

# Assignment - 4

Assignment Date	26 october 2022
Student Name	D.Yaakobu Stephin
Student roll number	512419104007
MaximumMarks	2 Marks

## QUESTION - 4:

Write 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. Upload document with wokwi share link and images of ibm cloud.

## CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

WiFiClient wifiClient;

#define ORG "1bklkq"
#define DEVICE_TYPE "abcd"
#define DEVICE_ID "rasp"
#define TOKEN "12345678"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/abcd_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="";
String lat="14.167589";
String lon="80.248510";
String name="point2";
```

```

String icon="";

long duration;
int 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(1000);
        }
        initManagedDevice();
        Serial.println();
    }
}

void initManagedDevice() {

```

```

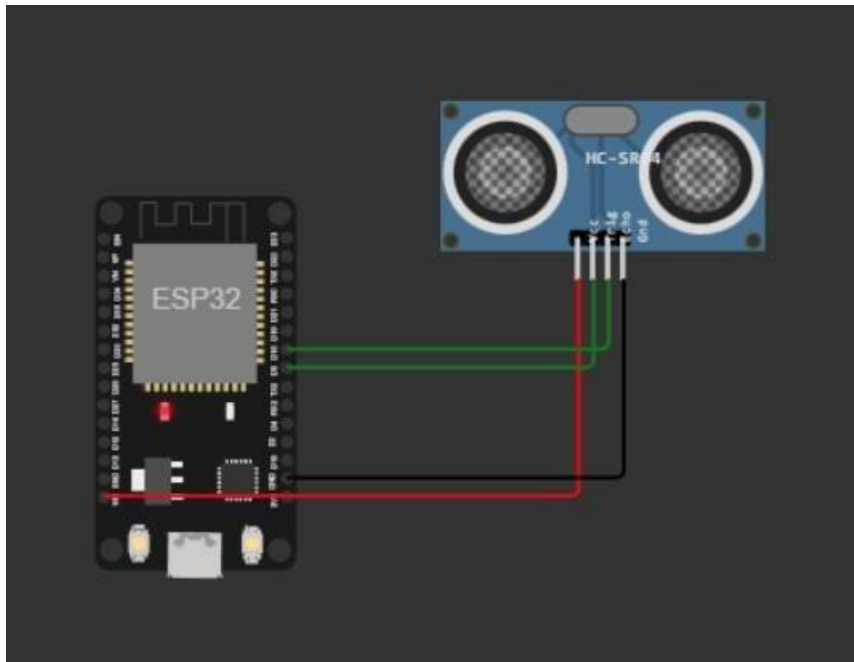
    if (client.subscribe(topic)) {
        Serial.println(client.subscribe(topic));
        Serial.println("subscribe to cmd OK");
    } else {
        Serial.println("subscribe to cmd FAILED");
    }
}

void publishData()
{
    digitalWrite(trigpin, LOW);
    digitalWrite(trigpin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin, LOW);
    duration=pulseIn(echopin, HIGH);
    dist=duration*speed/2;

    if(dist<100){
        dist=100-dist;
        icon="fa-trash";
    }else{
        dist=0;
        icon="fa-trash-o";
    }
    DynamicJsonDocument doc(1024);
    String payload;
    doc["Name"]=name;
    doc["Latitude"]=lat;
    doc["Longitude"]=lon;
    doc["Icon"]=icon;
    doc["FillPercent"]=dist;
    serializeJson(doc, payload);
    delay(3000);
    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/346587874175484499>

## OUTPUT:

**Sketch Code:**

```

1 #include <PubSubClient.h>
2 #include <ArduinoJson.h>
3
4 WiFiClient wifiClient;
5
6 #define ORG "1bk1kq"
7 #define DEVICE_TYPE "abcd"
8 #define DEVICE_ID "rasp"
9 #define TOKEN "12345678"
10 #define speed 0.034
11
12 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
13 char publishTopic[] = "iot-2/evt/abcd_1/fmt/json";
14 char topic[] = "iot-2/cmd/home/fmt/String";
15 char authMethod[] = "use-token-auth";
16 char token[] = TOKEN;
17 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
18 PubSubClient client(server, 1883, wifiClient);
19 void publishData();
20
21
22 const int trigPin=5;
23 const int echoPin=18;
24 String command;
25 String data="";
26 String lat="14.167589";
27 String lon="80.248510";
28 String name="point2";
29 String icon="";
30
31 long duration;
32 int dist;
33
34 void setup()
35 {
36   Serial.begin(115200);
37 }
38
39 void loop()
40 {
41   // Send data to the cloud
42   publishData();
43   // Read data from the cloud
44   if (client.available()) {
45     String message = client.readString();
46     // Parse the message
47     DynamicJsonDocument doc(1024);
48     deserializeJson(doc, message);
49     JsonObject root = doc.as<JsonObject>();
50     String name = root["Name"];
51     float latitude = root["Latitude"];
52     float longitude = root["Longitude"];
53     String icon = root["Icon"];
54     float fillPercent = root["FillPercent"];
55     // Print the data
56     Serial.println("Received: " + name + ", Latitude: " + latitude + ", Longitude: " + longitude + ", Icon: " + icon + ", FillPercent: " + fillPercent);
57   }
58 }
59
60 void publishData()
61 {
62   // Create a JSON object
63   DynamicJsonDocument doc(1024);
64   JsonObject root = doc.to<JsonObject>();
65   root["Name"] = "point2";
66   root["Latitude"] = latitude;
67   root["Longitude"] = longitude;
68   root["Icon"] = icon;
69   root["FillPercent"] = fillPercent;
70   // Convert the JSON object to a string
71   String message = serializeJson(root);
72   // Publish the message
73   client.publish(publishTopic, message);
74   // Wait for the message to be published
75   delay(1000);
76 }

```

**Simulation Output:**

```

o", "FillPercent":0}
Publish OK

Sending payload:
{"Name":"point2","Latitude":"14.167589","Longitude":"80.248510","Icon":"fa-trash-o", "FillPercent":0}
Publish OK

```

**IBM Watson IoT Platform Dashboard**

Device ID: abcd\_1, Status: Connected, Device Type: abcd, Class ID: Device, Date Added: Oct 26, 2022 6:53 PM

**Recent Events**

Event	Value	Format	Last Received
event_1	{"Alert Distance":83}	json	a few seconds ago
event_1	{"Alert Distance":59}	json	a few seconds ago
event_1	{"Alert Distance":7}	json	a few seconds ago
event_1	{"Alert Distance":30}	json	a few seconds ago
event_1	{"Alert Distance":51}	json	a few seconds ago