

Sprint-1

Date	02 November 2022
Team ID	PNT2022TMID19258
ProjectName	Project–Signs with smartconnectivity for better road safety.

Simulation Creation:

The screenshot shows the Wokwi web-based simulation environment. On the left, the 'sketch.ino' file is open, displaying the following code:

```
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 ORGANIZATION ID
14 #define DEVICE_TYPE "Ibmcloud_1" //Device type mentioned in ibm watson IOT Platform
15 #define DEVICE_ID "Ibmcloud_id1" //Device ID mentioned in ibm watson IOT Platform
16 #define TOKEN "Iv1wn8F-Ms*)in_EQF" //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() // configureing the ESP32
35 {
```

On the right, the 'Simulation' pane shows a 3D model of the hardware. An ESP32 microcontroller is connected to a DHT22 sensor via a breadboard. The sensor is connected to the ESP32's pins: VCC to pin 15, GND to pin 14, and data to pin 1. The ESP32 is also connected to a USB cable. The bottom status bar shows the system clock as 22:58 on 17-11-2022.

Wokwisimulationlink: <https://wokwi.com/projects/348127920425796179>

Code:

```
#include<WiFi.h>//libraryforwifi
#include<PubSubClient.h>//library forMQtt
#include"DHT.h"// Libraryfor dht11
#defineDHTPIN 15    //what pinwe're connectedto
#defineDHTTYPE DHT22    //define typeofsensorDHT 11
#defineLED 2
DHTdht (DHTPIN,DHTTYPE);//creatingtheinstanceby passingpin andtypr
ofdhtconnected

voidcallback(char* subscribetopic,byte*payload,unsigned intpayloadLength);

//-----credentialsof IBM Accounts-----

#defineORG "p9i6a2"//IBMORGANITION ID
#defineDEVICE_TYPE "Ibmcloud_1"//Devicetypementionedin ibmwatson IOTPlatform
#defineDEVICE_ID "Ibmcloud_1id"//DeviceID mentionedin ibmwatson IOTPlatform
#defineTOKEN "!V1wn8F-Ms*)in_EQF"
                                //Token

Stringdata3;
float,
t;intspeed;

//-----Customisetheabove values -----
charserver[] =ORG ".messaging.internetofthings.ibmcloud.com";//Server
NamecharpublishTopic[] ="iot-2/evt/Data/fmt/json";// topicname
andtypeofeventperform andformat inwhich datatobesend
charsubscribetopic[] ="iot-2/cmd/command/fmt/String";// cmd
                                REPRESENT

commandtype ANDCOMMAND ITEST OFFORMATSTRING
charauthMethod[] ="use-token-auth";//
authenticationmethodchartoken[] =TOKEN;
charclientId[] ="d:" ORG:"DEVICE_TYPE":" DEVICE_ID;//clientid

//-----
WiFiClientwifiClient; //creating theinstance
forwificlientPubSubClientclient(server,
1883,callback,wifiClient);//callingthepredefinedclient idby
passingparameterlikeserver id,portandwificredential
voidsetup()// configureingthe ESP32
{
    Serial.begin(115200);dht.b
    egin();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();wific
onnect();mqttconnect();
}
intR1, R2,R3, R4;
voidloop();// RecursiveFunction
{

h=dht.readHumidity();
t=dht.readTemperature();R1=digitalRea
d(13);R2=digitalRead(12);R3=digitalRe
ad(14);R4=digitalRead(27);Serial.prin
t("Temperature:");Serial.println(t);S
erial.print("Humidity:");Serial.print
ln(h);speed=round((h+t)/2);Serial.pri
nt("Speed:");Serial.println(speed);

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

/*.....retrievingto
Cloud.....*/

voidPublishData(float temp,float humid,int speed,int R1,int R2,int R3,intR4)
{
mqttconnect();//functioncall forconnecting toibm
/*
creatingthe Stringin inform JSonto updatethe datato ibmcloud
*/
Stringpayload ="{\"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("Sendingpayload: ");
Serial.println(payload);

if(client.publish(publishTopic, (char*)payload.c_str())) {
    Serial.println("Publishok");// if itsucessfullyupload dataon thecloudthenit
willprint publishokinSerial monitoror elseit willprint publishfailed
}else {
    Serial.println("Publishfailed");
}
}

voidmqttconnect() {
    if(!client.connected())
    {Serial.print("Reconnectingclient
to");Serial.println(server);
    while(!!!client.connect(clientId, authMethod,token)) {
        Serial.print(".");dela
y(500);
    }

    initManagedDevice();
    Serial.println();
}
}

voidwificonnect() //functiondefinationforwificonnect
{
    Serial.println();Serial.print("Co
nnectingto");

```

```

    WiFi.begin("Wokwi-GUEST","",6);//passingthe wificredentials
toestablishtheconnection
    while(WiFi.status() !=WL_CONNECTED)
    {delay(500);
    Serial.print(".");
    }
    Serial.println("");Serial.println
("WiFiconnected");Serial.println(
"IPAddress:");Serial.println(WiFi
.localIP());
}

voidinitManagedDevice() {
    if(client.subscribe(subscribetopic))
    {Serial.println((subscribetopic));Serial.p
rintln("subscribeto cmdOK");
    }else {
    Serial.println("subscribeto cmdFAILED");
    }
}

voidcallback(char* subscribetopic,byte*payload,unsigned intpayloadLength)
{

    Serial.print("callbackinvoked fortopic: ");
    Serial.println(subscribetopic);
    for(int i=0;i<payloadLength; i++){
        //Serial.print((char)payload[i]);da
        ta3+= (char)payload[i];
    }

    Serial.println("data:"+
    data3);if(data3=="lighton")
    {
Serial.println(data3);digitalWrite(LED,
HIGH);

    }

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

    }
    data3="";
}

```

}