

SPRINT-4

DATE	18 NOVEMBER 2022
TEAM ID	PNT2022TMID40922
PROJECT NAME	HAZARDOUS AREA MONITORING FOR INDUSTRIAL PLANT POWERED BY IoT

WOKWI CODE :

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>
#include "DHT.h"// Library for dht11
#define DHTPIN 15 // what pin we're connected to
#define DHTTYPE DHT22

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 "aqudbz"//IBM ORGANITION ID
#define DEVICE_TYPE "NodeMCU"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "EON8Q6-UN@GTJ&zH-Q" //Token
String data3;
float Humidity, Temp;

//----- 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();
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

```

```
void loop()// Recursive Function
```

```

{

  Humidity = dht.readHumidity();

```

```
Temp = dht.readTemperature();  
Serial.print("Temp:");  
Serial.println(Temp);  
Serial.print("Humidity:");  
Serial.println(Humidity);
```

```
PublishData(Temp,Humidity);  
delay(1000);  
if (!client.loop()) {  
    mqttconnect();  
}  
}
```

/.....retrieving to Cloud /

```
void PublishData(float Temp, float Humidity) {  
    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 += Humidity;  
    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];
}
}

```

WOKWI OUTPUT :

The screenshot displays the Wokwi IDE interface. On the left, the code editor shows a C++ program for an ESP32. The code includes a setup function for WiFi and a callback function for a topic. The main loop connects to a WiFi network and prints the IP address. The callback function processes a payload and appends it to a data string.

```

106 Serial.print("Connecting to ");
107
108 WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
109 while (WiFi.status() != WL_CONNECTED) {
110     delay(500);
111     Serial.print(".");
112 }
113 Serial.println("");
114 Serial.println("WiFi connected");
115 Serial.println("IP address: ");
116 Serial.println(WiFi.localIP());
117 }
118
119 void initManagedDevice() {
120     if (client.subscribe(subscribetopic)) {
121         Serial.println(subscribetopic);
122         Serial.println("subscribe to cmd OK");
123     } else {
124         Serial.println("subscribe to cmd FAILED");
125     }
126 }
127
128 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
129 {
130
131     Serial.print("callback invoked for topic: ");
132     Serial.println(subscribetopic);
133     for (int i = 0; i < payloadLength; i++) {
134         //Serial.print((char)payload[i]);
135         data3 += (char)payload[i];
136     }
137 }
138 }

```

On the right, the simulation window shows a visual representation of the ESP32 board connected to a WiFi module and a DHT22 sensor. Below the simulation, the output console displays the following text:

```

Connecting to .....
WiFi connected
IP address:
10.10.0.2
Reconnecting client to aqdbz.messaging.internetofthings.ibmcloud.com

```

LINK : <https://wokwi.com/projects/348655340794937938>

IBM WATSON PLATFORM :

