

SOURCE CODE

TEAM ID	PNT2022TMID27367
TOPIC	Smart Waste Management for metropolitan cities

CODE

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <LiquidCrystal_I2C.h>
#include <ESP32Servo.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

#define ORG "i5w9m5"
#define DEVICE_TYPE "smart"
#define DEVICE_ID "smartID"
#define TOKEN "123456789"
char server[] = ORG
".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
//
WiFiClient wifiClient; // creating instance for wificlient
PubSubClient client(server, 1883, wifiClient);
#define echoPin 14
#define trigPin 12
#define led 2
Servo myservo;
long duration;
float distance;
void setup()
{
```

```

    pinMode(12, OUTPUT);
    pinMode(14, INPUT);
    pinMode(2, OUTPUT);
    myservo.attach(13);
    Serial.begin(115200);
    lcd.init();
    lcd.backlight();
    lcd.setCursor(1, 0);
    lcd.print("");
    wifiConnect();
    mqttConnect();
}
float readcmCM()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    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(distance>=100)

```

```

{
    Serial.println("distance:"+String(distance));
    digitalWrite(2, LOW);
    myservo.write(0);

}
else
{
    Serial.println("ALERT!! The garbage has reached the
maximum capacity value");
    digitalWrite(2, HIGH);
    myservo.write(180);

}

String payload = "{\\\"Warning!!\\\":\\\"";
payload += distance; 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(); }

```

