**TEAM ID : PNT2022TMID02412**

#include <WiFi.h>

#include <PubSubClient.h>

#include "DHT.h"

#define DHTPIN 15

#define DHTTYPE DHT22

#define LED 2

DHT dht (DHTPIN, DHTTYPE);  //Creating the instances for DHT22

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

payloadLength);

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

#define ORG "nk0g35"//IBM ORGANITION ID

#define DEVICE\_TYPE "DHT11"//Device type mentioned in ibm watson IOT Platform

#define DEVICE\_ID "GK"//Device ID mentioned in ibm watson IOT Platform

#define TOKEN "PkQA1NO4!mrNh(1(Qd" //Token

float h, t; // variable for Humidity, Temperature

long gas\_random; // variable for random generator which is considered to be output of gas sensor

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

char publishTopic[] = "iot-2/evt/Data/fmt/json";

char subscribetopic[] = "iot-2/cmd/test/fmt/String";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

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

String data3;

WiFiClient wifiClient;

PubSubClient client(server, 1883, callback ,wifiClient);

void setup()

 {

**Serial**.begin(115200);

  dht.begin();

  pinMode(LED,OUTPUT);

  delay(10);

**Serial**.println();

  wificonnect();

  mqttconnect();

 }

void loop()

 {

  h = dht.readHumidity();

  t = dht.readTemperature();

  gas\_random = random(0,100);

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

**Serial**.println(t);

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

**Serial**.println(h);

**Serial**.print("Gas PPM:");

**Serial**.println(gas\_random);

  PublishData(t, h, gas\_random);

  delay(1000);

  if (!client.loop())

   {

    mqttconnect();

   }

 }

//--------Publishing the data to IBM Watson IOT platform--------

void PublishData(float temp, float humid, long gas)

 {

  mqttconnect();

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

  payload += temp;

  payload += "," "\"humidity\":";

  payload += humid;

  if (gas>=30)

   {     // Checking of gas leakage beyond the threshold (here we considered the safelimit as below 30 ppm)

     payload += "," "\"Alert !!! Detected gas PPM\":";

     payload += gas;

   }

  payload += "}";

**Serial**.print("Sending payload: ");

**Serial**.println(payload);

//-------Checking of the connection between Wokwi and IBM Watson IOT platform---------

  if (client.publish(publishTopic, (char\*) payload.c\_str()))

   {

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

   }

  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()

 {

**Serial**.println();

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

  WiFi.begin("Wokwi-GUEST", "", 6);

  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);

data3="";

}

**Output:**







