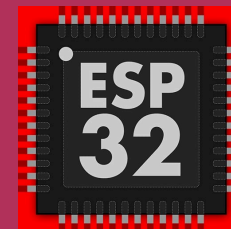
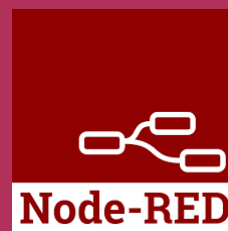


MQTT Broker, Node-RED and ESP32 IoT Development Board

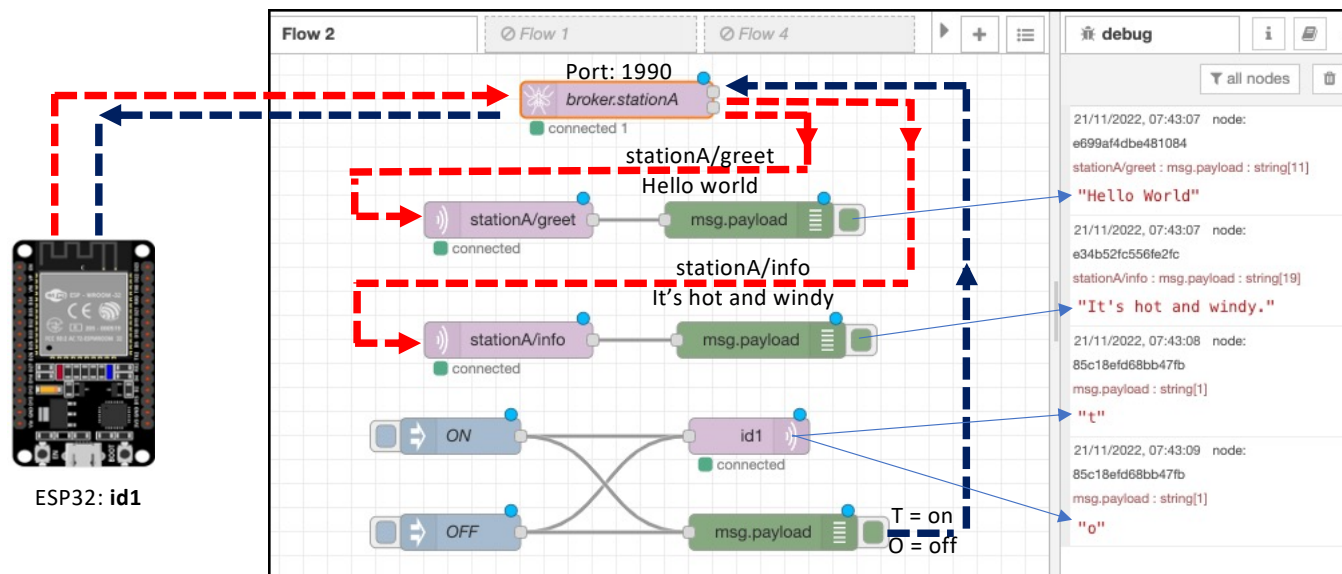
[PART 1b]



