broker
33  void reconnect_mqtt() {
34      while (!client.connected()) {
35          Serial.print("Connecting to MQTT...");
36          if (client.connect("ESP8266")) {
37              Serial.println("Connected to MQTT!");
38          } else {
39              Serial.print("Failed, retrying in 5 seconds...");
40              delay(5000);
41          }
42      }
43  }
```

Code | Function 3

```
45  // Function to Read Sensor Data
46  void read_sensor(float &temperature, float &humidity) {
47      temperature = dht.readTemperature();
48      humidity = dht.readHumidity();
49
50      if (isnan(temperature) || isnan(humidity)) {
51          Serial.println("Failed to read from DHT sensor!");
52          temperature = -1; // Invalid value
53          humidity = -1;      // Invalid value
54      }
55  }
```

Code | Function 4

```
58  void publish_sensor_data(float temperature, float humidity) {
59      if (temperature == -1 || humidity == -1) {
60          return;  // Do not publish invalid values
61      }
62
63      String temp_payload = String(temperature);
64      String hum_payload = String(humidity);
65
66      client.publish("/work_group_01/room_temp/temperature", temp_payload.c_str());
67      client.publish("/work_group_01/room_temp/humidity", hum_payload.c_str());
68
69      Serial.println("Published: Temperature = " + temp_payload + "°C");
70      Serial.println("Published: Humidity = " + hum_payload + "%");
71  }
```

Code | setup() and loop(), native functions of the Arduino Framework

```
73  void setup() {    //run once at the beginning
74      Serial.begin(115200);
75      setup_wifi(); //call Function1
76      client.setServer(mqtt_server, 1883);
77      dht.begin();
78  }
79
80  void loop() {    //run forever
81      if (!client.connected()) {
82          reconnect_mqtt(); //call Function2
83      }
84      client.loop();
85
86      float temperature, humidity;
87      read_sensor(temperature, humidity); //call Function3
88      publish_sensor_data(temperature, humidity); //call Function4
89
90      delay(5000);  // Send data every 5 seconds
91  }
```

Viewing real-time sensor data from MQTT

```
gabriel@MacBookAir-2 ~ % mosquitto_sub -h 157.173.101.159 -t "/work_group_01/room_temp/#" -v
/work_group_01/room_temp/temperature 25.80
/work_group_01/room_temp/humidity 60.00
/work_group_01/room_temp/temperature 25.80
/work_group_01/room_temp/humidity 60.00
/work_group_01/room_temp/temperature 25.80
/work_group_01/room_temp/humidity 60.00
/work_group_01/room_temp/temperature 25.80
/work_group_01/room_temp/humidity 60.00
/work_group_01/room_temp/temperature 25.80
/work_group_01/room_temp/humidity 60.00
/work_group_01/room_temp/temperature 25.80
/work_group_01/room_temp/humidity 60.00
/work_group_01/room_temp/temperature 25.80
/work_group_01/room_temp/humidity 60.00
```

🔶 Installing Mosquitto (Depends on Your OS)

📌 On Linux (Ubuntu/Debian)

```sh
sudo apt update
sudo apt install -y mosquitto mosquitto-clients
```

📌 On macOS (Homebrew)

```sh
brew install mosquitto
```

🔶 After installing, **start Mosquitto as a background service:**

```sh
brew services start mosquitto
```

📌 On Windows

1️⃣ Download Mosquitto from the official website:
👉 https://mosquitto.org/download/
2️⃣ Install it and ensure the `mosquitto` binaries are in your system path.