

Sprint-1

Date	02 November 2022
Team ID	PNT2022TMID01100
Project Name	Project – Signs with smart connectivity for better safety.

Simulation Creation:

The screenshot displays the Wokwi web-based simulation environment. On the left, the 'sketch.ino' file is open, showing C++ code for an ESP32 microcontroller. The code includes libraries for WiFi, MQTT, and DHT11, and configures the device to connect to an IBM Watson IoT Platform. On the right, a 'Simulation' window shows a virtual circuit board with an ESP32 module, a breadboard with resistors, and a DHT11 temperature and humidity sensor. The components are connected with colored wires. The top of the browser window shows the URL 'wokwi.com/projects/348127920425796179' and various browser tabs. The bottom of the screen shows a Windows taskbar with the date and time as 17-11-2022, 22:58.

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include "DHT.h" // Library for dht11
4 #define DHTPIN 15 // what pin we're connected to
5 #define DHTTYPE DHT22 // define type of sensor DHT 11
6 #define LED 2
7 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of dht conn
8
9 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "p9i6a2" //IBM ORGANITION ID
14 #define DEVICE_TYPE "Ibmcloud_1" //Device type mentioned in ibm watson IOT Platform
15 #define DEVICE_ID "Ibmcloud_id" //Device ID mentioned in ibm watson IOT Platform
16 #define TOKEN "!V1wn8F-Ms*" //Token
17 String data3;
18 float h, t;
19 int speed;
20
21
22 //----- Customise the above values -----
23 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
24 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform
25 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type A
26 char authMethod[] = "use-token-auth"; // authentication method
27 char token[] = TOKEN;
28 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
29
30 //-----
31
32 WiFiClient wifiClient; // creating the instance for wificlient
33 PubSubClient client(server, 1883, callback, wifiClient); //calling the predefi
34 void setup() // configuring the ESP32
35 {
```

Wokwi simulation link: <https://wokwi.com/projects/348127920425796179>

Code:

```
#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 typr of
dht connected

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

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

#define ORG "p9i6a2"//IBM ORGANITION ID
#define DEVICE_TYPE "Ibmcloud_1"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "Ibmcloud_1id"//Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "!V1wn8F-Ms*)in_EQF"    //Token
String data3;
float h, t;
int speed;

//----- 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/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 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);
    pinMode(13,INPUT);//Road1
```

```

    pinMode(12, INPUT); //Road2
    pinMode(14, INPUT); //Road3
    pinMode(27, INPUT); //Road4
    // pinMode(13, INPUT);
    // pinMode(13, INPUT);
    //pinMode(13, INPUT);
    //pinMode(13, INPUT);
    delay(10);
    Serial.println();
    wificonnect();
    mqttconnect();
}
int R1, R2, R3, R4;
void loop() // Recursive Function
{

    h = dht.readHumidity();
    t = dht.readTemperature();
    R1=digitalRead(13);
    R2=digitalRead(12);
    R3=digitalRead(14);
    R4=digitalRead(27);
    Serial.print("Temperature:");
    Serial.println(t);
    Serial.print("Humidity:");
    Serial.println(h);
    speed=round((h+t)/2);
    Serial.print("Speed:");
    Serial.println(speed);

    PublishData(t, h, speed, R1, R2 ,R3 ,R4);
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

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

void PublishData(float temp, float humid, int speed, int R1, int R2, int R3,
int R4) {
    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 += "," "\"Speed\":";
payload += speed;
payload += "," "\"Road1\":";
payload += R1;
payload += "," "\"Road2\":";
payload += R2;
payload += "," "\"Road3\":";
payload += R3;
payload += "," "\"Road4\":";
payload += R4;
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=="light on")
    {
        Serial.println(data3);
        digitalWrite(LED,HIGH);

    }

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

    }
    data3="";
}

```

}