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

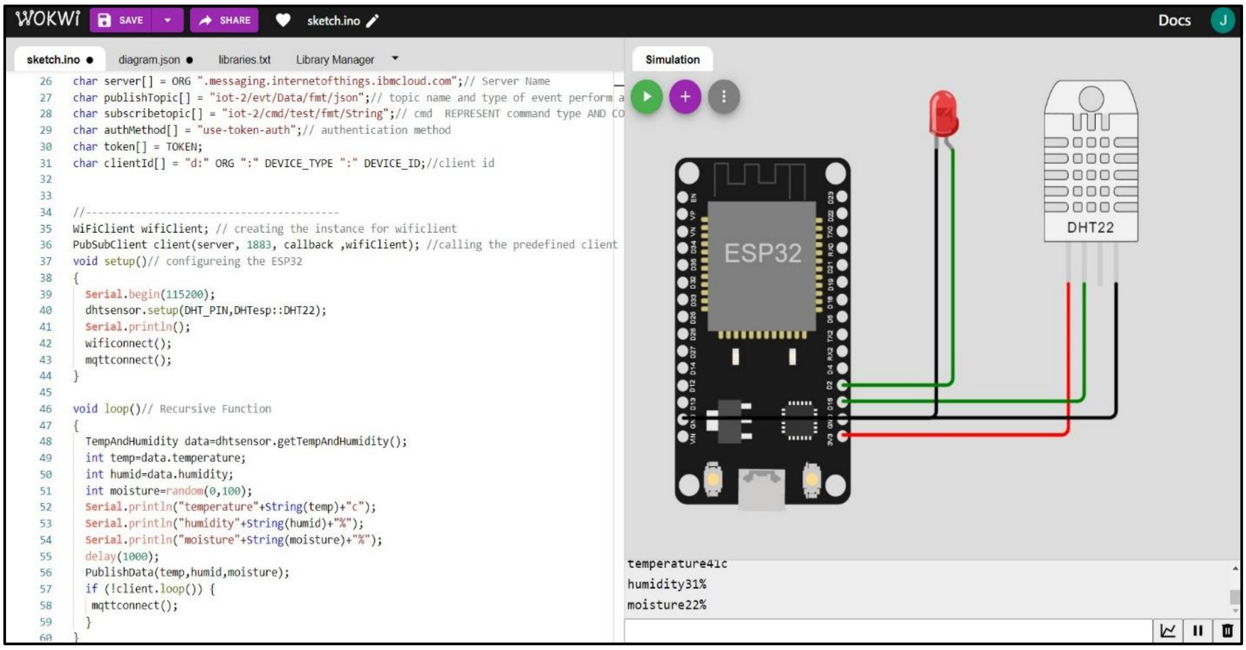
SOURCE CODE:

```
#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,portand wificredential
void setup()// configureing 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 in form 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="";


}
```

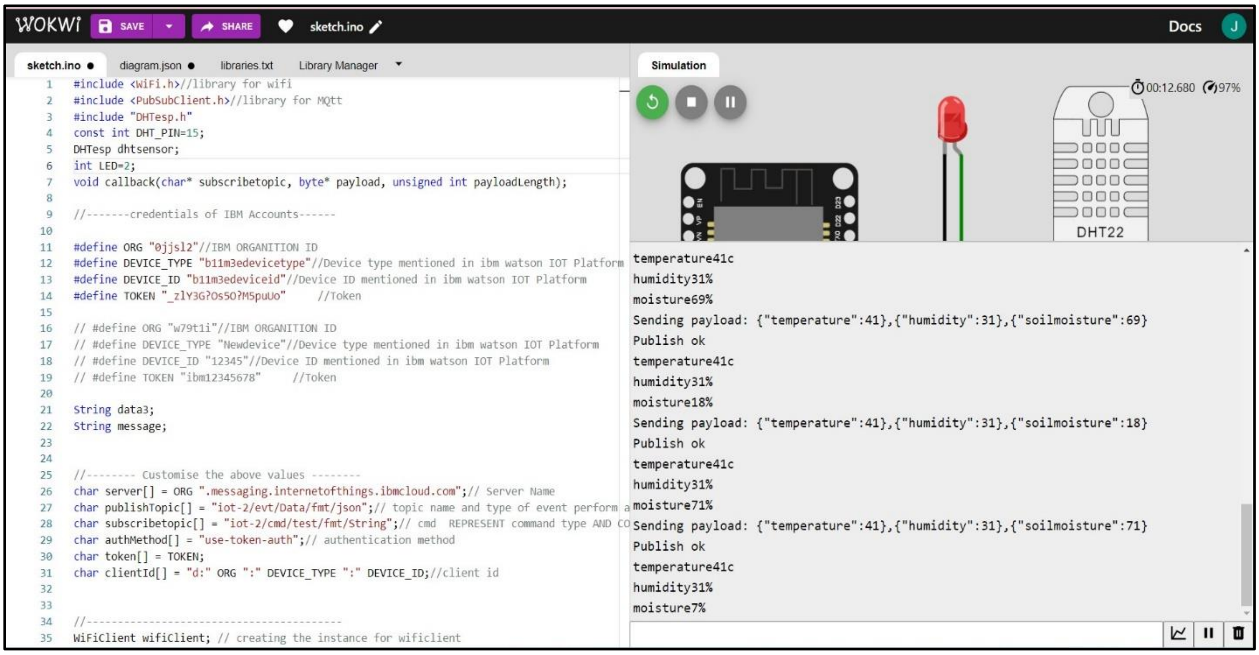
CIRCUIT DIAGRAM:



```
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform a
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND CO
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
void setup()// configureing 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();
  }
}
```

```
temperature41c
humidity31%
moisture22%
```

SENDING DATA TO IBM WATSON CLOUD:



```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHTesp.h"
const int DHT_PIN=15;
DHTesp dhtsensor;
int LED=2;
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

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

#define ORG "0jjsl2"//IBM ORGANITION ID
#define DEVICE_TYPE "b11m3edevicetype"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "b11m3edeviceid"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "_zlY3G?Os5O?M5puUo"     //Token

// #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 a
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND CO
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
```

```
temperature41c
humidity31%
moisture69%
Sending payload: {"temperature":41},{"humidity":31},{"soilmoisture":69}
Publish ok
temperature41c
humidity31%
moisture18%
Sending payload: {"temperature":41},{"humidity":31},{"soilmoisture":18}
Publish ok
temperature41c
humidity31%
moisture71%
Sending payload: {"temperature":41},{"humidity":31},{"soilmoisture":71}
Publish ok
temperature41c
humidity31%
moisture7%
```