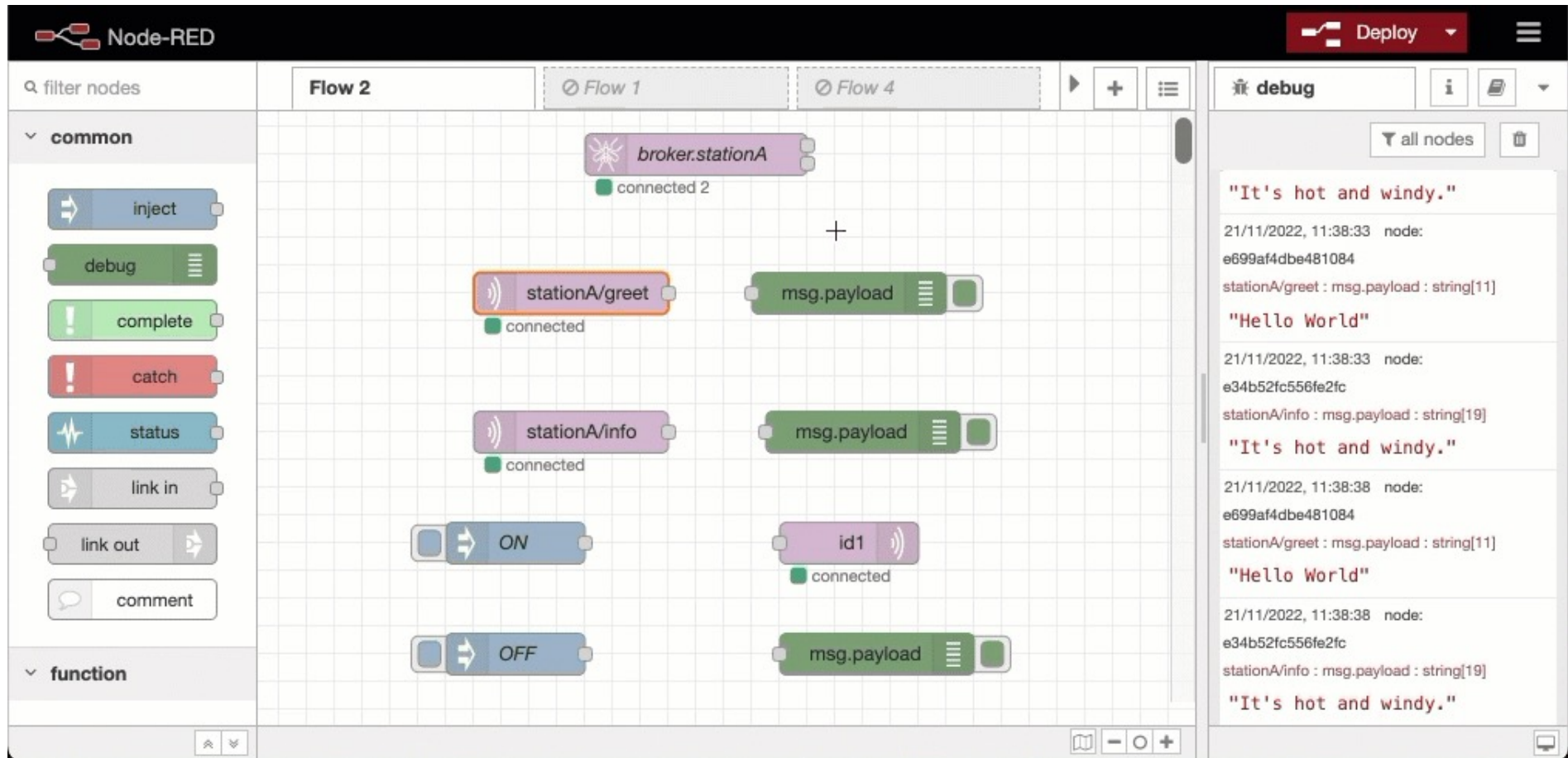
PART 1b

i. Integration MQTT and Node-RED with ESP32

- Add new **Flow**, **Aedes Broker**, **2 x MQTT In** nodes, **1 x MQTT Out**, **2 x Debug** and **2 x Inject** nodes.
- Set the broker's port to 1990.
- In this exercise, string of text will be sent from microcontroller to Node-RED and will be displayed at debug tab.
- LED at ESP32 will be switch on and off from Node-RED when inject node is click.



