# ASSIGNMENT -4

**Smart Waste Management System For Metropolitan Cities**

# TEAM ID-PNT2022TMID05191

**Name- Gowtham k**

**Register number- 921319106064**

# 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/Data/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 = “{\”Normal Distance\”:”;

Payload += dist;

Payload += “}”;

Serial.print(“\n”);

Serial.print(“Sending payload: “);

Serial.println(payload);

If (client.publish(publishTopic, (char\*) payload.c\_str())) {

Serial.println(“Publish OK”);

}

}

If(dist>101 && dist<111){

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);

}else {

Serial.println(“Publish 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++){ Dist += (char)payload[i];

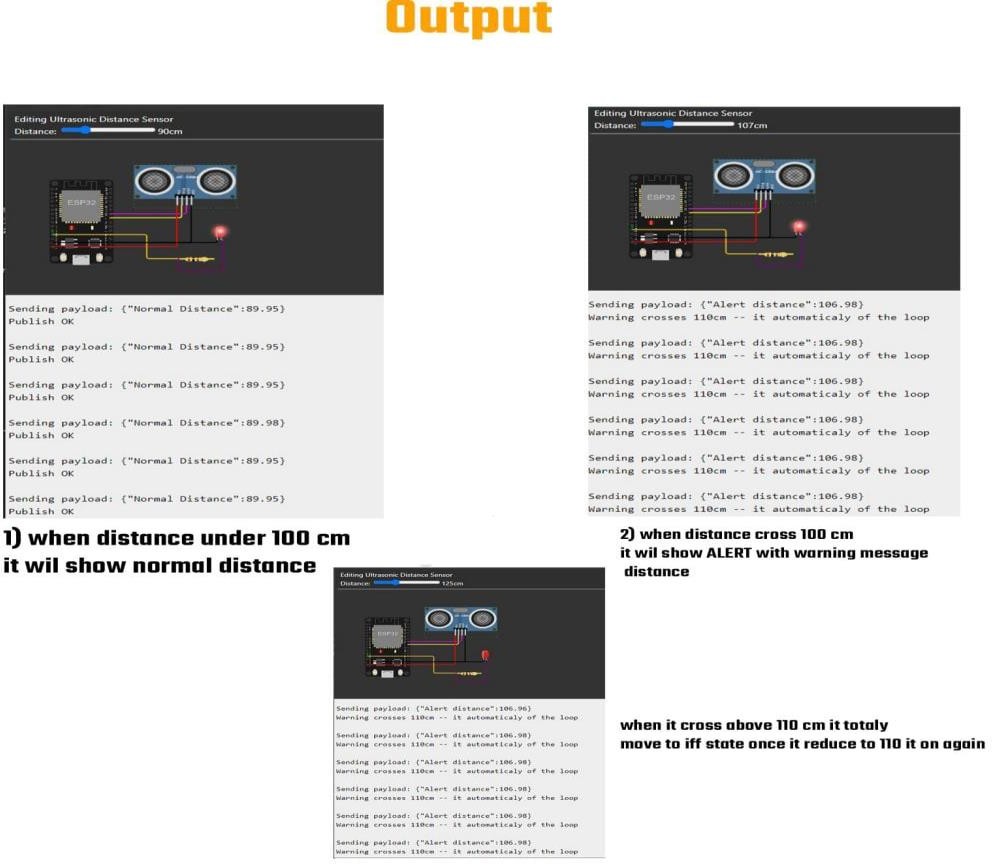
}

Serial.println(“data:”+ data3); If(data3==”lighton”){ Serial.println(data3); digitalWrite(led,HIGH);

}

Data3=””;

}



}

