

SPRINT -3

DATE	16November 2022
TEAM ID	PNT2022TMID08717
PROJECT NAME	SMART WASTE MANAGEMENT FOR METROPOLITAN CITIES

```
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <WiFiClient.h>
LiquidCrystal_I2C lcd(0x27, 20, 4); // I2C address 0x3F, 16 column and 2 rows

int trigPin = 2;    // TRIG pin
int echoPin = 15;   // ECHO pin

#define ORG "qippa4"
#define DEVICE_TYPE "Esp32"
#define DEVICE_ID "Waste"
#define TOKEN "C72(GeQy)UPSVtHdUw"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name
char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command
is test format of strings
char authMethod[] = "use-token-auth"; // authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id

WiFiClient wifiClient; // creating instance for wifi
PubSubClient client(server, 1883, wifiClient);

void setup() {
  lcd.init();           // initialize the lcd
  lcd.backlight();
  pinMode(5,OUTPUT);
  pinMode(18,OUTPUT);
  pinMode(19,OUTPUT);
  pinMode(23,OUTPUT);
  pinMode(34,INPUT);
  pinMode(14,OUTPUT);
  // open the backlight
  pinMode(trigPin, OUTPUT); // config trigger pin to output mode
  pinMode(echoPin, INPUT);
  Serial.begin(115200);
  wifiConnect();
}
```

```

    mqttConnect();
    // config echo pin to input mode
}

float readcmCM()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    int duration = pulseIn(echoPin, HIGH);
    return duration * 0.034 / 2;
}

void loop()
{
    lcd.clear();

    publishData();
    delay(500);
    if (!client.loop())
    {
        mqttConnect(); // function call to connect to IBM
    }
}

/* -retrieving to cloud */

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("IBM subscribe to cmd OK");
    }
    else
    {
        Serial.println("subscribe to cmd FAILED");
    }
}
void publishData()
{
    float cm = readcmCM();

    if(digitalRead(34))
    {
        Serial.println("Motion Detected"); Serial.println("Lid Opened");
        digitalWrite(14, HIGH);
    }
    else
    {
        digitalWrite(14, LOW);
    } //PIR motion detection

    if(digitalRead(34))
    {
        if(cm <= 100)
        //Bin level detection
        {
            digitalWrite(23, HIGH);
            Serial.println("High Alert!!!,Trash bin is about to be full");
            Serial.println("Lid Closed");
            lcd.print("Full! Don't use");
            delay(2000);
            lcd.clear();
        }
    }
}

```

```

digitalWrite(18,LOW);
digitalWrite(19,LOW);
digitalWrite(5,LOW);
}
else if(cm > 100 && cm < 200)
{
digitalWrite(5, HIGH);
Serial.println("Warning!!,Trash is about to cross 75% of bin level");
digitalWrite(18,LOW);
digitalWrite(19,LOW);
digitalWrite(23,LOW);
}
else if(cm > 200 && cm < 300)
{
digitalWrite(18, HIGH);
Serial.println("Warning!!,Trash is about to cross 50% of bin level");
digitalWrite(5,LOW);
digitalWrite(19,LOW);
digitalWrite(23,LOW);
}
else if(cm > 300 && cm <=400)
{
digitalWrite(19, HIGH);
Serial.println("Bin is available");
digitalWrite(5,LOW);
digitalWrite(18,LOW);
digitalWrite(23,LOW);
}
delay(10000);
Serial.println("Lid Closed");
}
else
{
Serial.println("No motion detected");
}

if(cm <= 100)
{

digitalWrite(21,HIGH);
String payload = "{\"High Alert!!\":\":";payload += cm; payload += "left\" }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str()))
// if data is uploaded to cloud successfully,prints publish ok or prints
publish failed
{

```

```
Serial.println("Publish OK");
}
}
if(cm <= 250)
{
digitalWrite(22,HIGH);
String payload = "{\"Warning!!\":\":";payload+= cm; payload += "left\" }";
Serial.print("\n");
Serial.print("Sending distance: ");
Serial.println(cm);
if(client.publish(publishTopic, (char*) payload.c_str()))
{
Serial.println("Publish OK");
}
else
{
Serial.println("Publish FAILED");
}
}

float inches = (cm / 2.54); //print on LCD
lcd.setCursor(0,0);
lcd.print("Inches");
lcd.setCursor(4,0);
lcd.setCursor(12,0);
lcd.print("cm"); lcd.setCursor(1,1);
lcd.print(inches, 1);
lcd.setCursor(11,1);
lcd.print(cm,1);
lcd.setCursor(14,1);
delay(1000);
lcd.clear();
}
```

