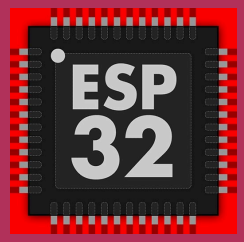
scriptWORKZ

Scriptworkz ent
30 Nov 2022

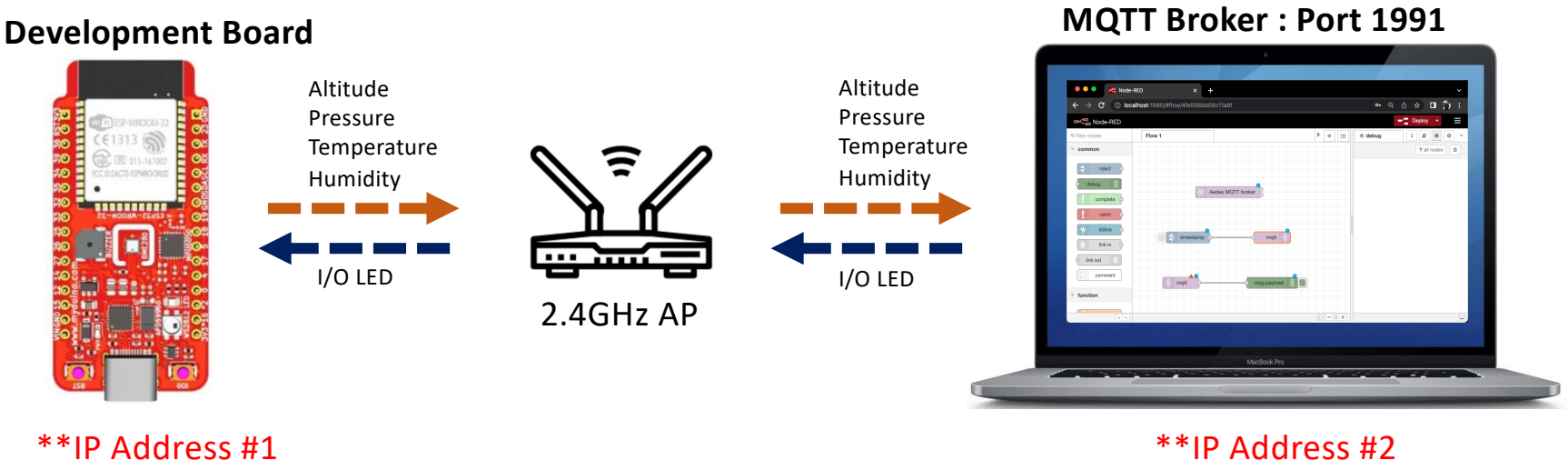
Built Your Own IoT Plaftorm

DIY Weather Station Dashboard with ESP34, Node-RED, MQTT & BME280



DIY Weather Station

- This activity will focus on developing a mini/portable weather station using ESP32 & BME280 as the hardware, and Node-RED & MQTT as the service to publish the information.
- The system not sending information such as altitude, barometer pressure, temperature & humidity, it also able to switch on and off the LED on the development board based on instruction send from the Node-RED's dashboard.
- The information will be displayed on the Node-RED's dashboard.



