

Project Development Phase Sprint -2

Team ID	PNT2022TMID33212
Project Name	Signs with Smart Connectivity For Better Road Safety

Wowki Simulation:

<https://wokwi.com/projects/348592973152256594>

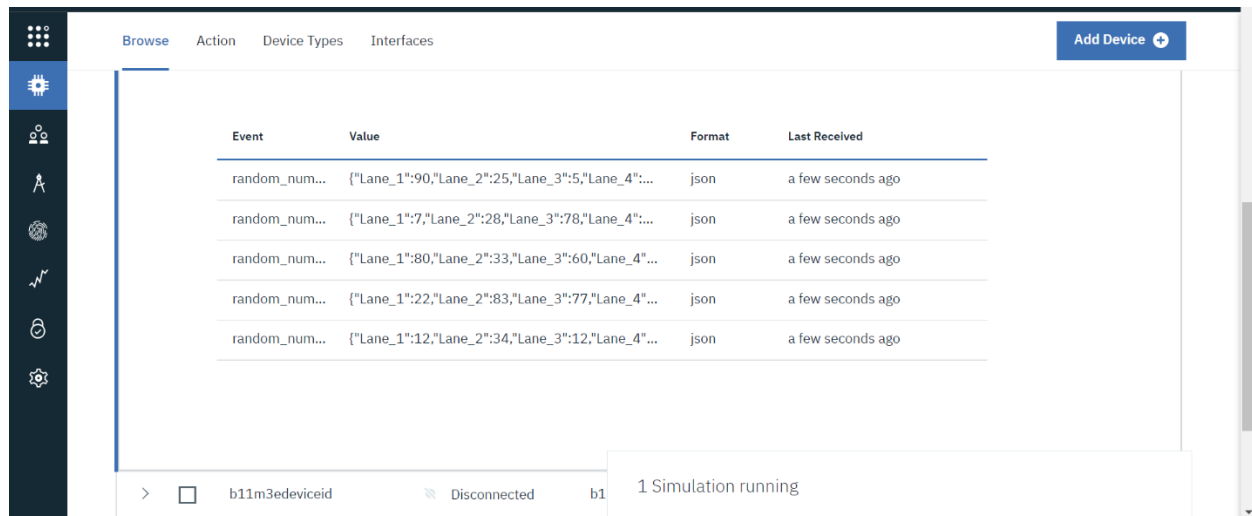
The screenshot displays the Wokwi simulation environment. On the left, the 'sketch.ino' file is open, showing the following code:

```
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
8
9 void callback(char* topic, byte* payload, unsigned int payloadLength) {
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "3n45bc" //IBM ORGANIZATION ID
14 #define DEVICE_TYPE "98nivetha" //Device type mentioned in ibm watson IOT
15 #define DEVICE_ID "98nivethaid" //Device ID mentioned in ibm watson IOT
16 #define TOKEN "gTlxIHv105MH)D?IfX" //Token
17 String data3;
18 float h, t;
19
20 //----- Customise the above values -----
21
22 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server
23 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of
24 char subscribTopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT
25 char authMethod[] = "use-token-auth"; // authentication method
26 char token[] = TOKEN;
27 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
```

On the right, the 'Simulation' pane shows a visual representation of the ESP32 microcontroller and the DHT22 sensor connected by wires. Below the simulation, the output console displays the following messages:


```
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
```

IoT Device- IoT Plat form

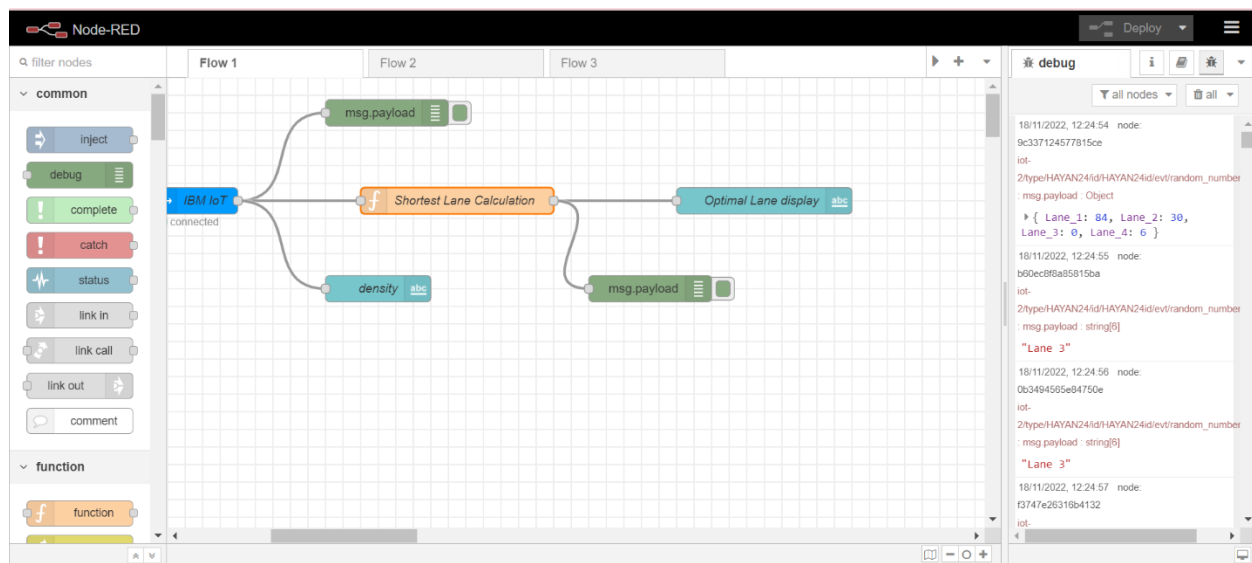


The screenshot shows an IoT platform interface with a sidebar on the left containing icons for various functions. The main area displays a table of events. Below the table, there is a status bar showing a device ID, its connection status, and the number of simulations running.

Event	Value	Format	Last Received
random_num...	{"Lane_1":90,"Lane_2":25,"Lane_3":5,"Lane_4":...	json	a few seconds ago
random_num...	{"Lane_1":7,"Lane_2":28,"Lane_3":78,"Lane_4":...	json	a few seconds ago
random_num...	{"Lane_1":80,"Lane_2":33,"Lane_3":60,"Lane_4":...	json	a few seconds ago
random_num...	{"Lane_1":22,"Lane_2":83,"Lane_3":77,"Lane_4":...	json	a few seconds ago
random_num...	{"Lane_1":12,"Lane_2":34,"Lane_3":12,"Lane_4":...	json	a few seconds ago

> ☐ b11m3deviceid  Disconnected b1 1 Simulation running

Node Red



Edit function node

Delete

Cancel

Done

Properties

Name

Shortest Lane Calculation

Setup

On Start

On Message

On Stop

```
1 var l1 = msg.payload.Lane_1;
2 var l2 = msg.payload.Lane_2;
3 var l3 = msg.payload.Lane_3;
4 var l4 = msg.payload.Lane_4;
5
6 mini = Math.min(l1,l2,l3,l4);
7
8 res = ""
9 switch(mini) {
10     case l1: res = "Lane 1"; break;
11     case l2: res = "Lane 2"; break;
12     case l3: res = "Lane 3"; break;
13     case l4: res = "Lane 4"; break;
14 }
15
16 msg.payload = res;
17
18 return msg;
19
```

Node Red Web UI

