SPRINT-1

DATE	29 october 2022
TEAM ID	**PNT2022TMID36746**
PROJECT NAME	SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

PROGRAM:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 5     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr
of dht connected

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

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

#define ORG "qguokr"//IBM ORGANITION ID
#define DEVICE_TYPE "ibm"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "wikki14"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "123456789"     //Token
String data3;
float h, t;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server
Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of
event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT
```

```
command 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 the
predefined client id by passing parameter like server id,portand
wificredential


void setup()// configureing the ESP32
{
  Serial.begin(115200);
  dht.begin();
  pinMode(33, INPUT); //North
  pinMode(25, INPUT); // South
  pinMode(26, INPUT); // East
  pinMode(27, INPUT); // West
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

int n, s, e, w;

void loop()// Recursive Function
{

  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.print("temp:");
  Serial.println(t);
```

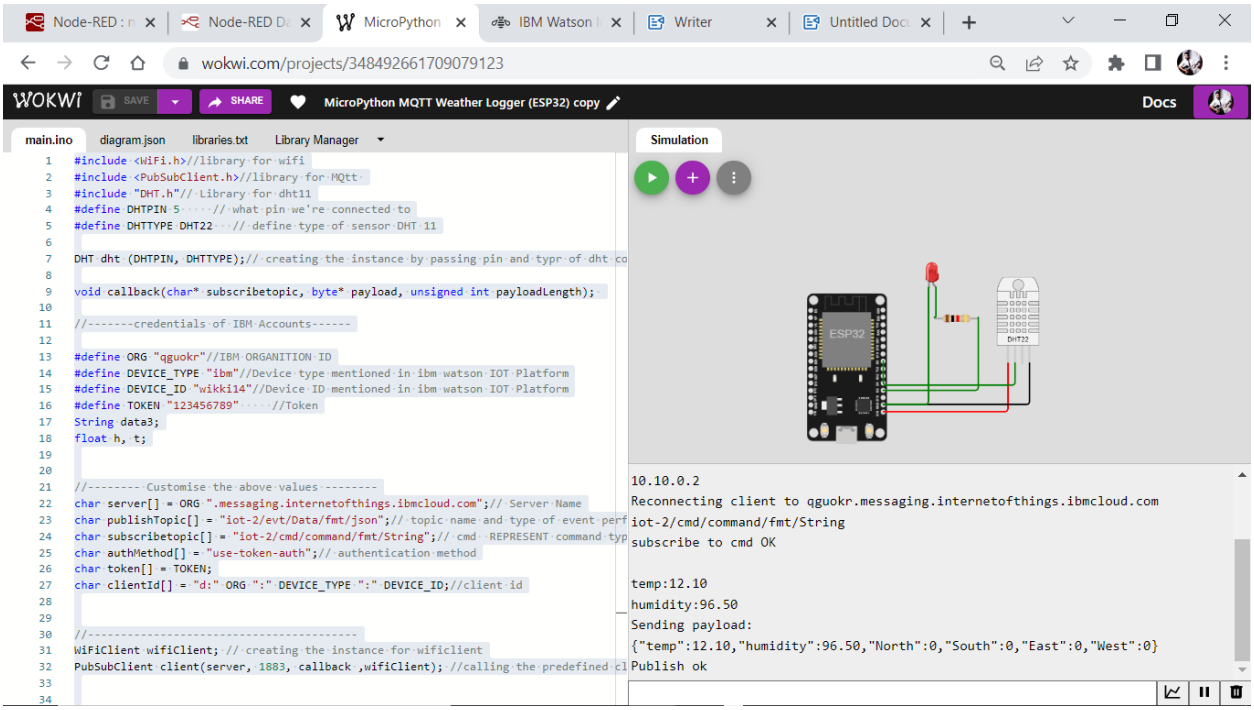
```
  Serial.print("humidity:");
  Serial.println(h);

  n = digitalRead(33);
  s = digitalRead(25);
  e = digitalRead(26);
  w = digitalRead(27);

  PublishData(t, h, n, s, e, w);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}



/*.....................................retrieving to
Cloud.............................*/

void PublishData(float temp, float humid, int n, int s, int e, int w) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"temp\":";
  payload += temp;
  payload += "," "\"humidity\":";
  payload += humid;
  payload += "," "\"North\":";
  payload += n;
  payload += "," "\"South\":";
  payload += s;
  payload += "," "\"East\":";
  payload += e;
  payload += "," "\"West\":";
```

