|  |  |
| --- | --- |
| Date | 10 November 2022 |
| Team ID | PNT2022TMID20469 |
| Project Title | Industry-Specific Intelligent Fire Management  System |

# Display the temperature values:

Submitted by

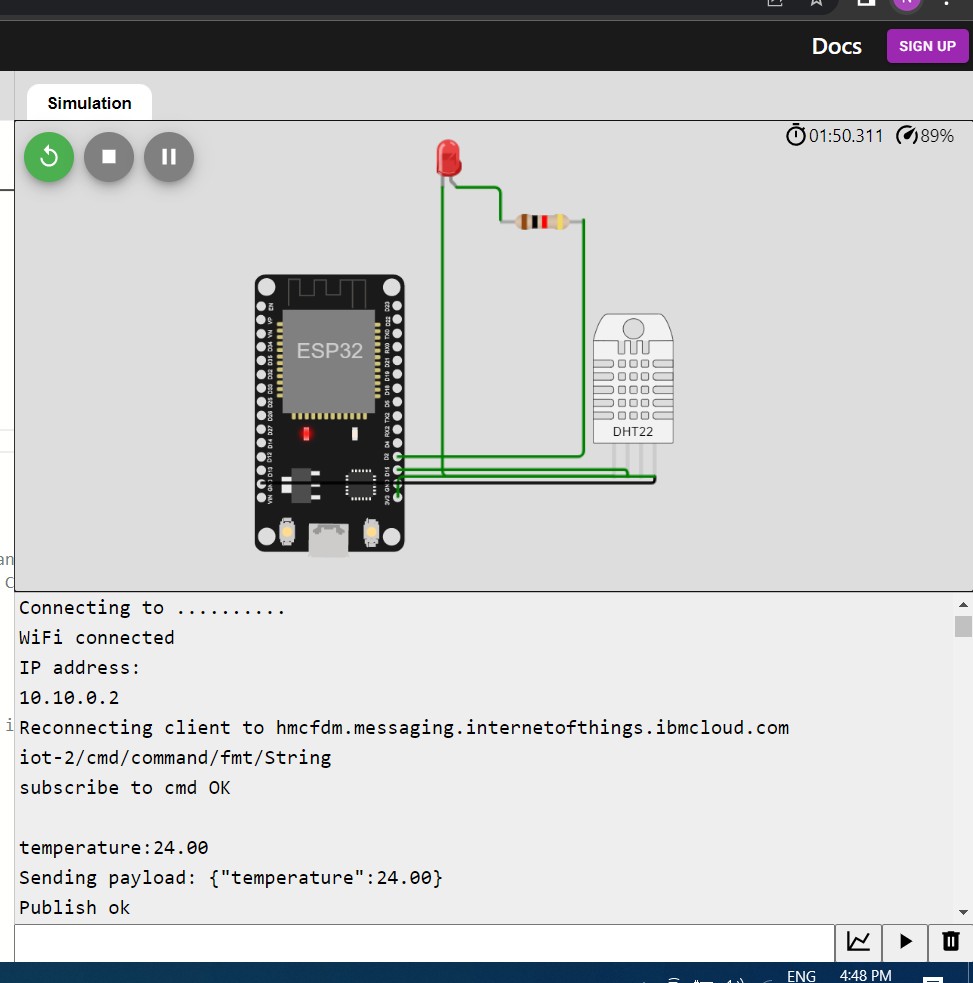
Manoj K – 49621911082

Pradeepa S – 49621911096

Mugesh S – 49621911033

Jayasri M - 49621911068

Wokwi link: <https://wokwi.com/projects/348683544624628306>



CODING:

#include <WiFi.h>//library for wifi #include <PubSubClient.h>//library for MQtt #include "DHT.h"// Library for dht11

#define DHTPIN 15 // what pin we're connected to #define DHTTYPE DHT22 // define type of sensor DHT 11 #define LED 2

DHT dht (DHTPIN, DHTTYPE);

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

//-------credentials of IBM Accounts------

#define ORG "hmcfdm"//IBM ORGANITION ID

#define DEVICE\_TYPE "IOT\_FIRE"//Device type mentioned in ibm watson IOT Platform #define DEVICE\_ID "261021"//Device ID mentioned in ibm watson IOT Platform #define TOKEN "1911063abcdefgh" //Token

String data3; float t;

//-------- Customise the above values --------

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

char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type ofevent perform and format in which data to be send

char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENTcommand type AND COMMAND IS TEST OF FORMAT STRING

char authMethod[] = "use-token-auth";// authentication method char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;//client id

// -

WiFiClient wifiClient; // creating the instance for wificlient

PubSubClient client(server, 1883, callback ,wifiClient); //calling thepredefined client id by passing parameter like server id,portand wificredential

void setup()// configureing the ESP32

{

**Serial**.begin(115200); dht.begin(); pinMode(LED,OUTPUT); delay(10); **Serial**.println(); wificonnect(); mqttconnect();

}

void loop()// Recursive Function

{

t = dht.readTemperature();

**Serial**.print("temperature:");

**Serial**.println(t); PublishData(t); delay(1000);

if (!client.loop()) { mqttconnect();

}

}

/\*.....................................retrieving to

Cloud. \*/

void PublishData(float temp) { mqttconnect();//function call for connecting to ibm

/\*

creating the String in in form JSon to update the data to ibm cloud

\*/

String payload = "{\"temperature\":"; payload += temp;

payload += "}"; **Serial**.print("Sending payload: "); **Serial**.println(payload);

if (client.publish(publishTopic, (char\*) payload.c\_str())) { **Serial**.println("Publish ok");// if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish failed

} else {

**Serial**.println("Publish failed");

}

}

void mqttconnect() {

if (!client.connected()) { **Serial**.print("Reconnecting client to "); **Serial**.println(server);

while (!!!client.connect(clientId, authMethod, token)) {

**Serial**.print("."); delay(500);

}

initManagedDevice();

**Serial**.println();

}

}

void wificonnect() //function defination for wificonnect

{

**Serial**.println(); **Serial**.print("Connecting to ");

WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials toestablish the connection

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++) {

//Serial.print((char)payload[i]); data3 += (char)payload[i];

}

**Serial**.println("data: "+ data3); if(data3=="lighton")

{

**Serial**.println(data3); digitalWrite(LED,HIGH);

}

else

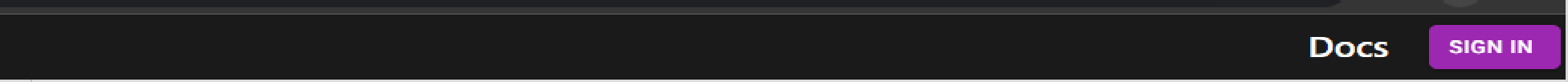
{

**Serial**.println(data3); digitalWrite(LED,LOW);

}

data3="";

}



**Simulation**

02:5 3.565 ^j 99%

Send i ng payloa d : { "I emper a tu ne“ : 24. 66}

Publish ok

I emper aI u ne : 24. 66

Send i ng payloa d : { "I emper a tu re“ : 24. 66}

Publish ok temperature:z4.00

Sending payload: ("temperature”:24.00} Publish ok

I emper'a I u re : 24. 66

Send i ng payloa d : { "I emper a tu ne“ : 24. 66}

Publish ok

I emp e aI u ne : 24. 66

Sen di ng p ay 1oa d : { "I emp e a tu r e“ : 24. 66}

Publish ok temperature:24.00

Sending payload: ("temperature”:24.00} Publish ok

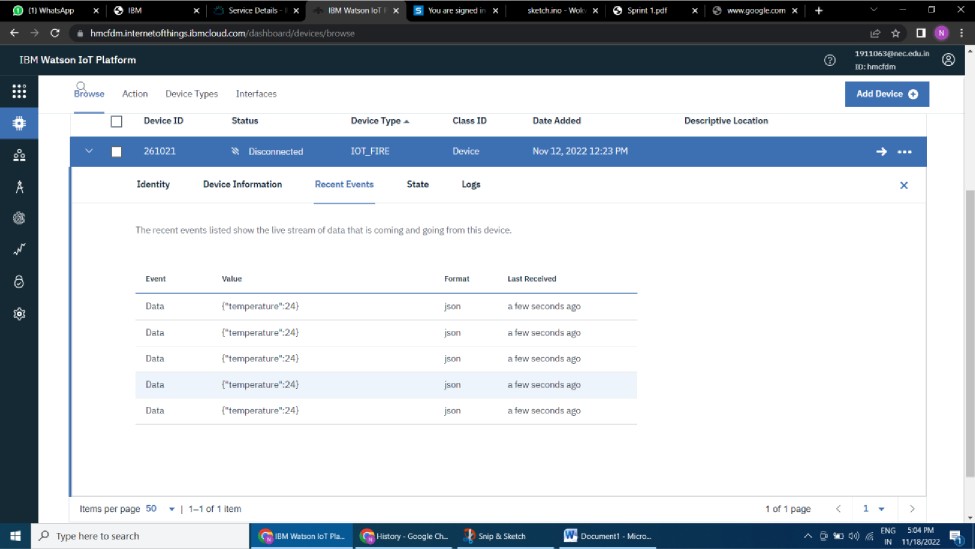
I emp e aI u r e : 24. 66

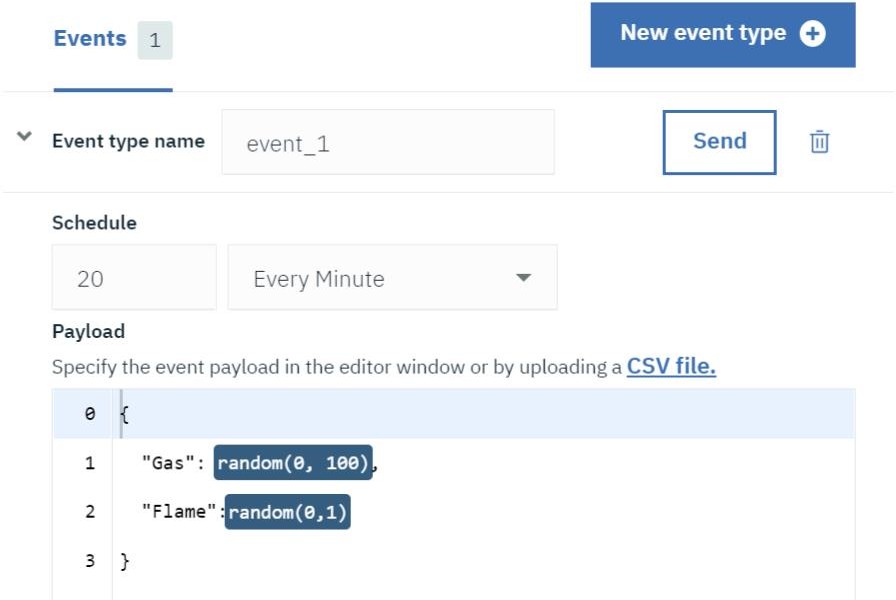
Sen di ng p ay 1oa d : { "I emp e r‘a tu r e“ : 24. 66}

Publish ok

I emp e aI u ne : 24. 66

Sending payload : ("temperature”:24.00} Publish ok

I emper aI u ne : 24. 66



**Displaying gas sensor & flame sensor values:**