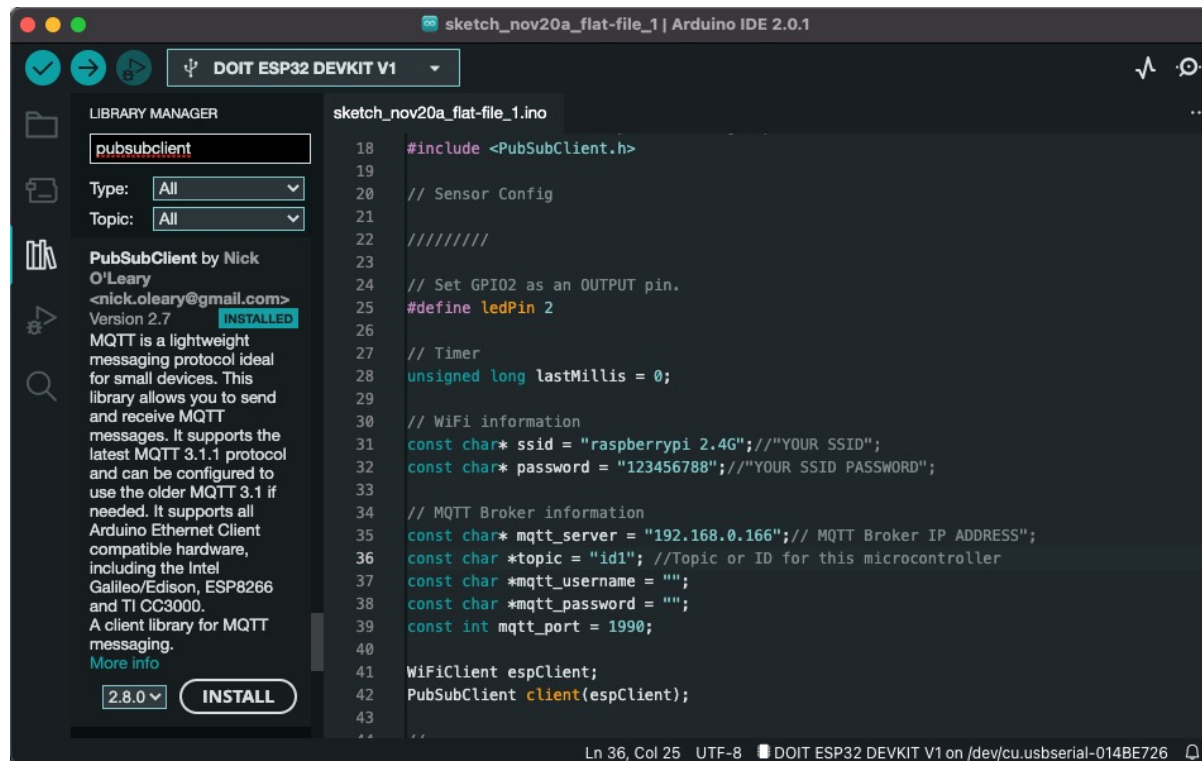
ii. How It Works



i. Arduino Client MQTT Library

- Install Nick O'Leary's PubSubClient library.

Sketch > Include Library > Manage Libraries



ii. Arduino Code Snippet

- Download the file working file from [sketch nov20a flat-file 1.ino](#).

sketch_nov20a_flat-file_1.ino

```
15
16 // #include <ESP8266WiFi.h> //if you are using nodemcu
17 #include <WiFi.h> //if you are using esp32
18 #include <PubSubClient.h>
19
20 // Set GPIO2 as an OUTPUT pin.
21 #define ledPin 2
22
```

Remove // at **#16** and put // at **#17** if you are using NodeMCU.

ESP32 has an on board led defined as GPIO2. **#21** mapped ledPin to 2. [2 = GPIO2].

ii. Arduino Code Snippet

Must use 2.4GHz WiFi only

- Download the file working file from [sketch nov20a flat-file 1.ino](#).

```
26 // WiFi information
27 const char* ssid = "raspberrypi 2.4G"; //"YOUR SSID";
28 const char* password = "123456788"; //"YOUR SSID PASSWORD";
29
30 // MQTT Broker information
31 const char* mqtt_server = "192.168.0.166"; // MQTT Broker IP ADDRESS";
32 const char *topic = "id1"; //Topic or ID for this microcontroller
33 const char *mqtt_username = "";
34 const char *mqtt_password = "";
35 const int mqtt_port = 1990;
36
37
38 WiFiClient espClient; //A WiFi library that Set ESP32 as client
39 PubSubClient client(espClient); //A client library that support MQTT
40
```

#27 & #28 Your access point ssid and password.

#31 Laptop / desktop / SBC IP address which running Node-RED service.

#32 'id1' is the identification for this microcontroller. Each microcontroller must have a unique identification.

#33 & #34 is broker's username & password. Double click aedes broker to set parameters. "" = no uname& pwd.

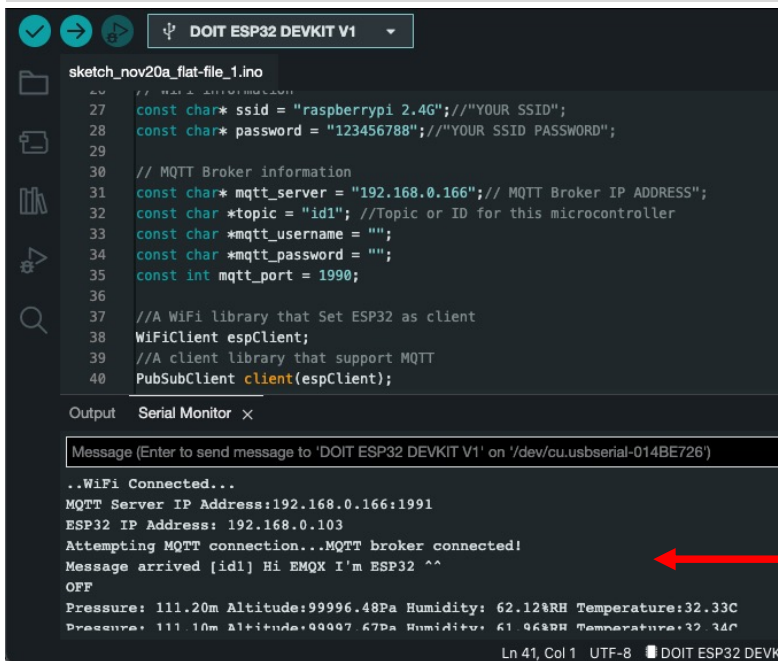
#35 Change default port number from 1883 to **1990**. Must aware when deciding port number, incorrect port will cause other program fail to execute.

ii. Arduino Code Snippet

- Download the file working file from [sketch nov20a flat-file 1.ino](#).

#41 to #63 and #65 to #79 are 2 functions that works on connecting the ESP32 to network and also to Aedes broker.

This process can be viewed thru Serial Monitor.



```
sketch_nov20a_flat-file_1.ino
// ...
27 const char* ssid = "raspberrypi 2.4G";//"YOUR SSID";
28 const char* password = "123456788";//"YOUR SSID PASSWORD";
29
30 // MQTT Broker information
31 const char* mqtt_server = "192.168.0.166";// MQTT Broker IP ADDRESS";
32 const char *topic = "idl"; //Topic or ID for this microcontroller
33 const char *mqtt_username = "";
34 const char *mqtt_password = "";
35 const int mqtt_port = 1990;
36
37 //A WiFi library that Set ESP32 as client
38 WiFiClient espClient;
39 //A client library that support MQTT
40 PubSubClient client(espClient);

Output Serial Monitor x
Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on '/dev/cu.usbserial-014BE726')

..WiFi Connected...
MQTT Server IP Address:192.168.0.166:1991
ESP32 IP Address: 192.168.0.103
Attempting MQTT connection...MQTT broker connected!
Message arrived [idl] Hi EMQX I'm ESP32 ^^
OFF
Pressure: 111.20m Altitude:99996.48Pa Humidity: 62.12%RH Temperature:32.33C
Pressure: 111.10m Altitude:99997.67Pa Humidity: 61.96%RH Temperature:32.34C
Ln 41, Col 1 UTF-8 DOIT ESP32 DEVKIT V1
```

