

SPRINT 3

Date	12 November 2022
Team ID	PNT2022TMID12911
Project Name	Smart Farmer – IoT Enabled Farming Application
Maximum Marks	8 Marks

SENDING SENSOR DATA FROM WOKWI TO IBM WATSON IOT PLATFORM:

PROGRAM FOR SENDING TEMPERATURE AND HUMIDITY VALUES USING MQTT PROTOCOL:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"
#define DHTPIN 15
#define DHTTYPE DHT22
#define LED 2

DHT dht (DHTPIN, DHTTYPE);
void callback(char* subscribtopic, byte* payload, unsigned int payloadLength);
#define ORG "tu4jce"//IBM ORGANITION ID
#define DEVICE_TYPE "NodeMCU"//Device type
#define DEVICE_ID "12345"//Device ID
#define TOKEN "2W?*d5U83t+ICiNhyJ" //Token
String data3;
float h, t;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribtopic[] = "iot-2/cmd/command/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
//-----
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
void setup()
{
  Serial.begin(115200);
  dht.begin();
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()
```

```

{

    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();
    }
}

void PublishData(float temp, float humid) {
    mqttconnect();
    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");
    } 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()

```

```

{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

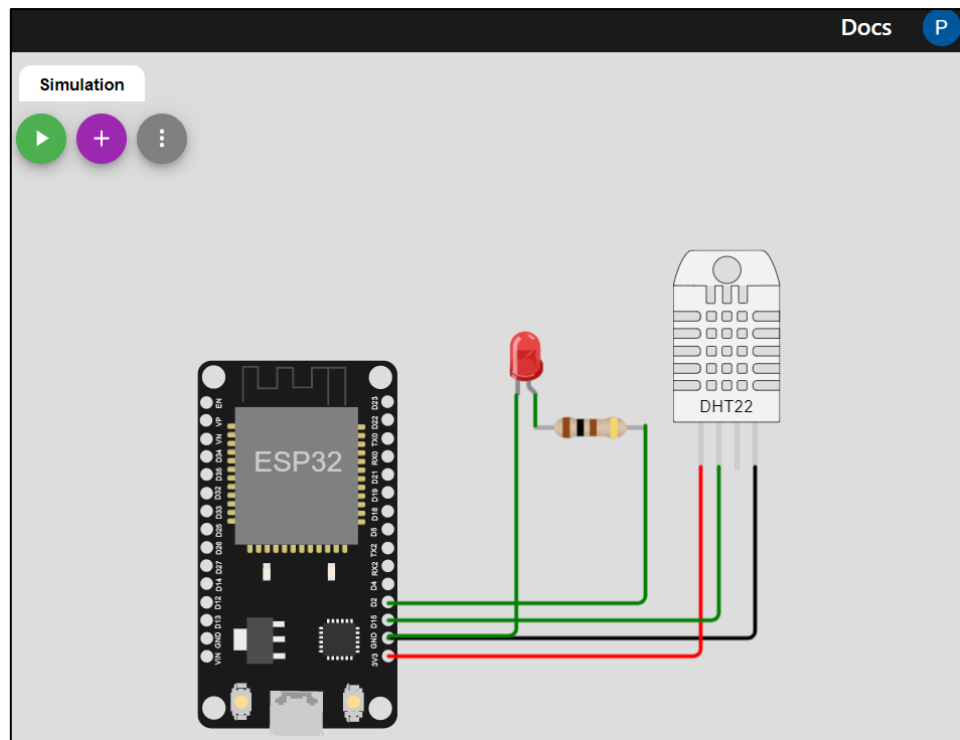
void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

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="";
}

```

Wokwi project link: <https://wokwi.com/projects/348379419871543890>

CIRCUIT:



WOKWI SERIAL MONITOR:

```
Connecting to ...  
WiFi connected  
IP address:  
10.10.0.2  
Reconnecting client to tu4jce.messaging.internetofthings.ibmcloud.com  
iot-2/cmd/command/fmt/String  
subscribe to cmd OK
```

Connecting to IBM Watson IoT platform

```
temp:24.00  
Humid:40.00  
Sending payload: {"temp":24.00,"Humid":40.00}  
Publish ok  
temp:24.00  
Humid:40.00  
Sending payload: {"temp":24.00,"Humid":40.00}  
Publish ok
```

Publishing temperature and humidity values to the IBM Watson IoT platform

IBM Watson IoT platform:

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main content area displays a table of devices. The first device, ID 12345, is highlighted. Below the table, a detailed view of the selected device is shown, including its identity, device information, recent events, state, and logs. The 'Device Information' tab is active, showing details such as Device ID, Device Type, Date Added, Added By, and Connection Status.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	NodeMCU	Device	Nov 15, 2022 11:11 AM	

Device Information

- Device ID: 12345
- Device Type: NodeMCU
- Date Added: Nov 15, 2022 11:11 AM
- Added By: pradeepsk17@gmail.com
- Connection Status: **Connected**
Connection Time: Nov 15, 2022 2:51 PM
Client Address: 185.178.200.130 Insecure

Connected Status in IBM Watson IoT platform

The screenshot shows the IBM Watson IoT Platform interface with the 'Recent Events' tab selected for device ID 12345. The tab displays a list of recent events, each containing an event type, value, format, and last received timestamp. The events are listed in a table format.

Event	Value	Format	Last Received
Data	{"temp":24,"Humid":40}	json	a few seconds ago
Data	{"temp":24,"Humid":40}	json	a few seconds ago
Data	{"temp":24,"Humid":40}	json	a few seconds ago
Data	{"temp":24,"Humid":40}	json	a few seconds ago
Data	{"temp":24,"Humid":40}	json	a few seconds ago

Recent Events in IBM Watson IoT platform

Once the sensor data like temperature and humidity gets updated in the IBM Watson IoT platform, those sensor data's will be available under recent events.