

SPRINT- 2

Date	11 November 2022
Team ID	PNT2022TMID02664
Project Name	Smart Waste Management System for Metropolitan Cities
Points	20

Created an IOT device to sense the level of bins and do code for the device using the Wokwi platform.

CODE:

Run the code here: <https://wokwi.com/projects/348577316647993940>

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);

#define ORG "edsau1"
#define DEVICE_TYPE "ESP"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
String data3;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);

const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
float level;

void setup() {
  Serial.begin(115200);
```

```

pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}

void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * SOUND_SPEED/2;
level = 400 - distance;
Serial.print("Distance (cm): ");
Serial.println(level);
if(level>300)
{
Serial.println("ALERT!!");
delay(1000);
PublishData(level);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
else
{
Publishdata2(level);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
delay(1000);
}

void PublishData(float dist) {
mqttconnect();
String payload = "{\"Level\":\"";
payload += dist;
payload += "\", \"ALERT!!\":\"\" \"Bin level less than 100 Kgs \";
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);
}

```

```

if (client.publish(publishTopic, (char*) payload.c_str())) {
  Serial.println("All good");
} else {
  Serial.println("Failed");
}
}

void Publishdata2(float dist) {
  mqttconnect();
  String payload = "{\"Level\":";
  payload += dist;
  payload += "}";
  Serial.print("Sending payload: ");
  Serial.println(payload);

  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");
  } 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()
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "", 6);
  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);
data3="";
}

```

Sensor circuit:

The screenshot displays the Wokwi IoT simulator interface. On the left, the code editor shows the following code:

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribetopic, byte* payload, unsigned int
4 payloadLength);
5
6
7 #define ORG "edsaul"
8 #define DEVICE_TYPE "Esp"
9 #define DEVICE_ID "12345"
10 #define TOKEN "12345678"
11 String data3;
12
13
14 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
15 char publishTopic[] = "iot-2/evt/data/fmt/json";
16 char subscribetopic[] = "iot-2/cmd/test/fmt/String";
17 char authMethod[] = "use-token-auth";
18 char token[] = TOKEN;
19 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
20
21 WiFiClient wifiClient;
22 PubSubClient client(server, 1883, callback, wifiClient);
23
24 const int trigPin = 5;
25 const int echoPin = 18;
26 #define SOUND_SPEED 0.034
27 long duration;
28 float distance;
29 float level;
30
31 void setup() {
32   Serial.begin(115200);
33   pinMode(trigPin, OUTPUT);

```

On the right, the simulation window shows a visual representation of the ESP32 microcontroller connected to a distance sensor module. Below the simulation, the console output displays the following messages:

```

All good
Distance (cm): 338.04
ALERT!!
Sending payload: {"Level":338.04,"ALERT!!":"Bin level less than 100 Kgs "}
All good
Distance (cm): 338.04
ALERT!!

```

Watson IoT Platform:

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar labeled 'Search by Device ID' is present. Below the search bar is a table listing devices with columns: Device ID, Status, Device Type, Class ID, Date Added, and Descriptive Location. The table contains 6 items. A 'Device Simulator' toggle is set to 'On'. At the bottom, it says '0 Simulations running'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
123	Disconnected	Bin1	Device	17 Nov 2022 2:59 PM	
1234	Disconnected	Bin2	Device	17 Nov 2022 3:00 PM	
12345	Connected	ESP	Device	17 Nov 2022 6:01 PM	
1312	Disconnected	RSP	Device	15 Nov 2022 7:10 PM	
Bin1_1	Disconnected	Bin1	Device	17 Nov 2022 3:02 PM	
RSP_1	Disconnected	RSP	Device	17 Nov 2022 2:58 PM	

The screenshot shows the 'Recent Events' view for device 12345. The top navigation bar is the same as the previous screenshot. The 'Recent Events' tab is selected, showing a table of events. The table has columns: Event, Value, Format, and Last Received. There are 3 events listed. Below the table, it says '0 Simulations running'.

Event	Value	Format	Last Received
Data	{"Level":338.04,"ALERT!":"Bin level less than 10..."}	json	a few seconds ago
Data	{"Level":338.04,"ALERT!":"Bin level less than 10..."}	json	a few seconds ago
Data	{"Level":338.04,"ALERT!":"Bin level less than 10..."}	json	a few seconds ago