void setup()

```
41 void setup() {
42     Serial.begin(115200);
43
44     // Set ledPin as Output
45     pinMode(ledPin, OUTPUT);
46     // Switch off Led
47     digitalWrite(ledPin, HIGH);
48
49     // Connecting to a MQTT broker
50     client.setServer(mqtt_server, mqtt_port);
51     client.setCallback(callback);
52
53     //function call for WiFi connection
54     setup_wifi();
55
56     Serial.println("WiFi Connected..");
57     Serial.print("MQTT Server IP Address:");
58     Serial.print(mqtt_server);
59     Serial.print(":");
60     Serial.println(String(mqtt_port));
61     Serial.print("ESP32 IP Address: ");
62     Serial.println(WiFi.localIP());
63 }
```

void setup_wifi()

```
64 //connecting to WiFi - 2.4GHz only
65 void setup_wifi() {
66     delay(10);
67     // Start by connecting to a WiFi network
68     Serial.println();
69     Serial.print("Connecting to ");
70     Serial.println(ssid);
71
72     WiFi.begin(ssid, password);
73
74     while (WiFi.status() != WL_CONNECTED) {
75         delay(500);
76         Serial.print(".");
77     }
78 }
79
80
```


ii. Arduino Code Snippet

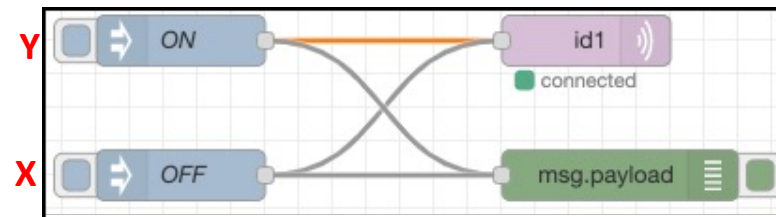
- Download the file working file from [sketch nov20a flat-file 1.ino](#).

```
80
81 // Instructions received from broker
82 void callback(char* topic, byte* payload, unsigned int length) {
83     /* To display message sent from broker */
84     Serial.print("Message arrived [");
85     Serial.print(topic);
86     Serial.print("] ");
87
88     for (int i = 0; i < length; i++) {
89         Serial.print((char)payload[i]);
90     }
91
92     /* This section is for controlling output: LED, servo etc */
93     /* ----- */
94     Serial.println();
95     if ((char)payload[0] == 't') {
96         Serial.println("ON");
97         digitalWrite(ledPin, LOW);
98     } else {
99         Serial.println("OFF");
100        digitalWrite(ledPin, HIGH);
101    }
102    /* ----- */
103 }
104
```

This section deals with incoming message from Node-RED i.e., Aedes broker.

Y will switch on built in led on ESP, while **X** will switch off the led.

Every time the **X/Y** is press, **t** or **o** will be transmitted to ESP32. and process at this section.



```
22/11/2022, 17:37:26 node:
85c18efd68bb47fb
msg.payload : string[1]
"t"

22/11/2022, 17:37:28 node:
85c18efd68bb47fb
msg.payload : string[1]
"o"
```

ii. Arduino Code Snippet

- Download the file working file from [sketch nov20a flat-file 1.ino](#).

```
130
131 void loop() {
132     //Function to establish connection to MQTT Broker
133     if (!client.connected()) {
134         reconnect();
135     }
136
137     // publish a message roughly every second.
138     if (millis() - lastMillis > 5000) {
139         lastMillis = millis();
140         client.publish("stationA/greet", "Hello World"); //PUBLISH TO TOPIC /hello MSG world order
141         client.publish("stationA/info", "It's hot and windy."); //PUBLISH TO TOPIC /hello MSG world order
142         /// Serial.println (" Hello World ");
143     }
144
145     client.loop();
146 }
147
```

The syntax `client.publish()` is used to send information or data from ESP32 to Node-RED.

Incorrect topic on either side, i.e., ESP32 or Aedes broker, causes no information will be publish.

Messages will be transmitted every 5 seconds, whereby 1000 is equivalent with 1 second.

END OF PART 1b