```
  payload += w;
  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 ");
```

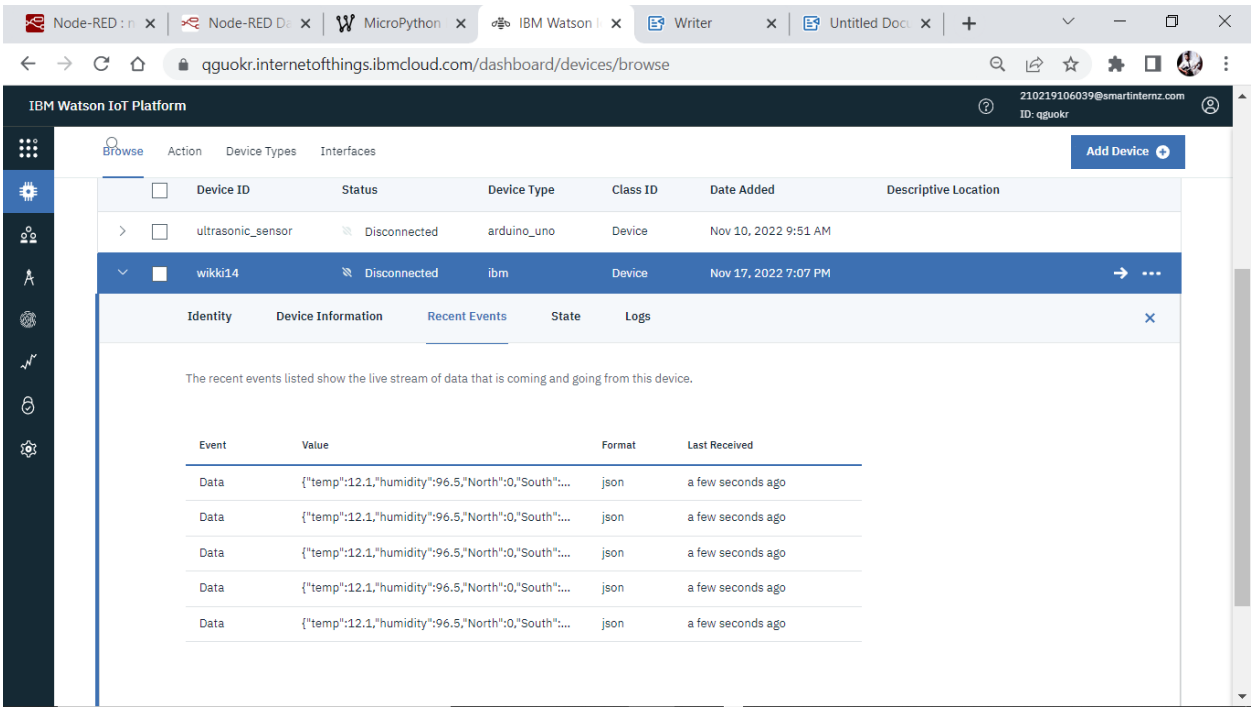
```
  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to
establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

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="";
}
```

REF :https://wokwi.com/projects/348492661709079123



WATSON OUTPUT:


Node-RED : n
Node-RED De
MicroPython
IBM Watson I
Writer
Untitled Doc
qguokr.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
210219106039@smartinternz.com
ID: qguokr
Browse
Action
Device Types
Interfaces
Add Device
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
ultrasonic_sensor
Disconnected
arduino_uno
Device
Nov 10, 2022 9:51 AM
wikki14
Disconnected
ibm
Device
Nov 17, 2022 7:07 PM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
Data
{"temp":12.1,"humidity":96.5,"North":0,"South":...
json
a few seconds ago
Data
{"temp":12.1,"humidity":96.5,"North":0,"South":...
json
a few seconds ago
Data
{"temp":12.1,"humidity":96.5,"North":0,"South":...
json
a few seconds ago
Data
{"temp":12.1,"humidity":96.5,"North":0,"South":...
json
a few seconds ago
Data
{"temp":12.1,"humidity":96.5,"North":0,"South":...
json
a few seconds ago