**IP Address #1

**IP Address #2

**IP addresses will be released once the devices connected to the 2.4GHz AP.

i. System Requirement

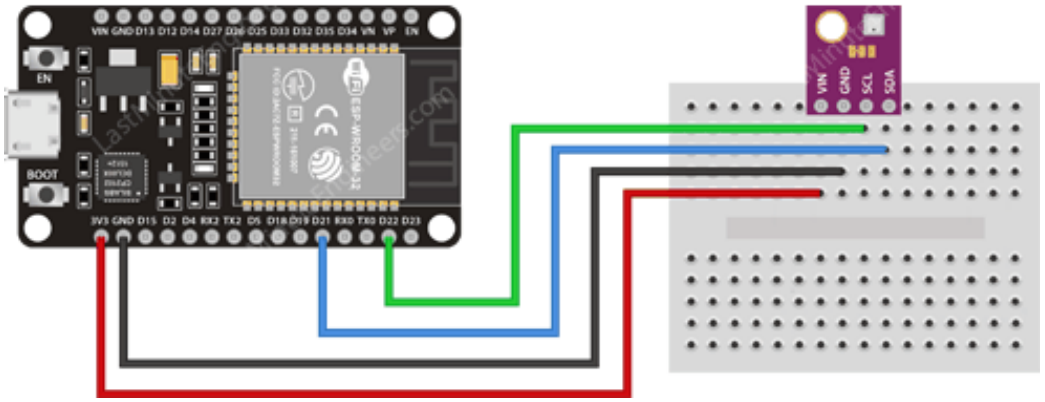
- 1 x ESP32 board with BME280 sensor, breadboard & jumpers

  OR
- 1 x ESP32 IoT Development Board integrated with BME280 sensor
- PC/Laptop with any OS that has been installed Node-RED
- Launch Node-RED service by typing **node-red** at command prompt or terminal.
- Open any browser and type **localhost:1880** at address bar launch the Node-RED editor. You may use your **machine's IP address** instead of **localhost:1880** to surf the Node-RED editor (don't forget the port of **youripaddress:1880** at the end of the IP address). This link will teach you in finding your PC/Laptop IP address.

ii. Wiring Diagram

- Do the following steps if you are using ESP32 with BME280, breadboard and jumpers, or **skip if you are using ESP32 with built-in BME280 sensor.**

Last Minute ENGINEERS.com



- Temperature: -40°C to 85°C (±1.0°C accuracy)
- Humidity: 0 to 100% RH (±3% accuracy)
- Pressure: 300hPa to 1100hPa (±1hPa accuracy)
- Altitude: 0 to 30,000ft (±1 meter accuracy)

Wiring BME280 Sensor to ESP32

Connections are fairly simple. Start by connecting **VIN** pin to the 3.3V output on the ESP32 and connect **GND** to ground.

Next, Connect the **SCL** pin to the I2C clock **D22** pin on your ESP32 and connect the **SDA** pin to the I2C data **D21** pin on your ESP32.

https://lastminuteengineers.com/bme280-esp32-weather-station/

iii. Programming your Board

- Download the following file sketch_nov25b_bme_mqtt.ino.
- Change to your WiFi's ssid and password at line #20 and #21.

```
20  const char* ssid = "YOUR SSID";
21  const char* password = "YOUR SSID PASSWORD";
```

- Key in the Node-RED's IP address too at line #24.

```
24  const char* mqtt_server = "MQTT Broker IP ADDRESS";
```

- Then upload the edited sketch to your ESP32 board.
- In this sketch, Adafruit BME280 – environmental sensor library has been included. Refer this link for other projects.
- Adafruit_BME280 elements:

```
7   // include the Adafruit_BME280 header file.
8   #include <Adafruit_BME280.h>
```

```
56  // Initialize BME280 sensor and if the connection is not successful,
57  // print the failed status to the Serial Monitor.
58  if (!bme.begin()){
59    Serial.println("Failed to find Hibiscus Sense BME280 chip");
60  }
```

iii. Programming your Board

- The sensors reading in ESP32 are in numbers, thus, we need to convert the values to string before transmit them to MQTT broker [line #130 to #133].
- Line #129 are set of arrays which meant to store the converted number (as string).
- dtostrf(float_value, minimum_length, number_digits_after_decimal, where_to_store_string)

```
#130 dtostrf(bme.readAltitude(1013.25),8,2,altitudeValue);
e.g. Altitude: 100199.50
```

***8 digits length**
***2 decimal or number of floating point is 2**

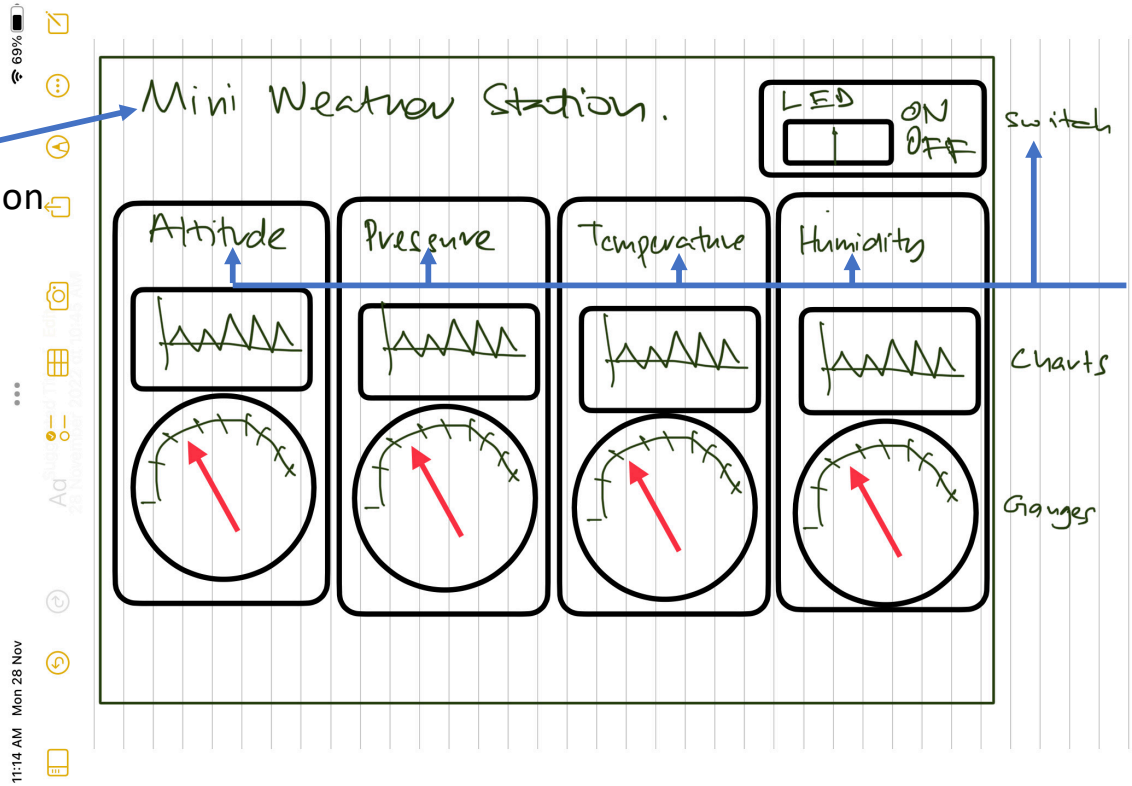
- **client.publish** syntax is used to pass the value captured by ESP32 to MQTT broker.
- The information will be published according to its **topic**.

```
143     client.publish("stationA/altitude", altitudeValue);
144     client.publish("stationA/pressure", pressureValue);
145     client.publish("stationA/humid", humidValue);
146     client.publish("stationA/temp", tempValue);
```

iv. MQTT Broker – Draft Dashboard Layout

- Since this exercise will use dashboard, it will be better that we draft the dashboard layout first, together with the title or properties of the panel.

Tab:
Mini Weather Station



Groups:
Switch
Groups
Altitude
Pressure
Temperature
Humidity

Dashboard Tips:
-Tab = Title
-Group = Sub title
-Topic = mqtt in / mqtt out

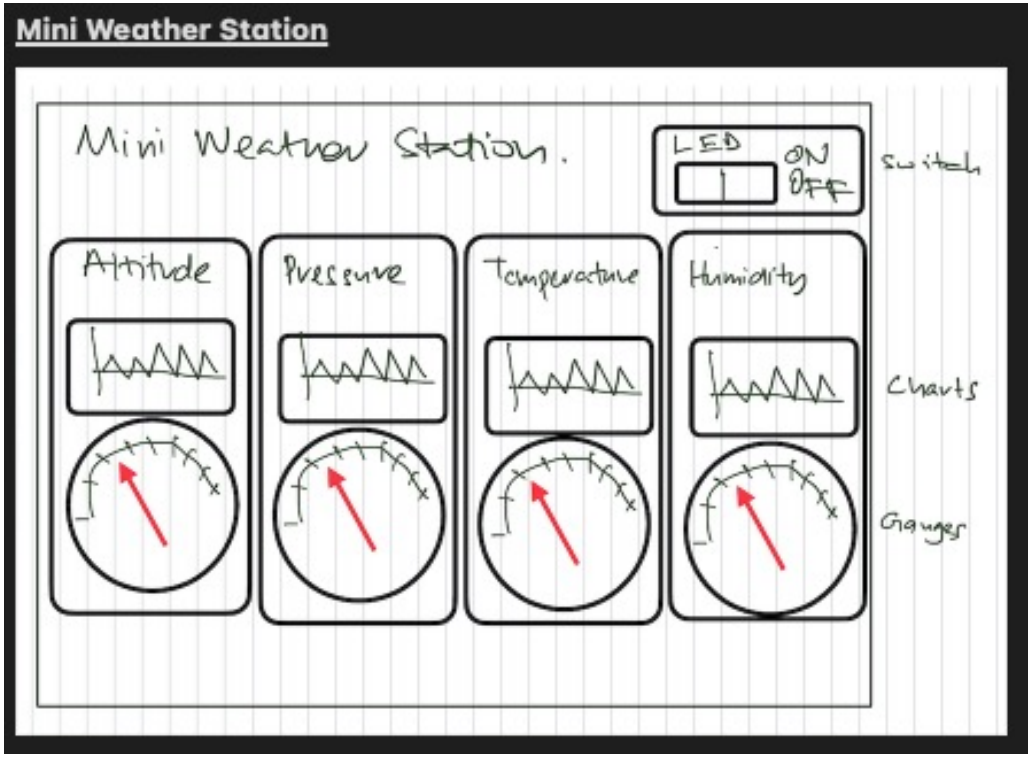
Port:
1991

Topic:
mqtt in:
stationA/altitude
stationA/pressure
stationA/temp
stationA/humid

mqtt out:
id1

iv. MQTT Broker – Create Dashboard

- We will start by creating the dashboard properties because it is **easier** compared drag and drop nodes, then create dashboard.



Part 1: Dashboard
1. Design dashboard as drafted.
2. Add 1 tab and 5 groups.
3. Tab: Mini Weather Station.
4. Groups: Altitude, Pressure, Temperature, Humidity and LED Switch.
5. Press Deploy tab to compile.

Part 2: Nodes
1. Place 1 Aedes Broker, 4 MQTT In nodes and 1 MQTT Out node.
2. Name Aedes Broker as **Stesyen KPT UTHM.**
3. Change the MQTT port from **1883** to **1991.**
4. Remember to find the Node-RED's IP address of your laptop/pc.
—>Mine is **192.168.0.166.**
—>Use **ipconfig** or **ifconfig** to find IP address.
3. MQTT In: [waiting data from ESP32]
—>Topic: **stationA/altitude, stationA/pressure, stationA/temp, stationA/humid.**
4. MQTT Out: [sending data to ESP32]
—>Topic: id1
5. Press Deploy tab to compile.
6. Include 4 gauges & 4 charts and wire them to every MQTT In nodes.
7. Add switch interface to connect to MQTT Out node.
8. Double click gauges & charts. Set its group as what have been done in Part 1.
9. Deploy.
10. You may add debug nodes at every MQTT.
—> Its another method of viewing incoming information.
—> Restart the Node-RED server if nothing appear at debug tab.

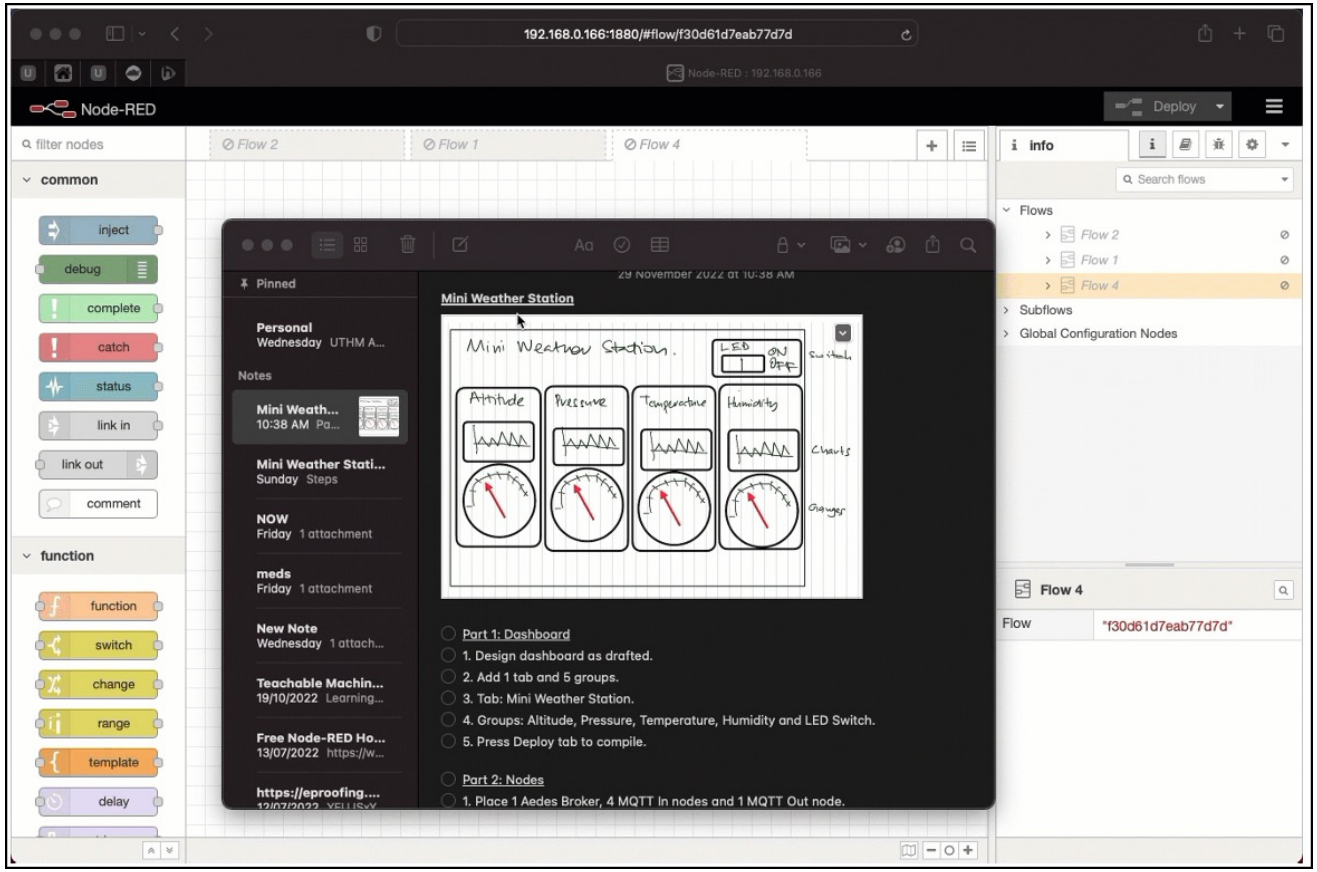
FINALLY.. Testing the whole system. Troubleshoot if you encounter any problems.
+You may also import this exercise into your system, don't forget to change the IP address of the broker and also the IP address at sketch.
+To download your work or email —> EXPORT
+To upload js file into your system, use IMPORT

+If problem persists, mail me at scriptworkz.01@gmail.com

OK THE END... tq

iv. MQTT Broker – Create Dashboard

- Download full video here.



END OF EXERCISE