

## DELIVERY OF SPRINT 2

Date	14 November 2022
Team ID	PNT2022TMID04392
Project Name	Smart waste management system for metropolitan cities

Interfacing Load Sensor HX711 with  
ESP32

WOKWI Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "4yi0vc"
#define DEVICE_TYPE "nodeMcu"
#define DEVICE_ID "Assignment4"
#define TOKEN "123456789"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Gayathri/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();

const int trigpin=5;
const int echopin=18;
String command;
String data="";

long duration;
float dist;

void setup()
{
  Serial.begin(115200);
  pinMode(led, OUTPUT);
```

```

    pinMode(trigpin, OUTPUT);
    pinMode(echopin, INPUT);
    wifiConnect();
    mqttConnect();
}

void loop() {
    bool isNearby = dist < 100;
    digitalWrite(led, isNearby);

    publishData();
    delay(500);

    if (!client.loop()) {
        mqttConnect();
    }
}

void wifiConnect() {
    Serial.print("Connecting to "); Serial.print("Wifi");
    WiFi.begin("Wokwi-GUEST", "", 6);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.print("WiFi connected, IP address: ");
    Serial.println(WiFi.localIP());
}

void mqttConnect() {
    if (!client.connected()) {
        Serial.print("Reconnecting MQTT client to "); Serial.println(server);
        while (!client.connect(clientId, authMethod, token)) {
            Serial.print(".");
            delay(500);
        }
        initManagedDevice();
        Serial.println();
    }
}

void initManagedDevice() {
    if (client.subscribe(topic)) {
        // Serial.println(client.subscribe(topic));
        Serial.println("IBM subscribe to cmd OK");
    } else {
        Serial.println("subscribe to cmd FAILED");
    }
}

```

```

void publishData()
{
    digitalWrite(trigpin,LOW);
    digitalWrite(trigpin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin,LOW);
    duration=pulseIn(echopin,HIGH);
    dist=duration*speed/2;
    if(dist<100){
        String payload = "{\"Alert Distance\":\"";
        payload += dist;
        payload += "\"}";

        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);
        if(client.publish(publishTopic, (char*) payload.c_str())) {
            Serial.println("Warning crosses 110cm -- it automaticaly of the
loop");
            digitalWrite(led,HIGH);
        }

    }

    if(dist>101 && dist<111){
        String payload = "{\"Normal Distance\":\"";
        payload += dist;
        payload += "\"}";

        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);

    }

}

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++){
        dist += (char)payload[i];
    }
    Serial.println("data:" + data3);
    if(data3=="lighton"){
        Serial.println(data3);
        digitalWrite(led,HIGH);
    }
}

```

```
}  
data3="";  
}
```

## OUTPUT:

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

```
1 #include <WiFi.h>  
2 #include <PubSubClient.h>  
3 WiFiClient wifiClient;  
4 String data3;  
5 #define ORG "dyi0vc"  
6 #define DEVICE_TYPE "nodeMcu"  
7 #define DEVICE_ID "Assignment4"  
8 #define TOKEN "123456789"  
9 #define speed 0.034  
10 #define led 14  
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";  
12 char publishTopic[] = "iot-2/evt/Gayathri";  
13 char topic[] = "iot-2/cmd/home/fat/String";  
14 char authMethod[] = "use-token-auth";  
15 char token[] = TOKEN;  
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE;  
17 PubSubClient client(server, 1883, wifiClient);  
18 void publishData();  
19  
20  
21 const int trigpin=5;  
22 const int echopin=18;  
23 String command;  
24 String data="";  
25  
26 long duration;
```

On the right, the simulation window shows a visual representation of the ESP32 and the Ultrasonic Distance Sensor. Below the visual, the console output displays the following messages:

```
Warning crosses 110cm -- it automatically of the loop  
Sending payload: {"Alert Distance":90.97}  
Warning crosses 110cm -- it automatically of the loop  
Sending payload: {"Alert Distance":90.97}  
Warning crosses 110cm -- it automatically of the loop
```