

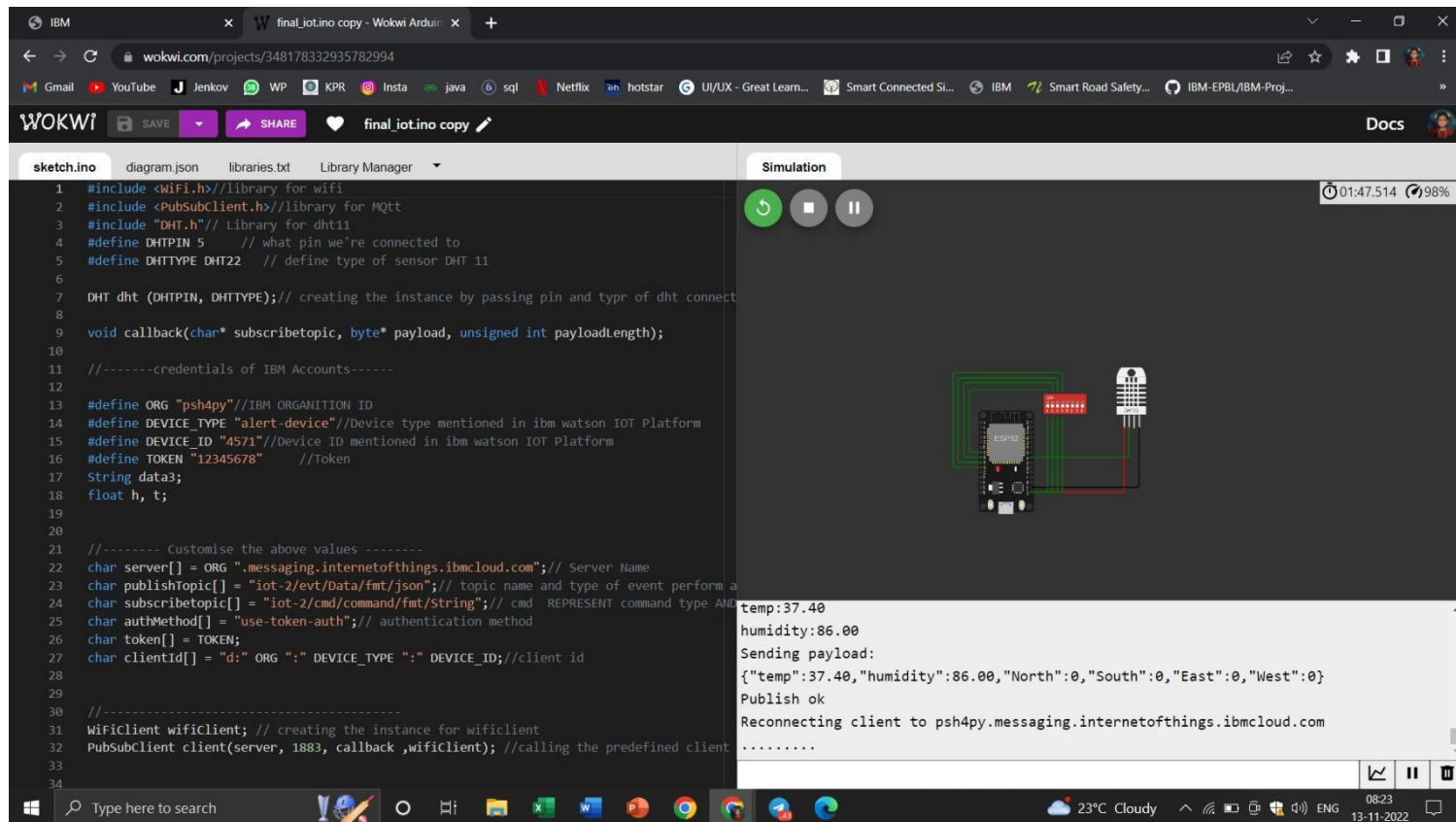
Project Development Phase Sprint II

Date	13 November 2022
Team ID	PNT2022TMID07843
Project Name	Signs with Smart Connectivity for better road safety

SPRINT TARGETS:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Safe Ride	USN-4	As a traveler , I should have a hustie free journey	20	Medium	Sneha Nivetha Jeyashri Pawana Prudhvi Naveen

Wokwi Simulation: <https://wokwi.com/projects/348178332935782994>



The screenshot displays the Wokwi web-based IDE interface. The left pane shows the 'sketch.ino' file with 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 dht connect
8
9 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "psh4py" // IBM ORGANIZATION ID
14 #define DEVICE_TYPE "alert-device" // Device type mentioned in IBM Watson IOT Platform
15 #define DEVICE_ID "4571" // 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 perform a
24 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND
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
30 //-----
31 WiFiClient wifiClient; // creating the instance for wifi client
32 PubSubClient client(server, 1883, callback, wifiClient); // calling the predefined client
33
34
```

The right pane shows the 'Simulation' window with a visual representation of the hardware (ESP32, DHT22 sensor, and breadboard) and a terminal 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
Reconnecting client to psh4py.messaging.internetofthings.ibmcloud.com
.....
```

The bottom status bar indicates the system time as 08:23 on 13-11-2022, with weather information showing 23°C Cloudy.

IoT Device – IoT Platform

Device Details

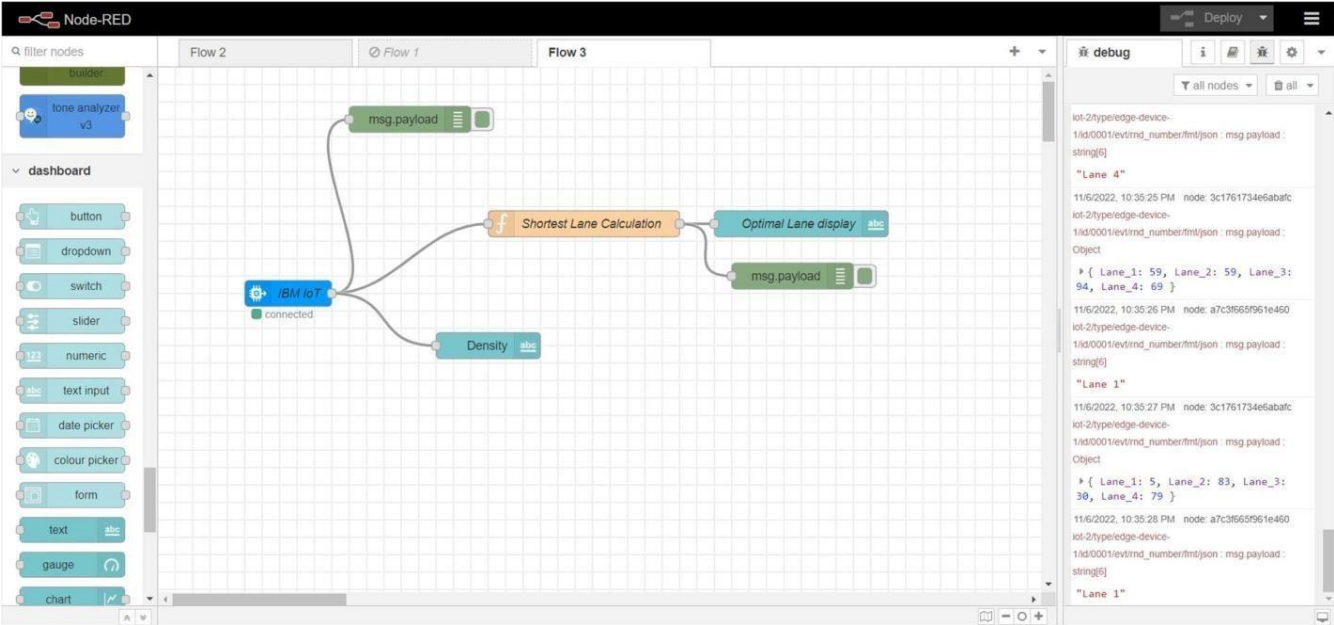
Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
0001	Disconnected	edge-device-1	Device	Nov 5, 2022 8:56 PM	

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

Event	Value	Format	Last Received
rnd_number	{"Lane_1":5,"Lane_2":83,"Lane_3":30,"Lane_4"...}	json	a few seconds ago
rnd_number	{"Lane_1":59,"Lane_2":59,"Lane_3":94,"Lane_4"...}	json	a few seconds ago
rnd_number	{"Lane_1":93,"Lane_2":88,"Lane_3":49,"Lane_4"...}	json	a few seconds ago
rnd_number	{"Lane_1":2,"Lane_2":61,"Lane_3":21,"Lane_4"...}	json	a few seconds ago
rnd_number	{"Lane_1":70,"Lane_2":11,"Lane_3":69,"Lane_4"...}	json	a few seconds ago

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
10 switch(mini) {
11     case l1: res = "Lane 1"; break;
12     case l2: res = "Lane 2"; break;
13     case l3: res = "Lane 3"; break;
14     case l4: res = "Lane 4"; break;
15 }
16
17 msg.payload = res;
18
19 return msg;
```

Node Red Web UI

