|  |  |
| --- | --- |
| TEAM ID | PNT2022TMID50655 |
| PROJECT NAME | **Industry-specific intelligent fire management system** |

Python Code :

#include <time.h>

#include <WiFi.h>

#include <PubSubClient.h>

#define ORG "7iw80s"

#define DEVICE\_TYPE "nodemcu"

#define DEVICE\_ID "44444"

#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);

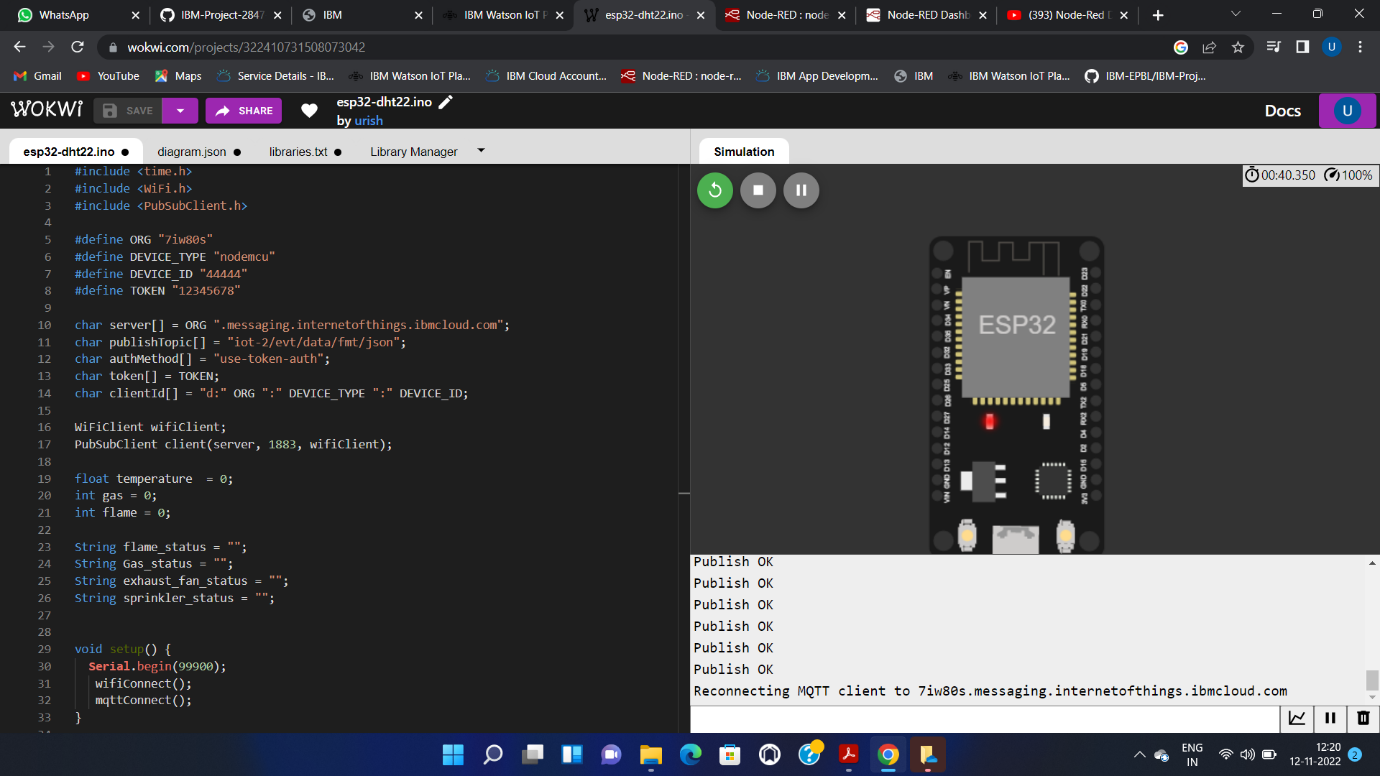
}

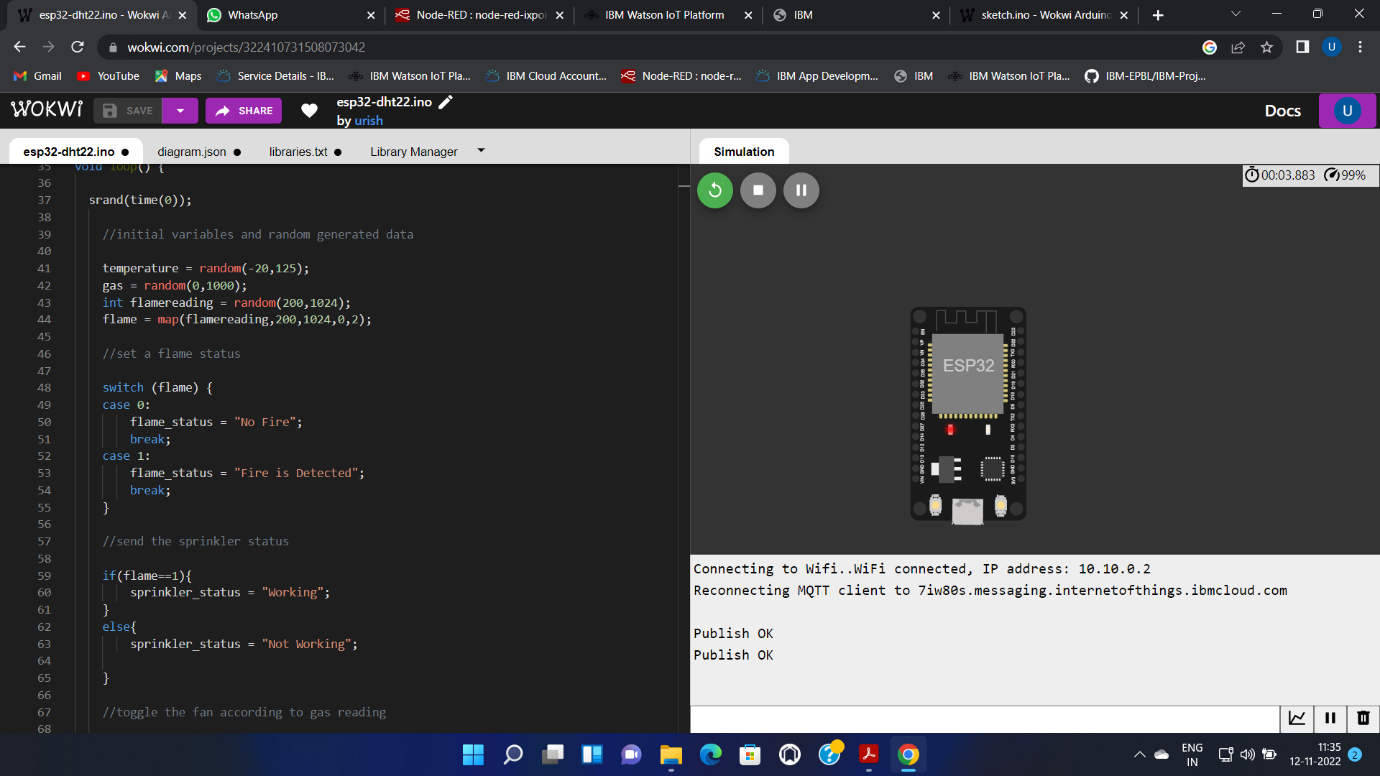
Serial.println();

