

Project Development Phase Sprint -1

Team ID	PNT2022TMID48891
Project Name	Signs with Smart Connectivity For Better Road Safety

Wowki Simulation:

The screenshot displays the WOKWI simulation environment. On the left, the 'sketch.ino' file is open, showing 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 5 // what pin we're connected to
5 #define DHTTYPE DHT22 // define type of sensor DHT 11
6
7 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of
8
9 void callback(char* topic, byte* payload, unsigned int payloadLength)
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "7yf9hac" //IBM ORGANIZATION ID
14 #define DEVICE_TYPE "umamaheswari" //Device type mentioned in ibm watson IOT Platform
15 #define DEVICE_ID "uma27" //Device ID mentioned in ibm watson IOT Platform
16 #define TOKEN "12345678" //Token
17 String data3;
18 float h, t;
19
20
21 //----- Customise the above values -----
22 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
23 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
24 char subscribTopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command
25 char authMethod[] = "use-token-auth"; // authentication method
26 char token[] = TOKEN;
27 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
28
29
```

On the right, the 'Simulation' window shows a visual representation of the hardware. An ESP32 microcontroller is connected to a DHT22 sensor via a breadboard. The console output on the right shows the following sequence of events:

```
{ "temp":24.00,"humidity":40.00,"North":0,"South":0,"East":0,"West":0}
Publish ok
temp:24.00
humidity:40.00
Sending payload:
{"temp":24.00,"humidity":40.00,"North":0,"South":0,"East":0,"West":0}
Publish ok
```

Identity Device Information **Recent Events** State Logs
✕

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"Temperature":65,"Humidity":97,"North":16,"So...	json	a few seconds ago
event_1	{"Temperature":53,"Humidity":95,"North":67,"So...	json	a few seconds ago
event_1	{"Temperature":22,"Humidity":42,"North":83,"So...	json	a few seconds ago
event_1	{"Temperature":5,"Humidity":13,"North":22,"Sou...	json	a few seconds ago
event_1	{"Temperature":85,"Humidity":78,"North":...		

1 Simulation running

The screenshot displays the Node-RED web interface. On the left, the 'network' palette lists various input and output nodes. The main workspace contains two flows. Flow 1 consists of a single 'msg payload' node. Flow 2 starts with an 'IBM IoT' node, which branches into two parallel paths: one through a 'temperature Node' to a 'Temperature' output node, and another through a 'humidity' node to a 'Humidity' output node. Below these, there is a sequence of nodes: '[get]/sensor' connects to 'httpfunctionnode', which connects to 'http'. At the bottom, 'Light on' and 'Light off' nodes connect to a 'command function node', which then connects to an 'http' node. The right sidebar shows a debug console with logs for the 'msg payload' node, displaying the following data:

```

11/17/2022, 9:38:52 PM: node:
9c3371245778150e
id:
2/type/HAYAN24Id/HAYAN24Id/event_1/first/json
msg payload: Object
> { Temperature: 7, Humidity: 13 }

11/17/2022, 9:38:54 PM: node:
9c3371245778150e
id:
2/type/HAYAN24Id/HAYAN24Id/event_1/first/json
msg payload: Object
> { Temperature: 58, Humidity: 22 }

11/17/2022, 9:38:57 PM: node:
9c3371245778150e
id:
2/type/HAYAN24Id/HAYAN24Id/event_1/first/json
msg payload: Object
> { Temperature: 51, Humidity: 91 }

11/17/2022, 9:37:01 PM: node:
9c3371245778150e
id:
2/type/HAYAN24Id/HAYAN24Id/event_1/first/json
msg payload: Object

```

Node Red Web UI:

