

## Sprint - 4

Team Id	PNT2022TMID23518
Title	Hazardous Area Monitoring for Industrial Plant using IoT

**Using the DHT sensor to read the real world values and push it to IBM IoT platform**

### Program

```
#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); // creating the instance by passing pin and type of dht connected

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

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

#define ORG "4wj0mx" //IBM ORGANIZATION ID
#define DEVICE_TYPE "ESP_Controller" //Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "Esp32_sensor" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "1234567890"
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 subscribtopic[] = "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 wifi client
PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client id by passing parameter like server
```

```

void setup()// configuring the ESP32
{
  Serial.begin(115200);
  dht.begin();
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

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

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

/*.....retrieving to cloud.....*/

void PublishData(float temp, float humid) {
  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 += "," "\"Humid\":";
  payload += humid;
  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 monitt
  } 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 ");

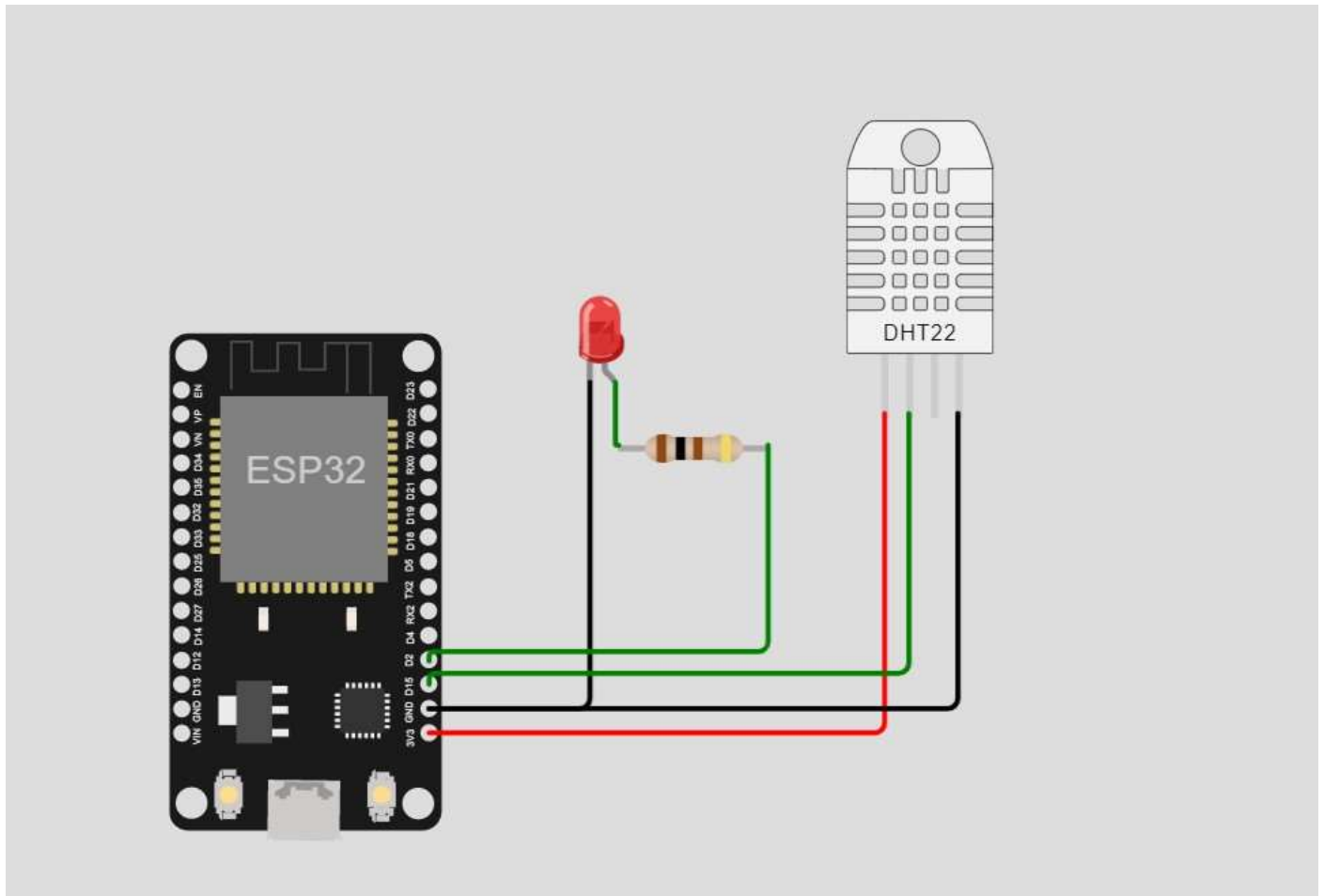
  WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());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

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="";
}

```

## Circuit Diagram



## Simulation Results

```
Reconnecting client to 4wj0mx.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK
```

```
temp:74.30
Humid:65.50
Sending payload: {"temp":74.30,"Humid":65.50}
Publish ok
temp:74.30
Humid:65.50
Sending payload: {"temp":74.30,"Humid":65.50}
Publish ok
```

**Pushed to IBM cloud**

Esp32\_sensor

Connected

ESP\_Controller

Device

18 Nov 2022 6:33 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":15.3,"Humid":65.5}	json	a few seconds ago
Data	{"temp":74.3,"Humid":65.5}	json	a few seconds ago
Data	{"temp":74.3,"Humid":65.5}	json	a few seconds ago
Data	{"temp":74.3,"Humid":65.5}	json	a few seconds ago
Data	{"temp":74.3,"Humid":65.5}	json	a few seconds ago