

## Sprint 1:

Date	01 November 2022
Team ID	PNT2022TMID42331
Project Name	Industry specific intelligent fire management system

### Program:

```
#include <WiFi.h>
```

```
#include <PubSubClient.h>
```

```
void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
```

```
#define ORG "tm546c"
```

```
#define DEVICE_TYPE "abcd"
```

```
#define DEVICE_ID "1234"
```

```
#define TOKEN "12345678"
```

```
String data3;
```

```
char server[]= ORG ".messaging.internetofthings.ibmcloud.com";
```

```
char publishTopic[]="iot-2/evt/Data/fmt/json";
```

```
char subscribeTopic[]="iot-2/cmd/test/fmt/String";
```

```
char authMethod[]="use-token-auth";
```

```
char token[]=TOKEN;
```

```
char clientID[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID";
```

```
WiFiClient wifiClient;
```

```
PubSubClient client(server,1883,callback,wifiClient);
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  analogReadResolution(10);
```

```
  pinMode(32,INPUT);
```

```
  pinMode(14,OUTPUT);
```

```
  wificonnect();
```

```
  mqttconnect();
```

```
}
```

```
void loop() {
```

```
  const float BETA = 3950; // should match the Beta Coefficient of the thermistor
```

```
  int analogValue = analogRead(A4);
```

```
  float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) - 273.15;
```

```
  //float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) - 273.15;
```

```
  Serial.print("Temperature: ");
```

```

Serial.print(temp);
Serial.println(" °C");
if(temp>=35){
    PublishData2(temp);
    digitalWrite(14, HIGH);
}else{
    digitalWrite(14, LOW);
    PublishData1(temp);
}
delay(1000);
if(!client.loop()){
    mqttconnect();
}

//delay(2000);
}
void PublishData1(float tem){
    mqttconnect();
    String payload= "{\"temp\":";
    payload += tem;
    payload+="}";

    Serial.print("Sending payload:");
    Serial.println(payload);

    if(client.publish(publishTopic,(char*)payload.c_str())){
        Serial.println("publish ok");
    } else{
        Serial.println("publish failed");
    }
}
void PublishData2(float tem){
    mqttconnect();
    String payload= "{\"ALERT\":";
    payload += tem;
    payload+="}";

    Serial.print("Sending payload:");
    Serial.println(payload);

    if(client.publish(publishTopic,(char*)payload.c_str())){
        Serial.println("publish ok");
    } else{
        Serial.println("publish failed");
    }
}
void mqttconnect(){
    if(!client.connected()){

```

```

    Serial.print("Reconnecting to");
    Serial.println(server);
    while(!client.connect(clientID, authMethod, token)){
        Serial.print(".");
        delay(500);
    }
    initManagedDevice();
    Serial.println();
}

void wificonnect(){
    Serial.println();
    Serial.print("Connecting to");

    WiFi.begin("Wokwi-GUEST","");
    while(WiFi.status()!=WL_CONNECTED){
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WIFI CONNECTED");
    Serial.println("IP address:");
    Serial.println(WiFi.localIP());
}

void initManagedDevice(){
    if(client.subscribe(subscribeTopic)){
        Serial.println((subscribeTopic));
        Serial.println("subscribe to cmd ok");
    }else{
        Serial.println("subscribe to cmd failed");
    }
}

void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength){
    Serial.print("callback invoked for topic:");
    Serial.println(subscribeTopic);
    for(int i=0; i<payloadLength; i++){
        data3 += (char)payload[i];
    }
    Serial.println("data:" + data3);
    if(data3=="lighton"){
        Serial.println(data3);
        digitalWrite(14,HIGH);
    }else{
        Serial.println(data3);
        digitalWrite(14,LOW);
    }
}

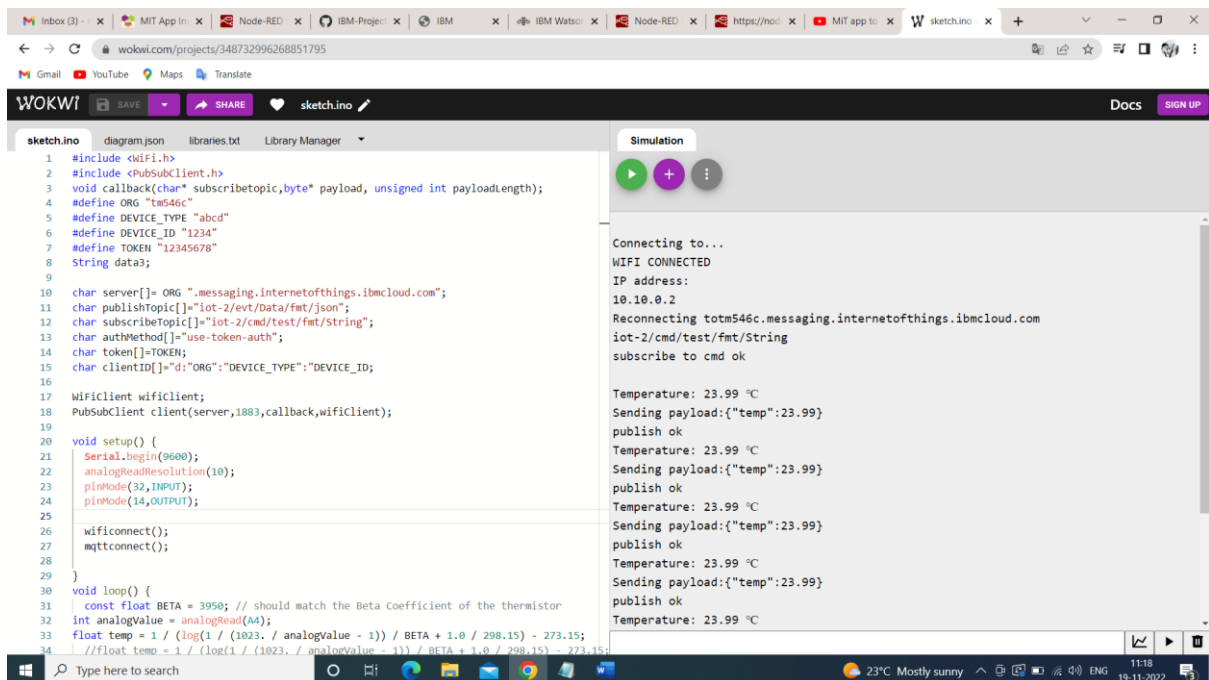
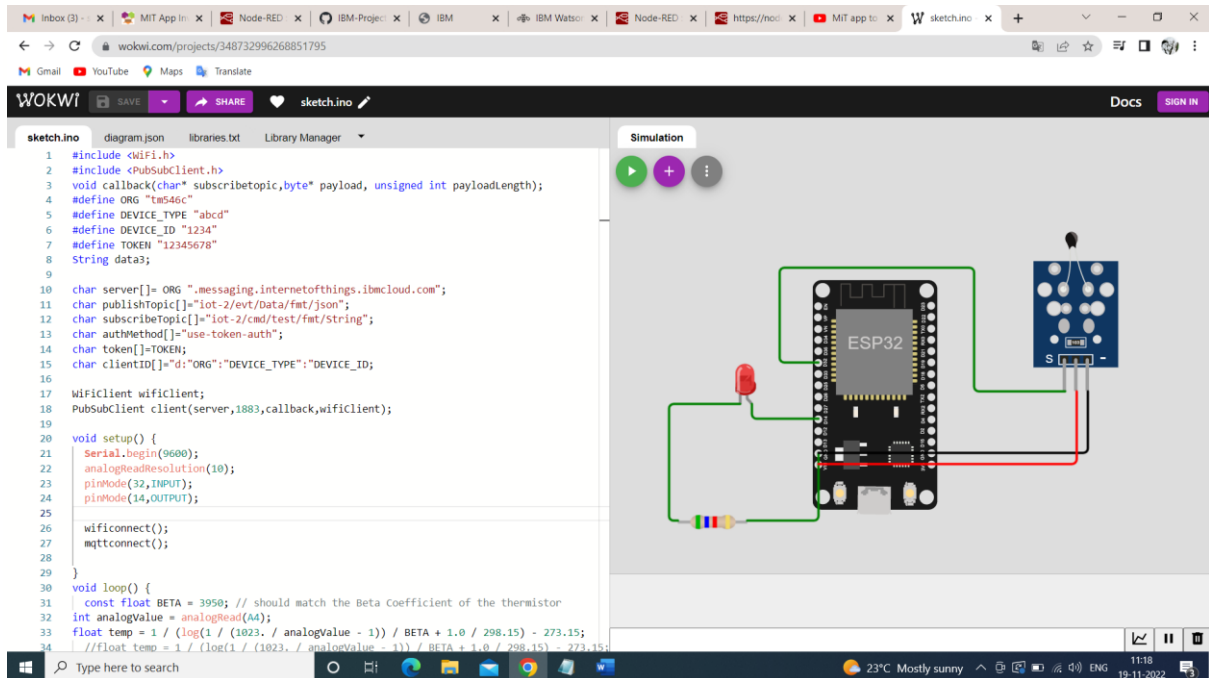
```

```

}
data3="";
}

```

## Wokwi output:



## Wokwi Link:

<https://wokwi.com/projects/348732996268851795>

The screenshot displays the IBM Watson IoT Platform interface. At the top, a browser window shows the URL `tm546c.internetofthings.ibmcloud.com/dashboard/devices/browse`. The platform header includes the IBM Watson IoT Platform logo and a user profile for `710119205005@smartinternz.com` with ID `tm546c`.

The main navigation bar contains 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left provides quick access to various system components. The central area shows a list of devices, with 'Device ID 1234' selected. This device is in a 'Connected' state. Below the device list, the 'Recent Events' tab is active, displaying a live stream of data events. A notification at the bottom right indicates '1 Simulation running'.

Event	Value	Format	Last Received
Data	{\"temp\":23.99}	json	a few seconds ago
Data	{\"temp\":23.99}	json	a few seconds ago
Data	{\"temp\":23.99}	json	a few seconds ago
Data	{\"temp\":23.99}	json	a few seconds ago
Data	{\"temp\":23.99}	json	a few seconds ago