

Assignment -4

Team ID	PNT2022TMID28587
Student Name	AJAY ASLEEN J
Student Roll Number	312819106002
Project Name	Smart Waste Management System for Metropolitan Cities.

Question:

Write a Code and Connections in wokwi for **ultrasonic sensor**. Whenever distance is less than 100 cms send “**alert**” to ibm cloud and display in device recent events.

Code :

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQTT
WiFiClient wifiClient;
String data3;
#define ORG "0x5bsz"
#define DEVICE_TYPE "Arduino"
#define DEVICE_ID "234566"
#define TOKEN "87654321"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Arduino/fmt/json";
char topic[] = "iot-2/cmd/status/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);

const int trigpin=19;
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 = "{\"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");
        }

    }
    if(dist>100){
        String payload = "{\"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");
        }
    }

}

}

```

OUTPUT:

1) When distance is less than 100 cm

The screenshot displays the Wokwi IoT simulation interface. On the left, the 'sketch.ino' file contains the following code:

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include <Arduino.h>
4 #include <Wire.h>
5 #define ORG "2bzw08"
6 #define DEVICE_TYPE "Arduino"
7 #define DEVICE_ID "3128"
8 #define TOKEN "12345678"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Arduino/fmt/json";
13 char topic[] = "iot-2/cmd/status/fmt/string";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wificlient);
18
19
20
21 const int trigpin=19;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
32 {
33   Serial.begin(115200);
34   pinMode(led, OUTPUT);
35   pinMode(trigpin, OUTPUT);
36   pinMode(echopin, INPUT);
37   wifiConnect();
38   mqttConnect();
39 }
```

The simulation window on the right shows an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The output of the simulation shows the following messages:

```
Publish OK
Sending payload: {"Alert Distance":86.96}
Publish OK
Sending payload: {"Alert Distance":86.96}
Publish OK
```

IBM RECENT EVENTS:

The screenshot displays the IBM Watson IoT Platform dashboard. The 'Recent Events' tab is selected, showing a table of events for device 3128. The table has the following columns: Event, Value, Format, and Last Received. The events are as follows:

Event	Value	Format	Last Received
Arduino	Alert Distance:86.96	json	a few seconds ago
Arduino	Alert Distance:86.96	json	a few seconds ago
Arduino	Alert Distance:86.96	json	a few seconds ago
Arduino	Alert Distance:86.96	json	a few seconds ago
Arduino	Alert Distance:86.96	json	a few seconds ago

A status message at the bottom right indicates '1 Simulation running'.

2) When distance is greater than 100 cm

The screenshot displays the Wokwi IoT Platform interface. On the left, the 'sketch.ino' file contains the following code:

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 WiFiClient wifiClient;
4 String data;
5 #define ORG "2bzw08"
6 #define DEVICE_TYPE "Arduino"
7 #define DEVICE_ID "3128"
8 #define TOKEN "12345678"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Arduino/fmt/json";
13 char topic[] = "iot-2/cmd/status/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wifiClient);
18
19
20
21 const int trigpin=19;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
32 {
33   Serial.begin(115200);
34   pinMode(led, OUTPUT);
35   pinMode(trigpin, OUTPUT);
36   pinMode(echopin, INPUT);
37   wifiConnect();
38   setConnect();
39 }
```

The simulation window on the right shows a physical representation of the ESP32 and HC-SR04 sensor connected by wires. Below the simulation, the terminal output shows the device sending MQTT payloads:

```
Publish OK
Sending payload: {"Distance":399.96}
Publish OK
Sending payload: {"Distance":399.92}
Publish OK
```

IBM RECENT EVENTS:

The screenshot shows the IBM Watson IoT Platform dashboard. The 'Recent Events' tab is selected, displaying a table of events for the device with ID 3128.

Event	Value	Format	Last Received
Arduino	{"Distance":399.96}	json	a few seconds ago
Arduino	{"Distance":399.96}	json	a few seconds ago
Arduino	{"Distance":399.92}	json	a few seconds ago
Arduino	{"Distance":399.98}	json	a few seconds ago
Arduino	{"Distance":399.96}	json	a few seconds ago

At the bottom of the dashboard, a status bar indicates "1 Simulation running".

WOKWI LINK:

<https://wokwi.com/projects/347102608744251988>