}

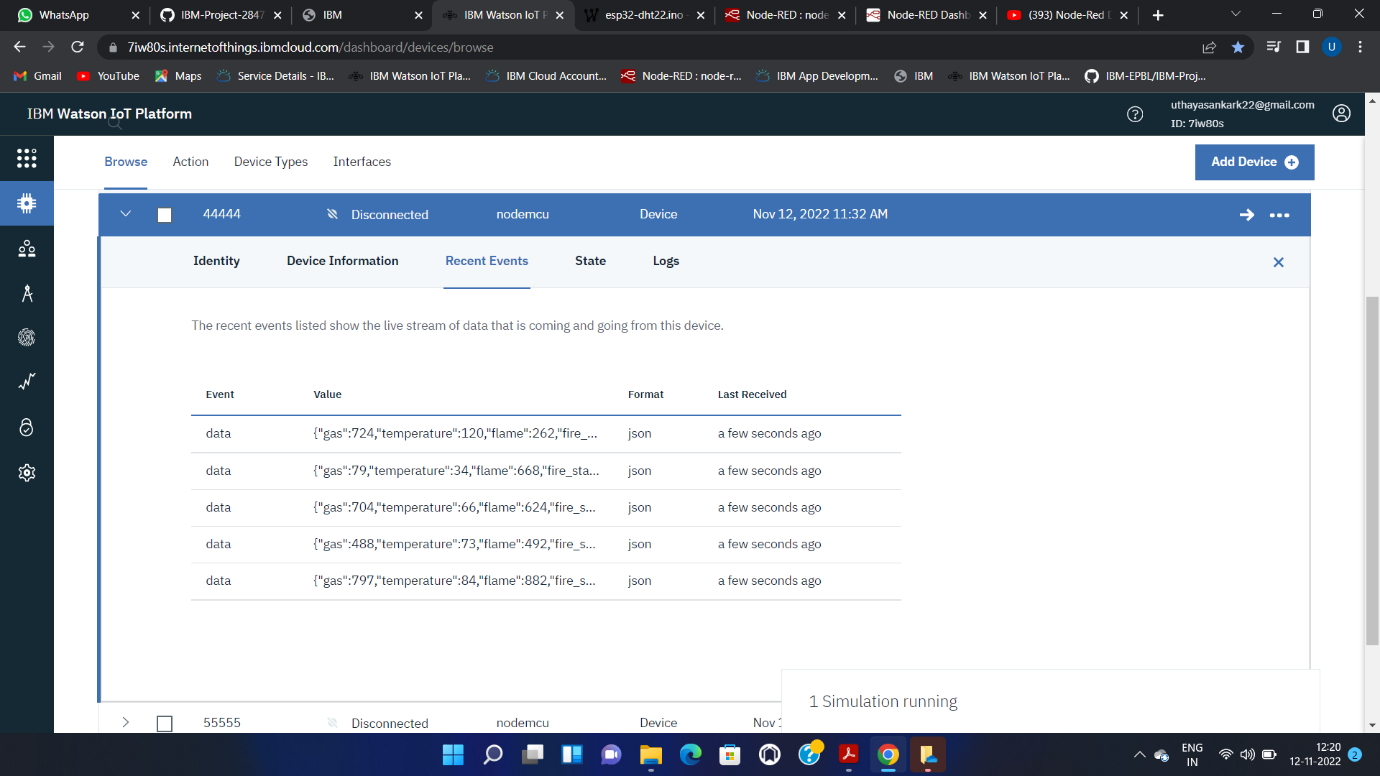
}

OUTPUT Screenshot :

