

## PROJECT DEVELOPMENT PHASE -SPRINT 1

Date	18 November 2022
Team ID	PNT2022TMID40881
Project Name	Industry specific intelligent fire management system

The screenshot displays the Wokwi IoT simulator interface. On the left, the 'sketch.ino' file is open, showing a C++ program for an ESP32. The code includes libraries for WiFi and MQTT, defines pins and constants, and implements a loop that reads temperature from an NTC sensor and publishes it to an MQTT topic. The right side of the interface shows a 'Simulation' window with a circuit diagram of the ESP32 and an NTC temperature sensor. Below the diagram, a console log displays the program's output, including MQTT connection status, temperature readings, and published payloads.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #define temp_pin 15
4 void callback(char* topic, byte* payload, unsigned int payloadLength);
5 #define ORG "tca72r"
6 #define DEVICE_TYPE "raspberrypi"
7 #define DEVICE_ID "12345"
8 #define TOKEN "12345678"
9 String data3;
10
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Data/fmt/json";
13 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;
17
18 WiFiClient wifiClient;
19 PubSubClient client(server, 1883, callback, wifiClient);
20 void setup() {
21   Serial.begin(9600);
22   analogReadResolution(10);
23   pinMode(32, INPUT);
24   pinMode(14, OUTPUT);
25
26   wifiConnect();
27   mqttConnect();
28 }
29
30 void loop() {
31   const float BETA = 3950;
32   int analogValue = analogRead(A4);
33   float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) - 273.15;
34
35   Serial.print("Temperature: ");
```

Simulation window output:

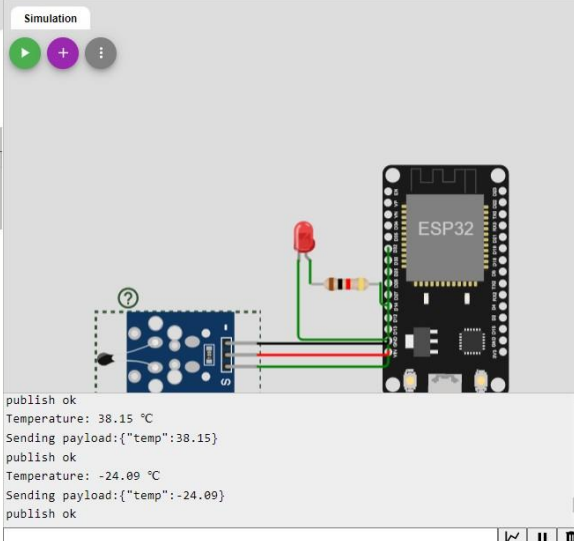
```
publish ok
Temperature: 6.06 °C
Sending payload:{"temp":6.06}
publish ok
Temperature: 46.94 °C
Sending payload:{"ALERT":46.94,"lat":12.879221 , "lon":79.122258}
publish ok
```

Wokwi project: 348643613434970708

