

SPRINT-3

| | |
|---------|--|
| PROJECT | INDUSTRY-SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM |
| TEAM ID | PNT2022TMID34516 |

PYTHON CODE:

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

#define ORG "ksgtfi"
#define DEVICE_TYPE "123"
#define DEVICE_ID "123_1"
#define TOKEN "12345678"

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))

```

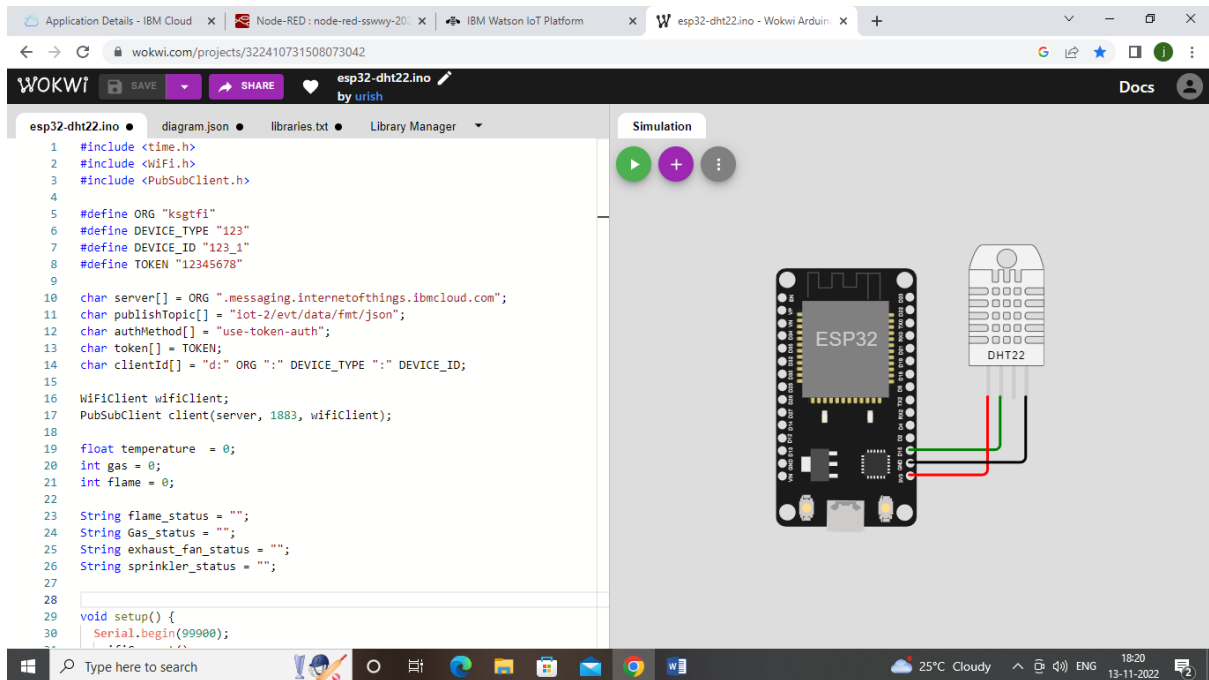
```

