

SPRINT 4

Team ID	PNT2022TMID53630
Project name	Smart Farmer - IoT Enabled Smart Farming Application

SOURCE CODE:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#include "DHTesp.h"
const int DHT_PIN=15;
DHTesp dhtsensor;
int LED=9;
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);

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

#define ORG "w79t1i" //IBM ORGANITION ID
#define DEVICE_TYPE "Newdevice" //Device type mentioned in ibm watson
IOT Platform
#define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "ibm12345678" //Token

String data3;
String message;

//----- 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, port and wifi credential
void setup() // configuring the ESP32
{
  Serial.begin(115200);
  dhtsensor.setup(DHT_PIN, DHTesp::DHT22);
  Serial.println();
  wifiConnect();
  mqttConnect();
}

void loop() // Recursive Function
{
  TempAndHumidity data = dhtsensor.getTempAndHumidity();
  int temp = data.temperature;
  int humid = data.humidity;
  int moisture = random(0, 100);
  Serial.println("temperature" + String(temp) + "c");
  Serial.println("humidity" + String(humid) + "%");
  Serial.println("moisture" + String(moisture) + "%");
  delay(1000);
  PublishData(temp, humid, moisture);
  if (!client.loop()) {
    mqttConnect();
  }
}

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

void PublishData(int d, int a, int b) {
  mqttConnect(); //function call for connecting to IBM
  /*
   creating the String in form of JSON to update the data to IBM cloud
  */
  String payload = "{\"temperature\"":";
  payload += d;
  payload += "}";
  payload += ", \"humidity\"":";
  payload += a;
}

```

```
payload += "}";  
payload += ", \"{\\\"soilmoisture\\\":}";  
payload += b;  
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=="motoron")
    {
        Serial.println(data3);
        digitalWrite(LED,HIGH);

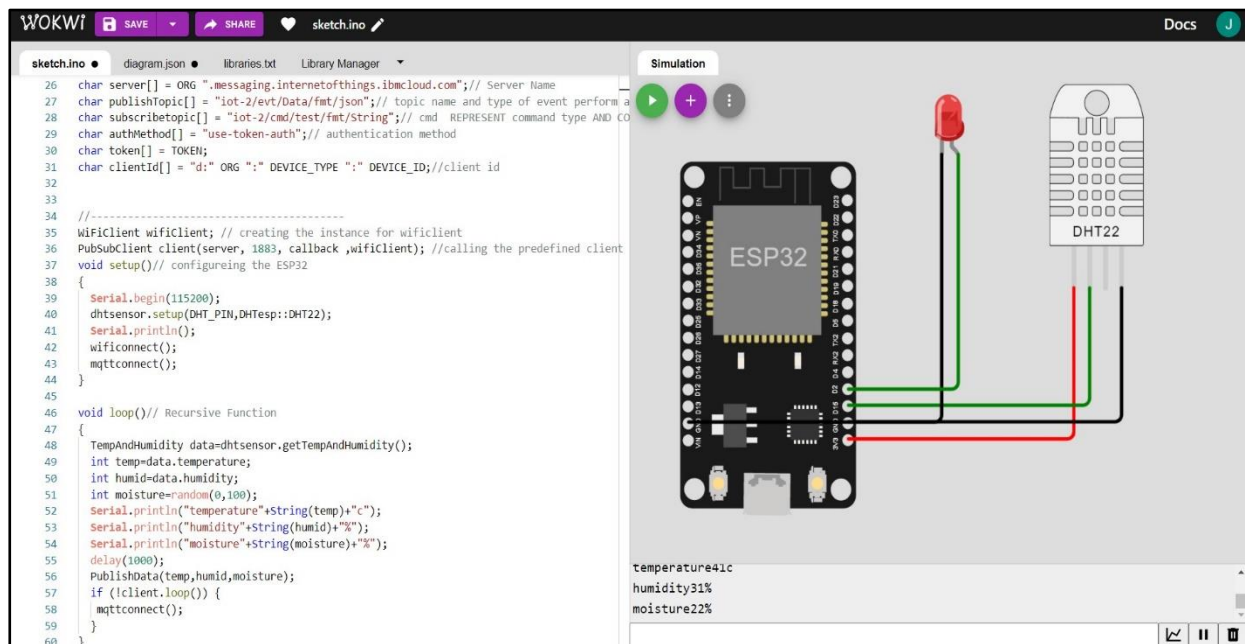
    }

    else
    {
        Serial.println(data3);
        digitalWrite(LED,LOW);

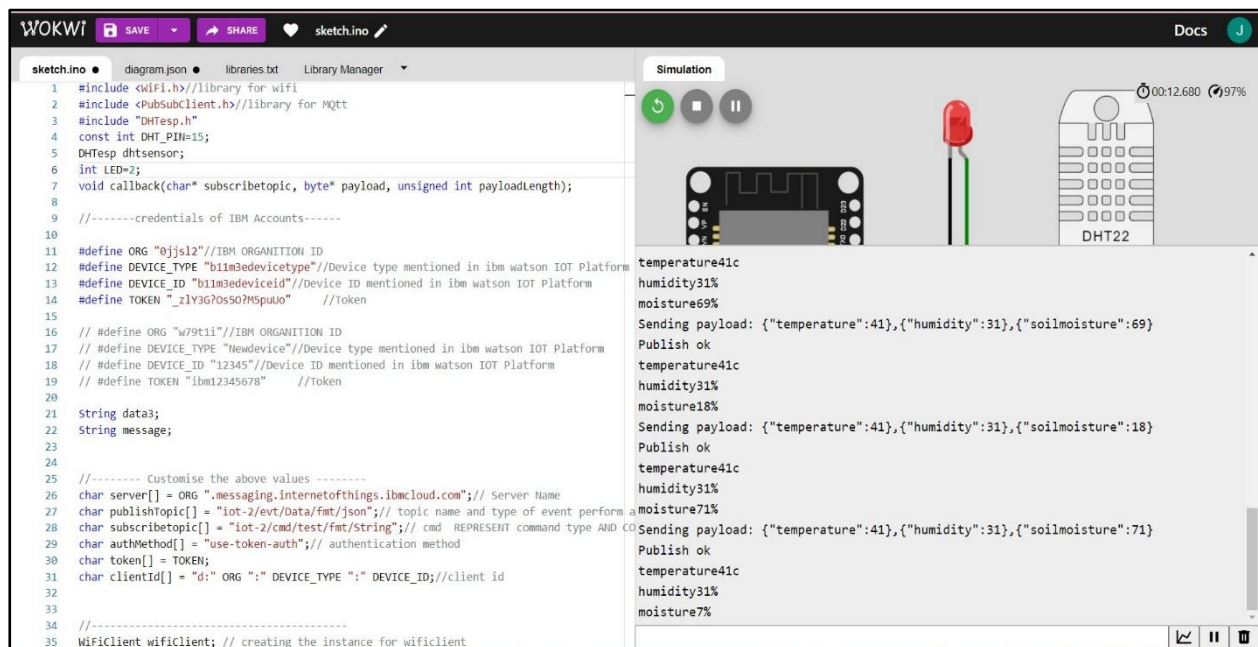
    }
    data3="";
}

```

CIRCUIT DIAGRAM:



SENDING DATA TO IBM WATSON CLOUD:



Reflecting the values of temperature , humidity and soil moisture values in gauges

Revilla Jyosthna:



Pannave K



Iyswarya S:



Pritha R:



MIT APP for user

Revilla Jyosthna:

The screenshot displays a mobile application interface with a green header bar at the top showing the time '10:46 PM' and status icons for signal, Wi-Fi, and battery (35%). Below the header is a grey bar with the title 'Monitoring and Controlling'. The main content area has a light grey background and contains three rows of data, each with a label on the left and a white input field on the right. The first row shows 'Temperature' with the value '49'. The second row shows 'Humidity' with the value '89'. The third row shows 'Soil Moisture' with the value '52'. Below these rows are two grey buttons labeled 'Motor On' and 'Motor off'. At the bottom of the screen is a white bar with three navigation icons: a square, a circle, and a triangle.

Temperature	49
Humidity	89
Soil Moisture	52

Motor On Motor off

Pannave K:

Screen1

Temperature 69

Humidity 65

Soil Moisture 54

Motor ON Motor OFF

Iyswarya S:

The screenshot shows a mobile application interface with a green status bar at the top displaying the time 10:46 PM, signal strength, Wi-Fi, and a 35% battery level. Below the status bar is a grey header with the text "Monitoring and Controlling". The main content area has a light grey background and contains three rows of data: "Temperature" with a value of 49, "Humidity" with a value of 89, and "Soil Moisture" with a value of 52. Each value is displayed in a white rectangular box. At the bottom of the main content area, there are two grey buttons labeled "Motor On" and "Motor off". The bottom of the screen features a white navigation bar with three icons: a square, a circle, and a triangle.

Monitoring and Controlling	
Temperature	49
Humidity	89
Soil Moisture	52

Motor On Motor off

Pritha R:

