

ASSIGNMENT-4

TEAM ID	PNT2022TMID52326
PROJECT NAME	SMART WASTE MANAGEMENT SYSTEM IN METROPOLITAN CITIES

CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribtopic, byte* payload, unsigned int
payloadLength);
#define ORG "tapwwl"//IBM ORGANITION ID
#define DEVICE_TYPE "ultrasonic"
#define DEVICE_ID "123456"
#define TOKEN "12345678" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribtopic[] = "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;
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;
    Serial.print("Distance (cm): ");
    Serial.println(distance);
    if(distance<100) {
        Serial.println("ALERT!!");
        delay(1000);
        PublishData(distance);
        delay(1000);
        if (!client.loop()) {
            mqttconnect();
        }
    }
    delay(1000);
}

void PublishData(float dist)
{
    mqttconnect();
    String payload = "{\"Distance\": ";
    payload += dist;
    payload += ", \"ALERT!!\": \"\"Distance less than 100cms\"\"";
    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="";
}

```

SIMULATION

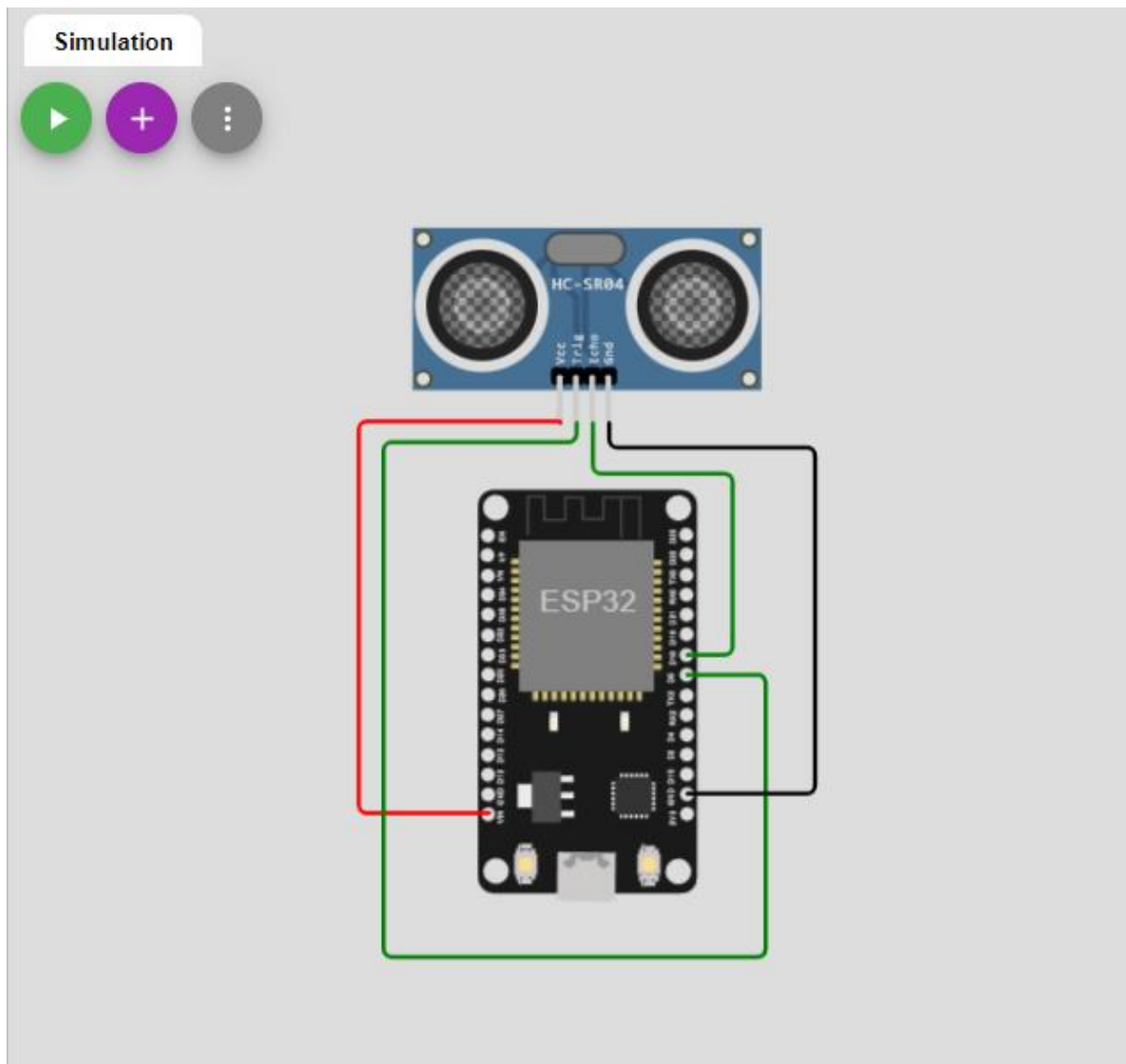


Diagram.json

```
1 {
2   "version": 1,
3   "author": "Ajees",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 59.33, "left": 5.33, "attrs": {} },
7     { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -66.04, "left": -26.83, "attrs": {} }
8   ],
9   "connections": [
10    [ "esp:TX0", "$serialMonitor:RX", "", [ ] ],
11    [ "esp:RX0", "$serialMonitor:TX", "", [ ] ],
12    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v-0.76", "h-97.78", "v196" ] ],
13    [
14      "ultrasonic1:TRIG",
15      "esp:D5",
16      "green",
17      [ "v9.24", "h-93.67", "v250", "h185.33", "v-138.67" ]
18    ],
19    [ "ultrasonic1:ECHO", "esp:D18", "green", [ "v24.58", "h67.77", "v90" ] ],
20    [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v11.91", "h99.88", "v152" ] ]
21  ]
22 }
```

IBM CLOUD OUTPUT

Browse

Action

Device Types

Interfaces

Add Device

Identity	Device Information	Recent Events	State	Logs
The recent events listed show the live stream of data that is coming and going from this device.				
Event	Value	Format	Last Received	
event_1	{"distance":7,"Alert":"Distance less than 10"}	json	a few seconds ago	
event_1	{"distance":9,"Alert":"Distance less than 10"}	json	a few seconds ago	
event_1	{"distance":8,"Alert":"Distance less than 10"}	json	a few seconds ago	
event_1	{"distance":9,"Alert":"Distance less than 10"}	json	a few seconds ago	

SIMULATION OUTPUT