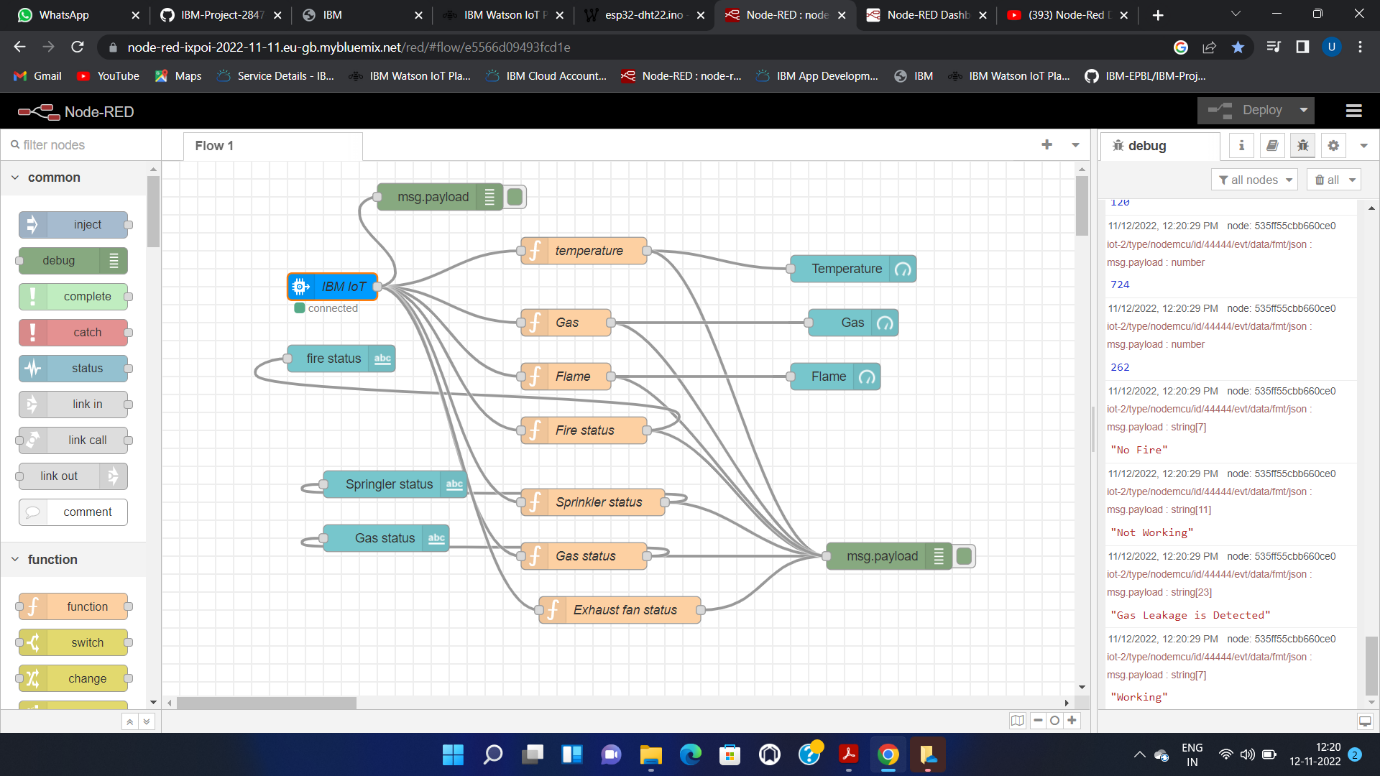




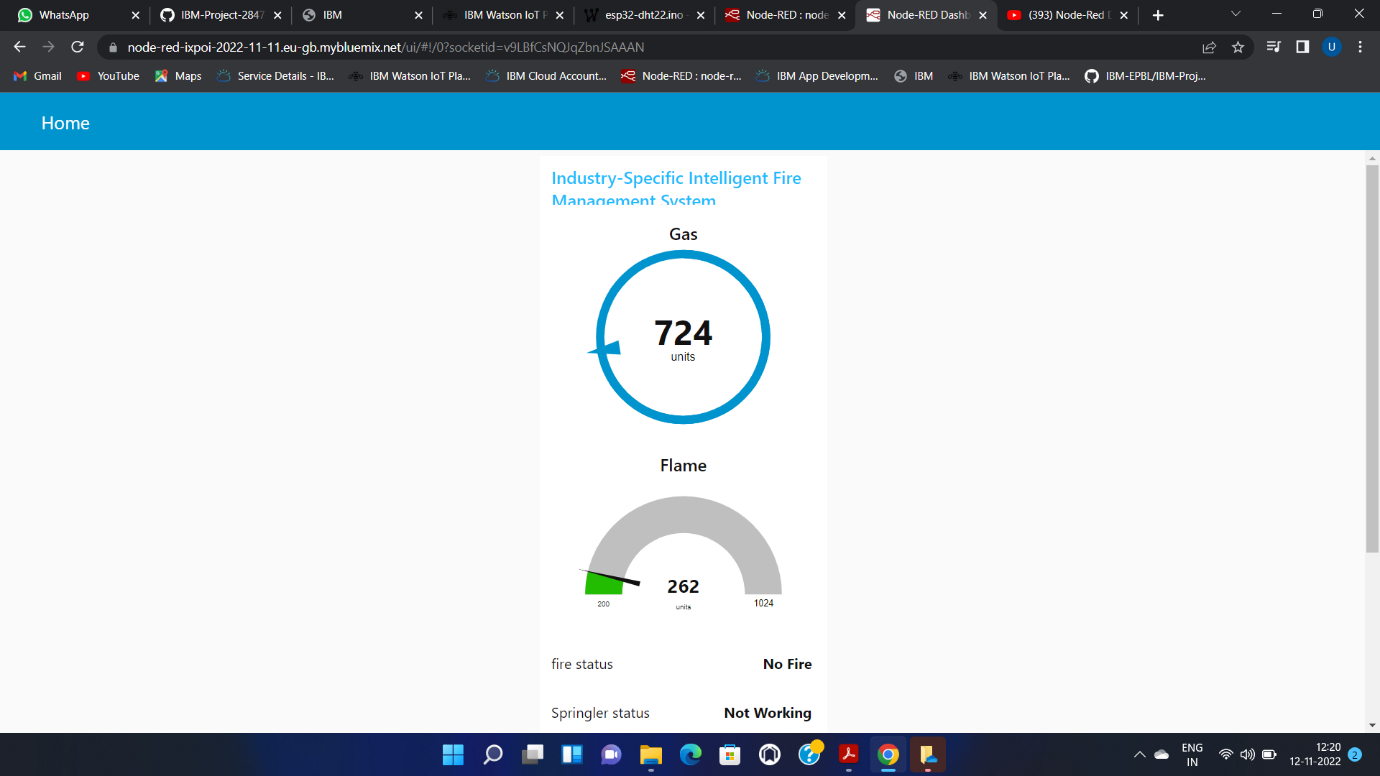
IBM Waston to IOT Platform :