```

34
35 Serial.print("Temperature: ");
36 Serial.print(temp);
37 Serial.println(" °C");
38 int lat=12879221;
39 int lon=79122258;
40
41
42 if(temp>=40){
43   PublishData2(temp);
44 }
45 digitalWrite(14, HIGH);
46 }else{
47   digitalWrite(14, LOW);
48   PublishData1(temp);
49 }
50 delay(1000);
51 if(!client.loop()){
52   mqttconnect();
53 }
54 //delay(2000);
55 }
56
57 void PublishData1(float tem){
58   mqttconnect();
59   String payload= "{\"temp\":\"";
60   payload += tem;
61   payload += "\"}";
62
63   Serial.print("Sending payload:");
64   Serial.println(payload);
65
66   if(client.publish(publishTopic,(char*)payload.c_str())){
67     Serial.println("publish ok");

```



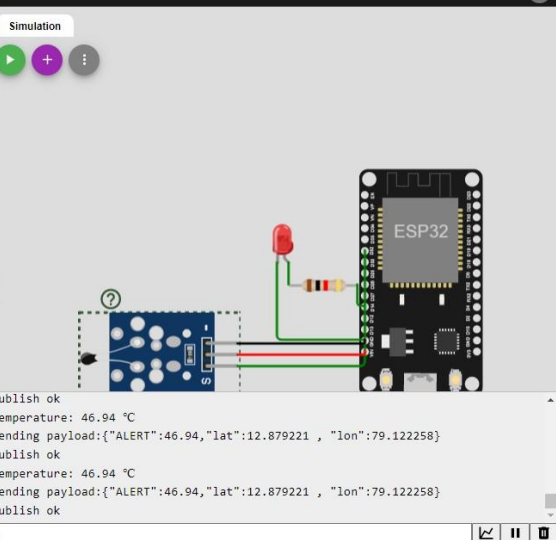
publish ok  
Temperature: 38.15 °C  
Sending payload:{"temp":38.15}  
publish ok  
Temperature: -24.09 °C  
Sending payload:{"temp":-24.09}  
publish ok

Wokwi project: 348643613434970708

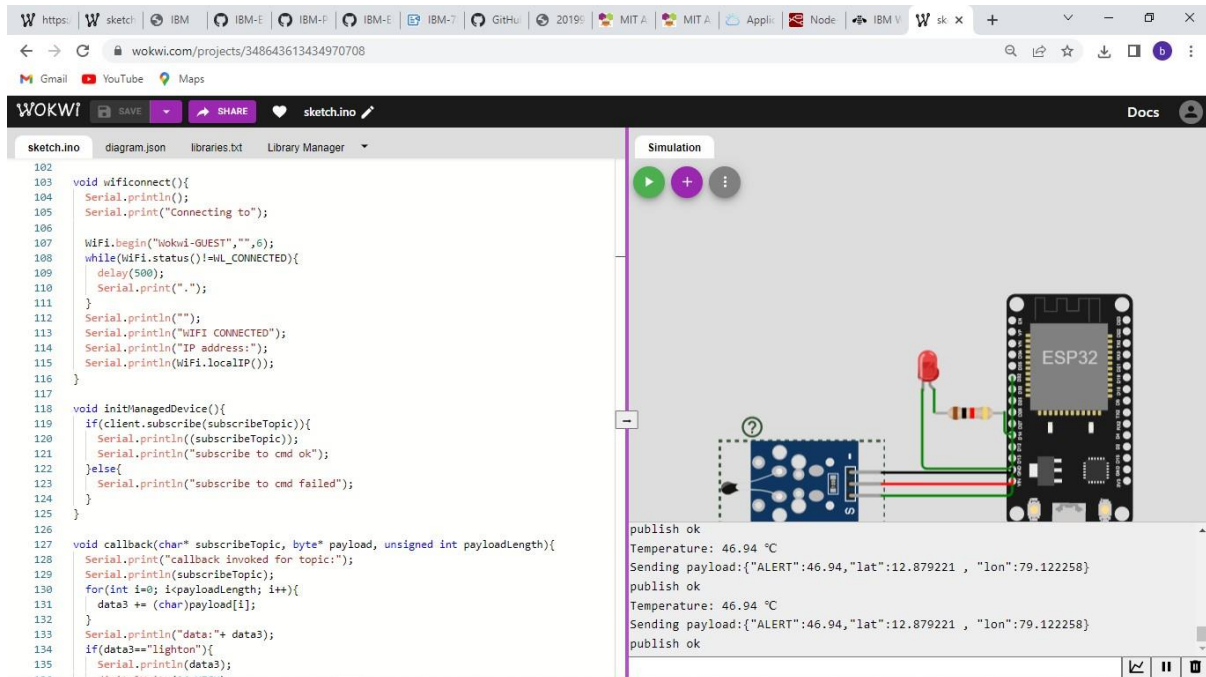
```

69   Serial.println("publish failed");
70 }
71
72 void PublishData2(float tem){
73   mqttconnect();
74   String payload= "{\"ALERT\":\"";
75   payload += tem;
76   payload += "\",\"";
77   payload += "\"lat\":\"12.879221 , \"lon\":\"79.122258\"";
78   payload += "\"}";
79
80   Serial.print("Sending payload:");
81   Serial.println(payload);
82
83   if(client.publish(publishTopic,(char*)payload.c_str())){
84     Serial.println("publish ok");
85   } else{
86     Serial.println("publish failed");
87   }
88 }
89
90 void mqttconnect(){
91   if(!client.connected()){
92     Serial.print("Reconnecting to");
93     Serial.println(server);
94     while(!client.connect(clientID, auth/method, token)){
95       Serial.print(".");
96       delay(500);
97     }
98     initManagedDevice();
99     Serial.println();
100 }
101 }
102

```



publish ok  
Temperature: 46.94 °C  
Sending payload:{"ALERT":46.94,"lat":12.879221 , "lon":79.122258}  
publish ok  
Temperature: 46.94 °C  
Sending payload:{"ALERT":46.94,"lat":12.879221 , "lon":79.122258}  
publish ok



## IMAGES OF IBM CLOUD:

