

## Project Development Phase

### Sprint – 4

Date	19 - 11 - 2022
Team ID	PNT2022TMID06114
Project Name	Industry Specific Intelligent Fire Management System

## Send Data from Wokwi to Watson IOT and Watson IOT to NodeRed:-

### Wokwi Simulation:

The screenshot displays the Wokwi web-based simulation environment. On the left, the 'sketch.ino' file is open, showing the following code:

```
1 #include <time.h>
2 #include <WiFi.h>
3 #include <PubSubClient.h>
4
5 #define ORG "wa9fzx"
6 #define DEVICE_TYPE "ESP32"
7 #define DEVICE_ID "4321"
8 #define TOKEN "87654321"
9
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/data/fmt/json";
12 char authMethod[] = "use-token-auth";
13 char token[] = TOKEN;
14 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
15
16 WiFiClient wificlient;
17 PubSubClient client(server, 1883, wificlient);
18
19 float temperature = 0;
20 int gas = 0;
21 int flame = 0;
22
23 String flame_status = "";
24 String Gas_status = "";
25 String exhaust_fan_status = "";
26 String sprinkler_status = "";
27
28
29 void setup() {
30   Serial.begin(99900);
31   //Serial.println("Setup");
32 }
```

On the right, the 'Simulation' window shows a 3D model of an ESP32 board. Below the model, the simulation log displays the following messages:

```
Connecting to Wifi...WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to
wa9fzx.messaging.internetofthings.ibmcloud.com
Publish OK
Publish OK
Publish OK
```

The bottom of the image shows a Windows taskbar with various application icons and a system clock indicating 9:35 PM on 17-Nov-22.

## Watson IOT Platform Output:

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for 'Application Details - IBM Cloud', 'Node-RED : node-red-efyo-20', 'IBM Watson IoT Platform', 'Inbox (2,660) - thehelpinghand', and 'sketch.ino - Wokwi Arduino an'. The browser address bar shows the URL 'wa9fzx.internetofthings.ibmcloud.com/dashboard/devices/browse'. The dashboard header includes the 'IBM Watson IoT Platform' logo, a user profile for 'goldenpandi24@gmail.com' with ID 'wa9fzx', and an 'Add Device' button.

The main content area is titled 'Browse' and shows a list of devices. The first device is selected, showing details for Device ID '4321', Status 'Connected', Device Type 'ESP32', Class ID 'Device', and Date Added 'Nov 18, 2022 10:56 AM'. Below this, a 'Recent Events' tab is active, displaying a table of live stream data.

Event	Value	Format	Last Received
data	{"gas":380,"temperature":3,"flame":819,"fire_sta...	json	a few seconds ago
data	{"gas":413,"temperature":59,"flame":985,"fire_s...	json	a few seconds ago
data	{"gas":811,"temperature":58,"flame":235,"fire_s...	json	a few seconds ago
data	{"gas":237,"temperature":40,"flame":587,"fire_s...	json	a few seconds ago

The bottom of the dashboard shows 'Items per page 50' and '1 of 1 page'. The Windows taskbar at the bottom indicates the time is 9:37 PM on 17-Nov-22.

## Node Red Output:

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow named 'Flow 3' with an 'IBM IoT' node (status: connected) on the left. Seven function nodes are connected to its output: 'temperature', 'gas', 'flame', 'Fire Status', 'Gas Status', 'Exhaust Fan', and 'Sprinkler'. All these function nodes have their outputs connected to a single 'msg.payload' node. The right-hand 'debug' console shows the following log entries:

```
number
708

11/17/2022, 10:21:00 PM node: 8398e0b1824d1d7a
iot-2/type/ESP32/id/4321/evl/data/fmt/json : msg.payload :
number
200

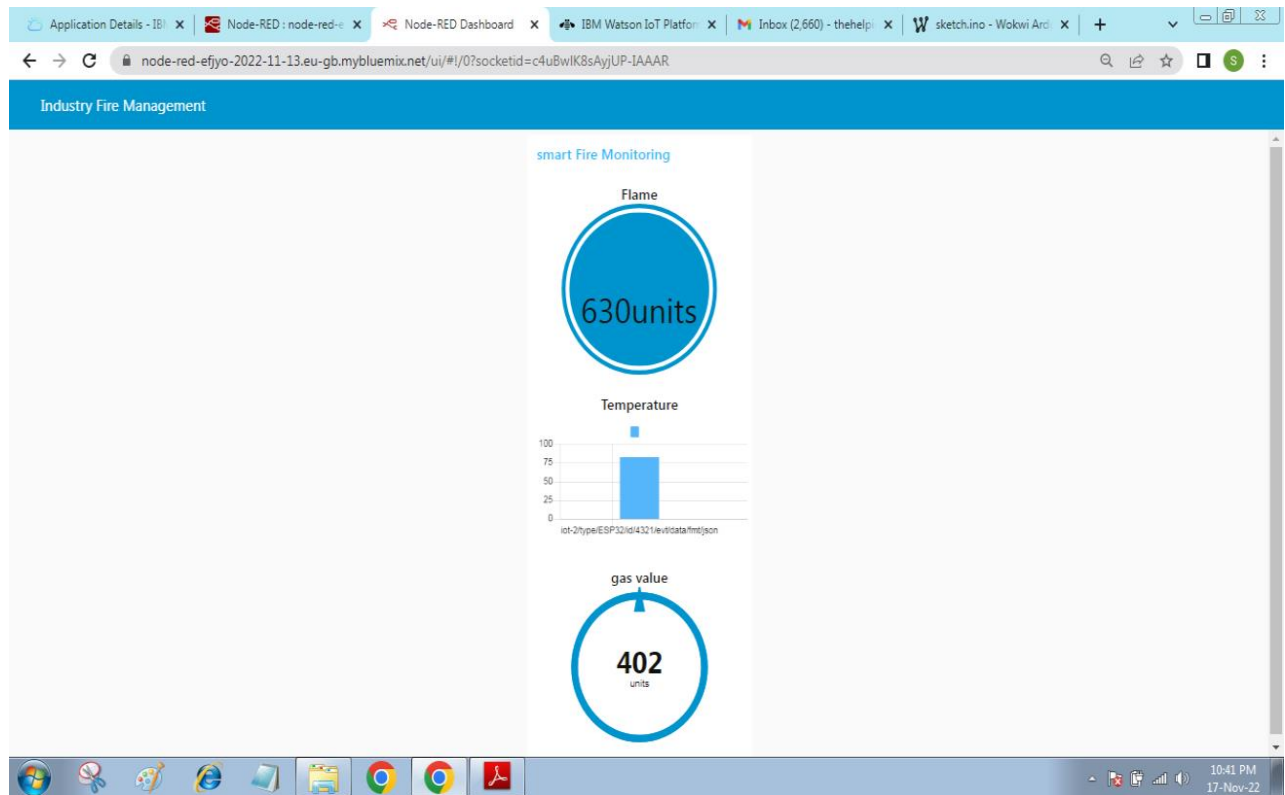
11/17/2022, 10:21:00 PM node: 8398e0b1824d1d7a
iot-2/type/ESP32/id/4321/evl/data/fmt/json : msg.payload :
string[7]
"No Fire"

11/17/2022, 10:21:00 PM node: 8398e0b1824d1d7a
iot-2/type/ESP32/id/4321/evl/data/fmt/json : msg.payload :
string[23]
"Gas Leakage is Detected"

11/17/2022, 10:21:00 PM node: 8398e0b1824d1d7a
iot-2/type/ESP32/id/4321/evl/data/fmt/json : msg.payload :
string[7]
"Working"

11/17/2022, 10:21:00 PM node: 8398e0b1824d1d7a
iot-2/type/ESP32/id/4321/evl/data/fmt/json : msg.payload :
string[11]
"Not Working"
```

## NodeRed Dashboard:



**Wokwi Link:** <https://wokwi.com/projects/348681481027060308>

**NodeRed Link:** <https://node-red-efiyo-2022-11-13.eu-gb.mybluemix.net/ui/#!/0?socketid=c4uBwIK8sAyiUP-IAAAR>

## Code:

```
#include <time.h>
#include <WiFi.h>
#include <PubSubClient.h>
#define ORG "wa9fzx"
#define DEVICE_TYPE "ESP32"
#define DEVICE_ID "4321"
#define TOKEN "87654321"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json"; char authMethod[] = "use-token-auth";
char token[] = TOKEN; char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);
float temperature = 0; int gas = 0;
int flame = 0;

String flame_status = "";

String Gas_status = "";
String exhaust_fan_status = "";
String sprinkler_status = "";
void setup() { Serial.begin(99900); wifiConnect();
mqttConnect();
}
void loop() {
srand(time(0));
//initial variables and random generated data
temperature = random(-20,125); gas = random(0,1000);
int flamereading = random(200,1024); flame = map(flamereading,200,1024,0,2); //set a flame
status
```

```

switch (flame) { case 0: flame_status = "No Fire"; break; case 1: flame_status = "Fire is
Detected";
break;
}
//send the sprinkler status
if(flame==1){ sprinkler_status = "Working";
} else{ sprinkler_status = "Not Working";
}
//toggle the fan according to gas reading
if(gas > 100){
Gas_status = "Gas Leakage is Detected"; exhaust_fan_status = "Working";
}
else{
Gas_status = "No Gas Leakage is Detected";
exhaust_fan_status = "Not Working";
}

//json format for IBM Watson
String payload = "{"; payload+="\"gas\":"; payload+=gas; payload+=",";
payload+="\"temperature\":"; payload+=(int)temperature; payload+=",";
payload+="\"flame\":"; payload+=flamereading; payload+=",";
payload+="\"fire_status\":"+"\""+flame_status+"\"",";
payload+="\"sprinkler_status\":"+"\""+sprinkler_status+"\"",";
payload+="\"Gas_status\":"+"\""+Gas_status+"\"",";
payload+="\"exhaust_fan_status\":"+"\""+exhaust_fan_status+"\""}";
if(client.publish(publishTopic, (char*) payload.c_str()))
{
Serial.println("Publish OK");
} else{
Serial.println("Publish failed");
}
delay(1000);

```

```
if (!client.loop())

{
  mqttConnect();
}

void wifiConnect()
{
  Serial.print("Connecting to ");
  Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());
}

void mqttConnect()
{
  if (!client.connected())

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

