

## SPRINT - 1

Date	01 November 2022
Team ID	PNT2022TMID17702
Project Name	Industry specific intelligent fire management system

### WOKWI WEB URL:

<https://wokwi.com/projects/347953543893746258>

### SNAPSHOTS OF SIMULATION:

The screenshot displays the Wokwi web interface for a project titled "Industry-specific intelligent fire management system.ino". The interface is split into two main sections: a code editor on the left and a simulation window on the right.

**Code Editor (Left):** The code is written in C++ for an ESP32 microcontroller. It includes the following components:

- Includes:** `<WiFi.h>` and `<PubSubClient.h>`.
- Constants:** `temp_pin` is defined as 15. The `callback` function is defined to take a topic, payload, and payload length.
- Configuration:** The `ORG` is "ova7j8", `DEVICE_TYPE` is "esp32", `DEVICE_ID` is "1234", and `TOKEN` is "12345678".
- Server Setup:** A `server` is defined as "messaging.internetofthings.ibmcloud.com". The `publishTopic` is "iot-2/evt/data/fmt/json", the `subscribeTopic` is "iot-2/cmd/test/fmt/String", the `authMethod` is "use-token-auth", and the `token` is "12345678".
- Client Setup:** A `WiFiClient` is created, and a `PubSubClient` is initialized with the server, port 1883, and the callback function.
- Setup Function:** The `setup` function initializes the serial port at 9600 baud, sets the `analogReadResolution` to 10, and configures pins 32 and 14 as `INPUT` and `OUTPUT` respectively.
- Connect Function:** The `wificonnect` function is called.

**Simulation Window (Right):** The simulation shows an ESP32 microcontroller connected to a temperature sensor (red LED) and a relay (blue component). The output log at the bottom of the simulation window shows the following data:

```
Temperature: 23.99 °C
Sending payload:{"temp":23.99}
publish ok
Temperature: 23.99 °C
Sending payload:{"temp":23.99}
publish ok
Temperature: 23.99 °C
```

WOKWI SAVE SHARE Industry-specific intelligent fire management system.ino Docs

sketch.ino diagram.json libraries.txt Library Manager

```

24 // should match the beta coefficient of the thermistor
25
26 void setup() {
27   Serial.begin(9600);
28   analogReadResolution(10);
29   pinMode(32, INPUT);
30   pinMode(14, OUTPUT);
31
32   wifiConnect();
33   mqttConnect();
34 }
35
36 void loop() {
37   const float BETA = 3950; // should match the Beta Coefficient of the thermistor
38   int analogValue = analogRead(A4);
39   float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) - 273.15;
40   //float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) - 273.15;
41   Serial.print("Temperature: ");
42   Serial.print(temp);
43   Serial.println(" °C");
44   if(temp >= 35){
45     PublishData2(temp);
46     digitalWrite(14, HIGH);
47   }else{
48     digitalWrite(14, LOW);
49     PublishData1(temp);
50   }
51   delay(1000);
52   if(!client.loop()){
53     mqttConnect();
54   }
55   //delay(2000);
56 }
57
58 void PublishData1(float tem){

```

Simulation 01:41.643 99%

Temperature: 23.99 °C  
 Sending payload:{"temp":23.99}  
 publish ok  
 Temperature: 23.99 °C  
 Sending payload:{"temp":23.99}  
 publish ok  
 Temperature: 23.99 °C

WOKWI SAVE SHARE Industry-specific intelligent fire management system.ino Docs

sketch.ino diagram.json libraries.txt Library Manager

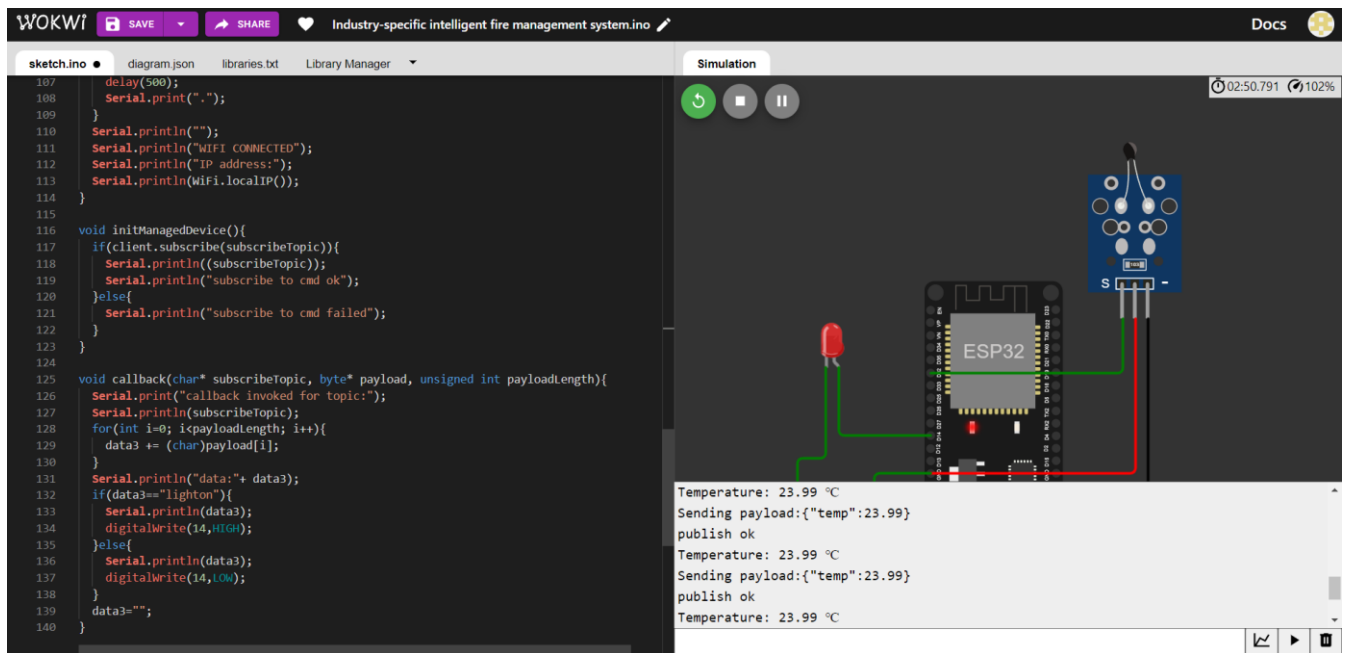
```

58 void PublishData1(float tem){
59   mqttConnect();
60   String payload= "{\"temp\":\"";
61   payload += tem;
62   payload += "\"}";
63
64   Serial.print("Sending payload:");
65   Serial.println(payload);
66
67   if(client.publish(publishTopic,(char*)payload.c_str())){
68     Serial.println("publish ok");
69   } else{
70     Serial.println("publish failed");
71   }
72 }
73 void PublishData2(float tem){
74   mqttConnect();
75   String payload= "{\"ALERT\":\"";
76   payload += tem;
77   payload += "\"}";
78
79   Serial.print("Sending payload:");
80   Serial.println(payload);
81
82   if(client.publish(publishTopic,(char*)payload.c_str())){
83     Serial.println("publish ok");
84   } else{
85     Serial.println("publish failed");
86   }
87 }
88 void mqttConnect(){
89   if(!client.connected()){
90     Serial.print("Reconnecting to");
91     Serial.println(server);

```

Simulation 02:17.060 100%

Temperature: 23.99 °C  
 Sending payload:{"temp":23.99}  
 publish ok  
 Temperature: 23.99 °C  
 Sending payload:{"temp":23.99}  
 publish ok  
 Temperature: 23.99 °C



## Images of IBM Watson IoT Platform Recent events:

