

ASSIGNMENT 4

ASSIGNMENT	4
Date	24 October 2022
Team ID	PNT2022TMID42279
Project Name	Smart Waste Management System for Metropolitan Cities

ASSIGNMENT 4:

Write code and connections in wokwi for ultrasonic sensors.

Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events.

Upload document with wokwi share link and images of ibm cloud

CODE:

```
#include <Wifi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

#define ORG "o1z9pr"

#define DEVICE_TYPE "raspberrypi"

#define DEVICE_ID "USE YOUR ID"

#define TOKEN "USE YOUR TOKEN"

#define speed 0.034

char server[] = ORG

".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/raspberrypi_1/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(trigpin, OUTPUT);

  pinMode(echopin, INPUT);

  wifiConnect();

  mqttConnect();
}

void loop()
{
  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("subscribe to cmd OK");
    }
    else
    {

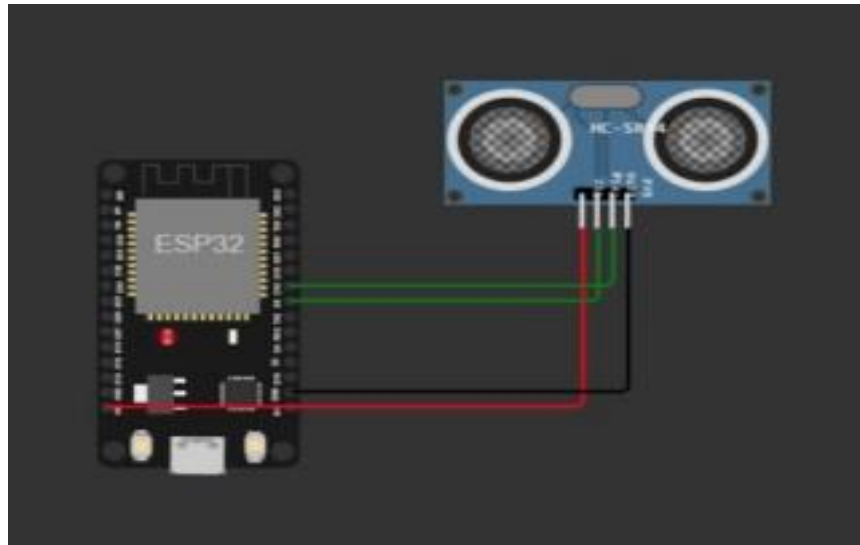
```

```

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("Publish OK");
        }
        else
        {
            Serial.println("Publish FAILED");
        }
    }
}

```

CONNECTIONS:



WOKWI LINK:

<https://wokwi.com/projects/346391868501656148>

OUTPUT:

```
sketch.ino  diagram.json  libraries.txt  Library Manager  Simulation
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4
5 WiFiClient wifiClient;
6
7 #define ORG "ol19pr"
8 #define DEVICE_TYPE "raspberrypi1"
9 #define DEVICE_ID "12345"
10 #define TOKEN "12345678"
11 #define speed 0.034
12
13 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/raspberrypi1/fmt/json";
15 char topic[] = "iot-2/cmd/home/fmt/String";
16 char authMethod[] = "use-token-auth";
17 char token[] = TOKEN;
18 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
19 PubSubClient client(server, 1883, wifiClient);
20 void publishData();
21
22 const int trigPin=5;
23 const int echoPin=4;
24 String command;
25 String date="";
26 String lat="14.167580";
27 String lon="88.248518";
28 String name="point2";
29 String icon="";
30
31 Connecting to Wifi...Wifi connected, IP address: 10.10.0.2
32 Reconnecting MQTT client to
33 ol19pr.messaging.internetofthings.ibmcloud.com
34 1
35 subscribe to cmd OK
```