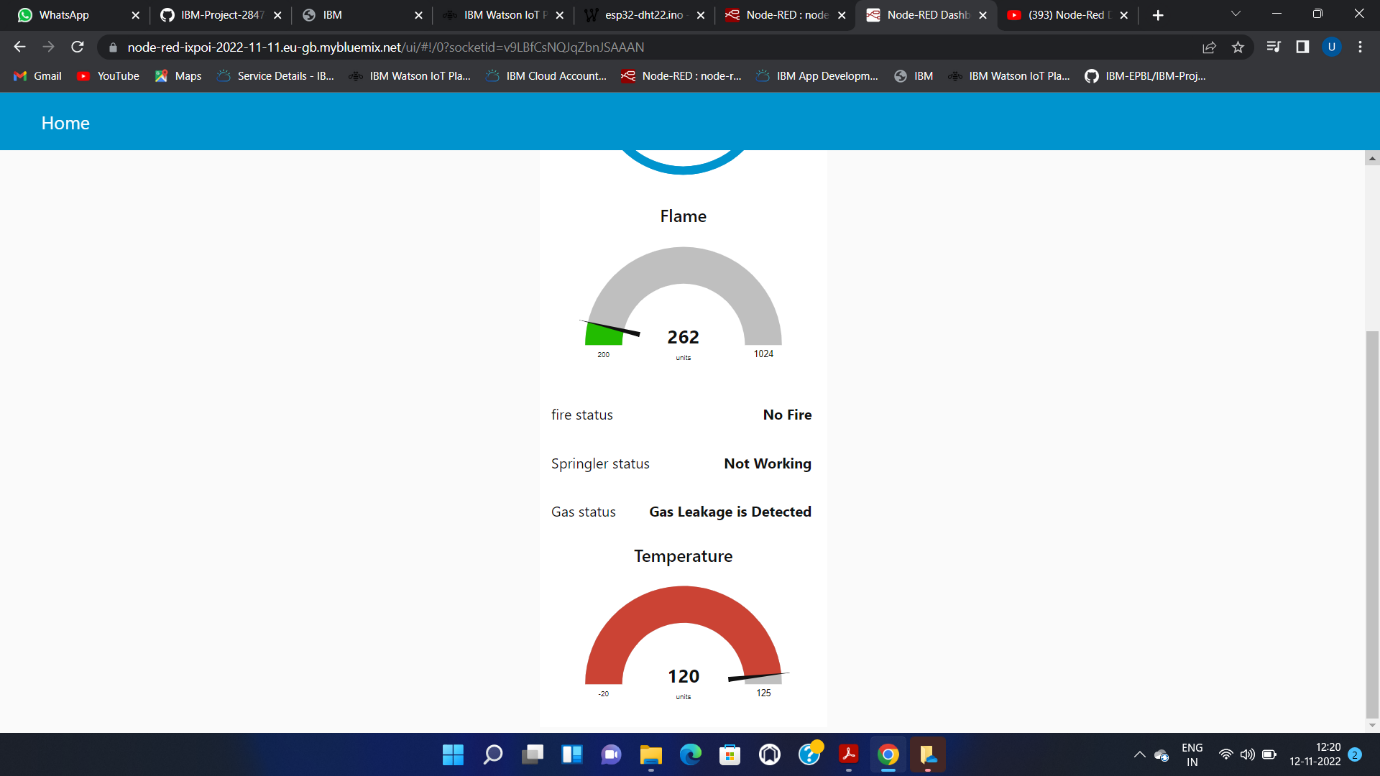


Web application Using Node-Red :



Node-Red Dashbord Status :





Suceessfully get the status for Flame, Fire, Gas , Temperture in Node-red Dashboard Using Node-Red Platform..