Project Development Phase
Sprint 3

Signs with smart connectivity for Better road safety

Team ID	PNT2022TMID17080
Project Name	Project - Signs with smart connectivity for Better road safety
Marks	8 Marks

WOWKI LINK : **https://wokwi.com/projects/348426120386839123**

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 5     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
//-------credentials of IBM Accounts------
#define ORG "b59mry"//IBM ORGANITION ID
#define DEVICE_TYPE "Event_1"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1111"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
float h, t;
```

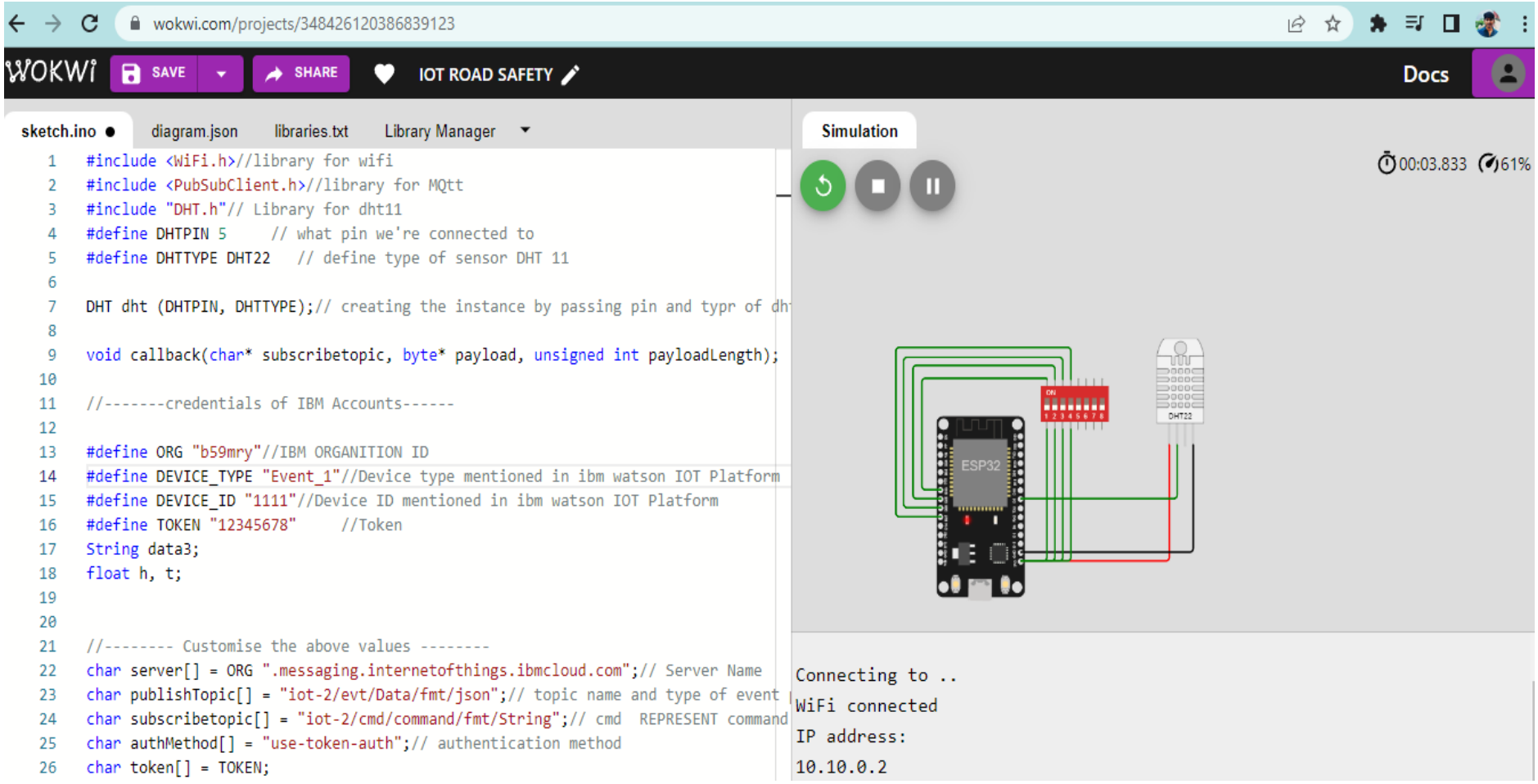
```
//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// 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(33, INPUT); //North
  pinMode(25, INPUT); // South
  pinMode(26, INPUT); // East
  pinMode(27, INPUT); // West
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}
int n, s, e, w;
void loop()// Recursive Function
{
  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.print("temp:");
  Serial.println(t);
```