    {
        Serial.print(".");
        delay(500);
    }

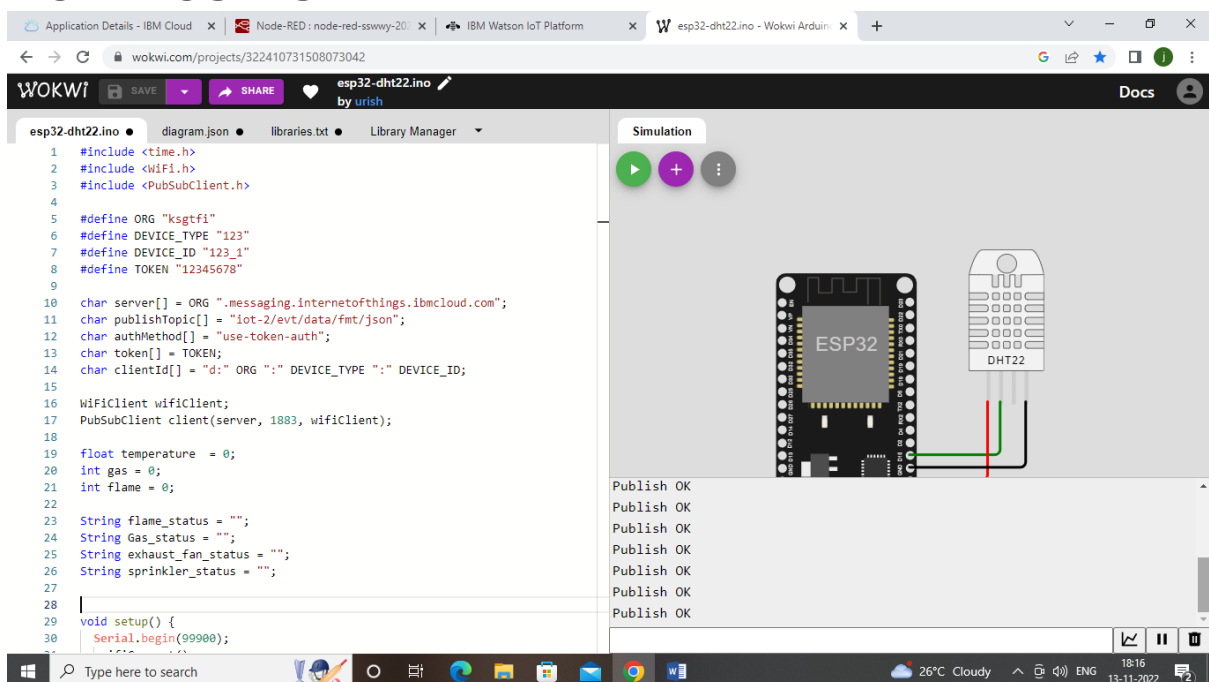
    Serial.println();
}
}

```

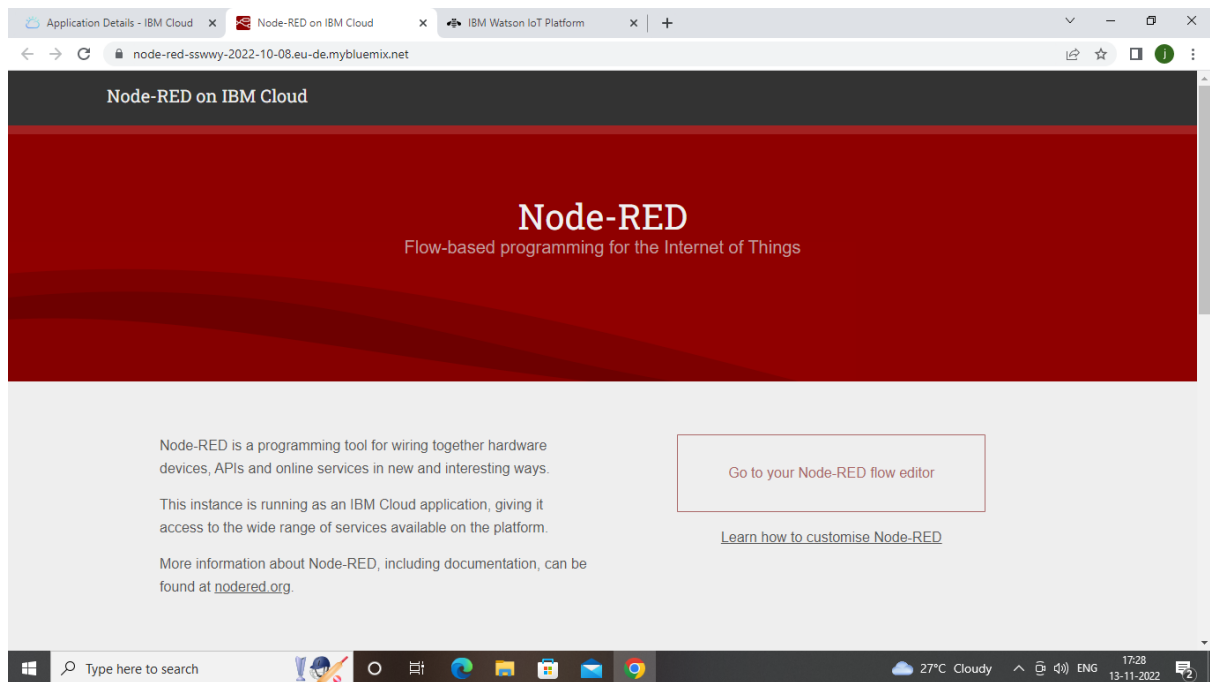
WOKWI CONNECTION:



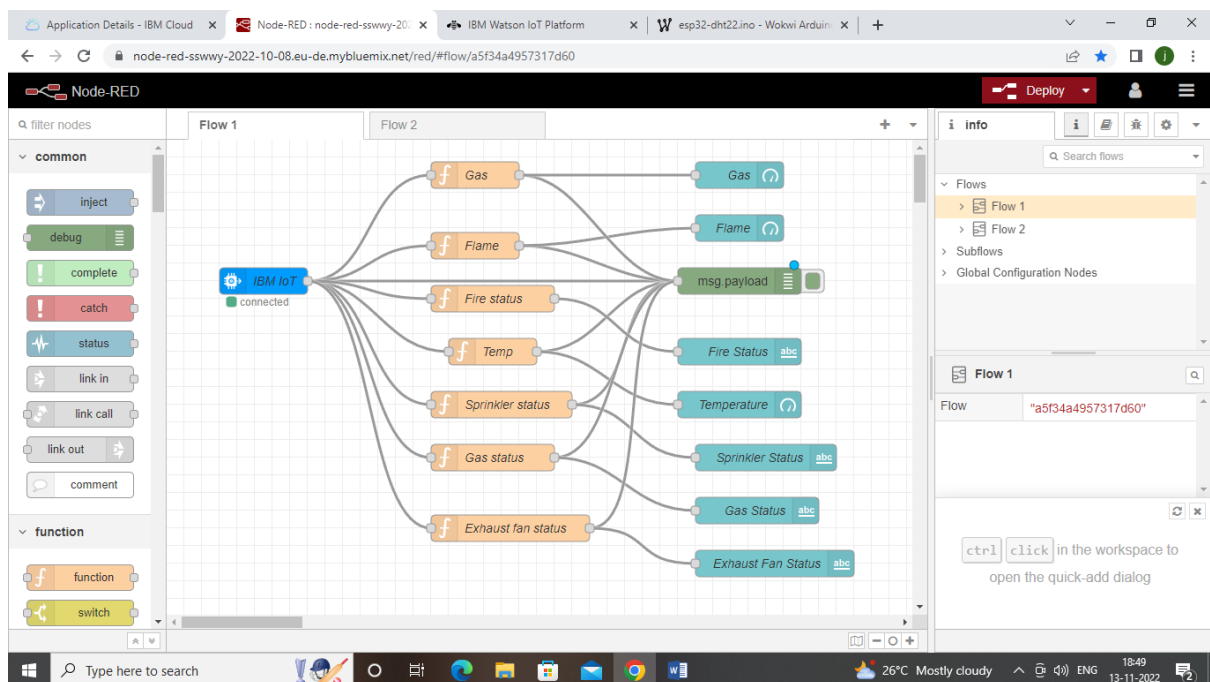
WOKWI OUTPUT:



NODE-RED :



NODE-RED CONNECTION:



OUTPUT:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with an 'IBM IoT' node connected to several function nodes: 'Gas', 'Flame', 'Fire status', 'Temp', 'Sprinkler status', 'Gas status', and 'Exhaust fan status'. These function nodes are connected to corresponding output nodes: 'Gas', 'Flame', 'msg.payload', 'Fire Status', 'Temperature', 'Sprinkler Status', 'Gas Status', and 'Exhaust Fan Status'. The 'msg.payload' node is connected to a 'debug' node. The left sidebar shows a list of nodes, and the right sidebar shows the debug console with logs for the flow.

Debug Console Logs:

```
11/12/2022, 11:36:24 PM node: f2f2649a.0d0d98  
iot-2/type/1234d/123_3/evt/eventtest/fmt/json :  
msg.payload: string[23]  
"Gas Leakage is Detected"  
11/12/2022, 11:36:24 PM node: f2f2649a.0d0d98  
iot-2/type/1234d/123_4/evt/eventtest/fmt/json :  
msg.payload: Object  
{ gas: 998, temp: 114, flame: 729,  
firestatus: "No Fire",  
sprinklerstatus: "Not Working" }  
11/12/2022, 11:36:24 PM node: f2f2649a.0d0d98  
iot-2/type/1234d/123456/evt/eventtest/fmt/json :  
msg.payload: Object  
{ gas: 629, temp: 229, flame: 565,  
firestatus: "No Fire",  
sprinklerstatus: "Not Working" }  
11/12/2022, 11:36:24 PM node: f2f2649a.0d0d98  
iot-2/type/1234d/123_4/evt/eventtest/fmt/json :  
msg.payload: number  
114  
11/12/2022, 11:36:24 PM node: f2f2649a.0d0d98  
iot-2/type/1234d/123_4/evt/eventtest/fmt/json :  
msg.payload: string[11]  
"Not Working"
```

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with an 'IBM IoT' node connected to several function nodes: 'Gas', 'Flame', 'Fire status', 'Temp', 'Sprinkler status', 'Gas status', and 'Exhaust fan status'. These function nodes are connected to corresponding output nodes: 'Gas', 'Flame', 'msg.payload', 'Fire Status', 'Temperature', 'Sprinkler Status', 'Gas Status', and 'Exhaust Fan Status'. The 'msg.payload' node is connected to a 'debug' node. The left sidebar shows a list of nodes, and the right sidebar shows the debug console with logs for the flow.

Debug Console Logs:

```
msg.payload: number  
321  
11/13/2022, 6:51:08 PM node: f2f2649a.0d0d98  
iot-2/type/1234d/123456/evt/eventtest/fmt/json :  
msg.payload: string[11]  
"Not Working"  
11/13/2022, 6:51:08 PM node: f2f2649a.0d0d98  
iot-2/type/1234d/123456/evt/eventtest/fmt/json :  
msg.payload: string[7]  
"Working"  
11/13/2022, 6:51:08 PM node: f2f2649a.0d0d98  
iot-2/type/1234d/123456/evt/eventtest/fmt/json :  
msg.payload: number  
848  
11/13/2022, 6:51:08 PM node: f2f2649a.0d0d98  
iot-2/type/1234d/123456/evt/eventtest/fmt/json :  
msg.payload: number  
697  
11/13/2022, 6:51:08 PM node: f2f2649a.0d0d98  
iot-2/type/1234d/123456/evt/eventtest/fmt/json :  
msg.payload: string[23]  
"Gas Leakage is Detected"
```