

# SPRINT - 03

Date :	12 November 2022
Team ID :	PNT022TMID37754
Project Name	Industry-specific intelligent fire management System

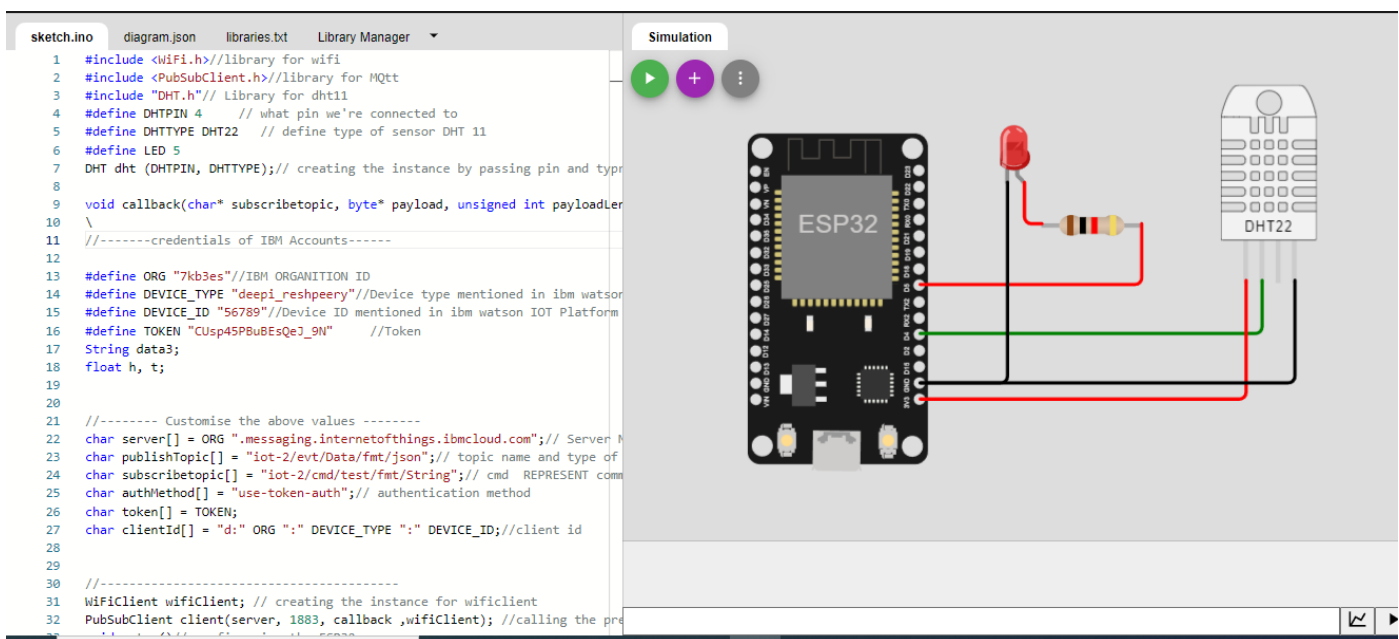
## SPRINT GOAL:

Integrate the hardware to be able to access the cloud functions and provide inputs to the same.

## POGRAM 01:

**AIM:** To find the Temperature and Humidity DHT22 and ESP32

**PLATFORM:** WOKWI



IBM Watson IoT Platform

410119106009@smartinternz.com  
ID: 7kb3es

Browse

Action

Device Types

Interfaces

Add Device

>

4321

Disconnected

assign4

Device

18 Nov 2022 02:07

▼

56789

Connected

deepi\_resheery

Device

22 Oct 2022 05:28

→ ...

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	{"temperature":6.1,"humidity":81}	json	a few seconds ago
Data	{"temperature":6.1,"humidity":81}	json	a few seconds ago
Data	{"temperature":26.2,"humidity":64.5}	json	a few seconds ago
Data	{"temperature":26.2,"humidity":64.5}	json	a few seconds ago
Data	{"temperature":-9.2,"humidity":64.5}	json	a few seconds ago

**PYTHON CODE:**

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#include "DHT.h" // Library for dht11
#define DHTPIN 4 // what pin we're connected to
```

```

#define DHTTYPE DHT22 // define type of sensor DHT 11
#define LED 5
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 "60ys35"//IBM ORGANITION ID
#define DEVICE_TYPE "iot"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "4321"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "987654321" //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/test/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(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}
void loop()// Recursive Function
{

  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.print("temperature:");
  Serial.println(t);
  Serial.print("Humidity:");
  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 = "{\"temperature\":";  
  payload += temp;  
  payload += "," " \"humidity\":";  
  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 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="";
}

```

### **Output link :**

<https://wokwi.com/projects/348728594810274388>

By using this Wokwi we determined the Temperature and Humidity for better road safety.

**THANK YOU...!!**