```
  Serial.print("humidity:");
  Serial.println(h);

  n = digitalRead(33);
  s = digitalRead(25);
  e = digitalRead(26);
  w = digitalRead(27);

  PublishData(t, h, n, s, e, w);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}
/*.....................................retrieving to Cloud...............................*/
void PublishData(float temp, float humid, int n, int s, int e, int w) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"temp\":";
  payload += temp;
  payload += "," "\"humidity\":";
  payload += humid;
  payload += "," "\"North\":";
  payload += n;
  payload += "," "\"South\":";
  payload += s;
  payload += "," "\"East\":";
  payload += e;
  payload += "," "\"West\":";
  payload += w;
```

```
  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);
DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht connected

}
```

WOWKI :




```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 5     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht

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

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

#define ORG "b59mry"//IBM ORGANITION ID
#define DEVICE_TYPE "Event_1"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1111"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
float h, t;


//-------- 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
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
```

wokwi.com/projects/348426120386839123

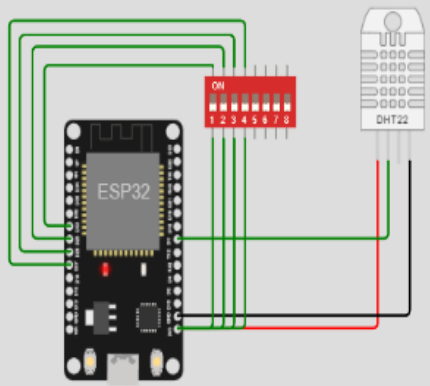
WOKWI SAVE SHARE IOT ROAD SAFETY Docs

sketch.ino diagram.json libraries.txt Library Manager

```
void setup()// configureing the ESP32
{
  Serial.begin(115200);
  dht.begin();
  pinMode(33, INPUT); //North
  pinMode(25, INPUT); // South
  pinMode(26, INPUT); // East
  pinMode(27, INPUT); // West
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

int n, s, e, w;

void loop()// Recursive Function
{

  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.print("temp:");
  Serial.println(t);
  Serial.print("humidity:");
  Serial.println(h);
```

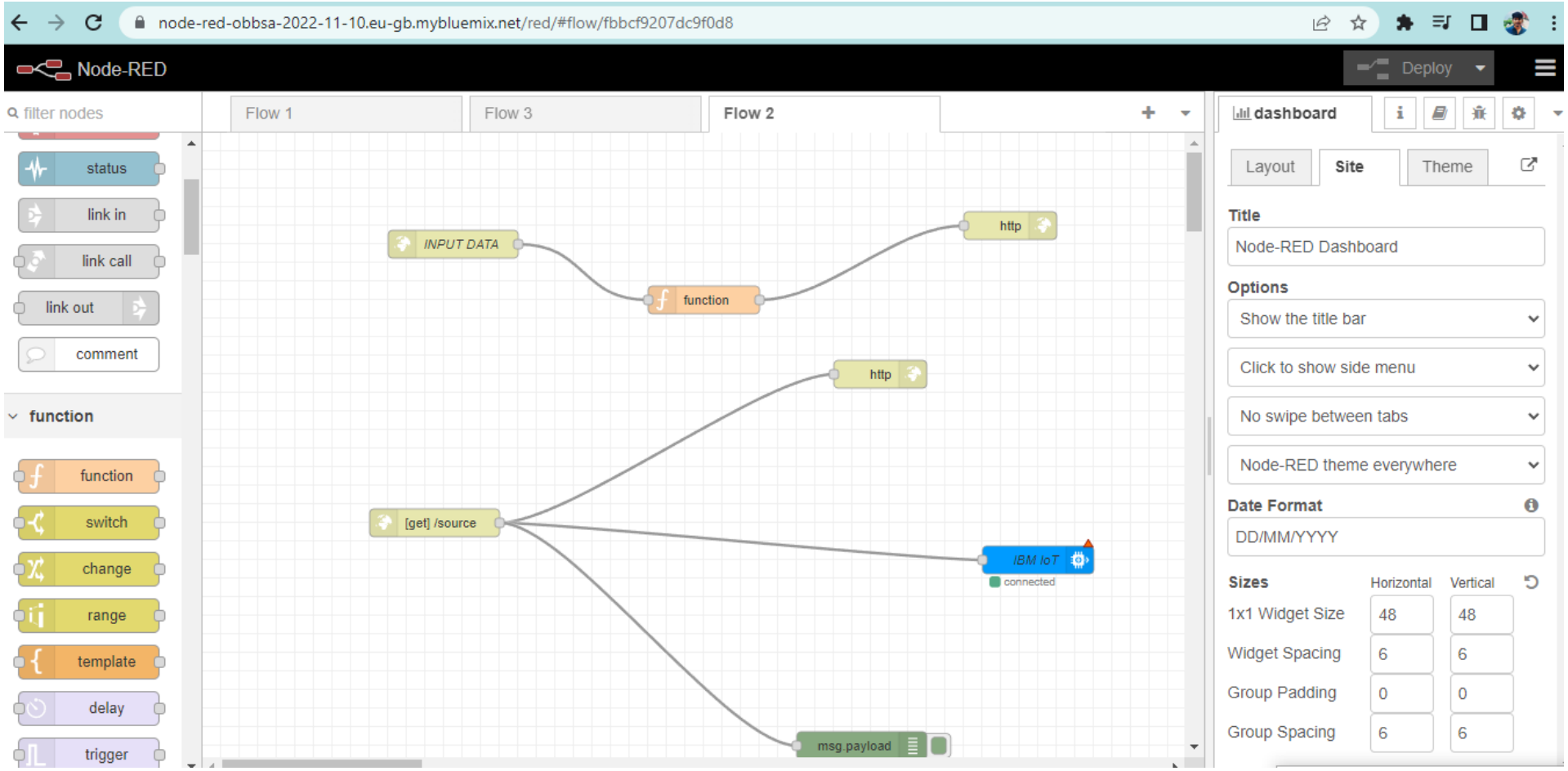
Simulation

00:17.099 65%




```
{"temp":3.00,"humidity":86.00,"North":0,"South":0,"East":0,"West":0}
Publish ok
temp:3.00
humidity:86.00
Sending payload:
{"temp":3.00,"humidity":86.00,"North":0,"South":0,"East":0,"West":0}
Publish ok
```

NODE RED:


node-red-obbsa-2022-11-10.eu-gb.mybluemix.net/red/#flow/fbbcf9207dc9f0d8
Node-RED
Deploy
filter nodes
Flow 1
Flow 3
Flow 2
status
link in
link call
link out
comment
function
function
switch
change
range
template
delay
trigger
INPUT DATA
function
http
http
[get] /source
IBM IoT
connected
msg.payload
dashboard
Layout
Site
Theme
Title
Node-RED Dashboard
Options
Show the title bar
Click to show side menu
No swipe between tabs
Node-RED theme everywhere
Date Format
DD/MM/YYYY
Sizes
Horizontal
Vertical
1x1 Widget Size
48
48
Widget Spacing
6
6
Group Padding
0
0
Group Spacing
6
6

OUTPUT:

```
temp:37.40
humidity:86.00
Sending payload:
{"temp":37.40,"humidity":86.00,"North":0,"South":0,"East":0,"West":0}
Publish ok
```