

Assignment -4
Wowki & IBM Cloud

Assignment Date	25 October 2022
Student Name	R.Sowmiya
Student Roll Number	911519106012
Maximum Marks	2 Marks

Question-1:

Write code and connections in wowki for ultrasonic sensor. Whenever the distance is less than 100 cms sent "alert" to ibm cloud and display in device recent events.

Solution:

Code:

```
#include <WiFi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG "3yngbh"
#define DEVICE_TYPE "Assignment"
#define DEVICE_ID "1234"
#define TOKEN "234567890"
#define speed 0.034
#define led 14

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/shreedharen/fmt/json";
char topic[] = "iot-2/cmd/led/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=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 = "{\"Alert Distance\":\"";
        payload += dist;
        payload += "\"}";

        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);
    }
}

```

```

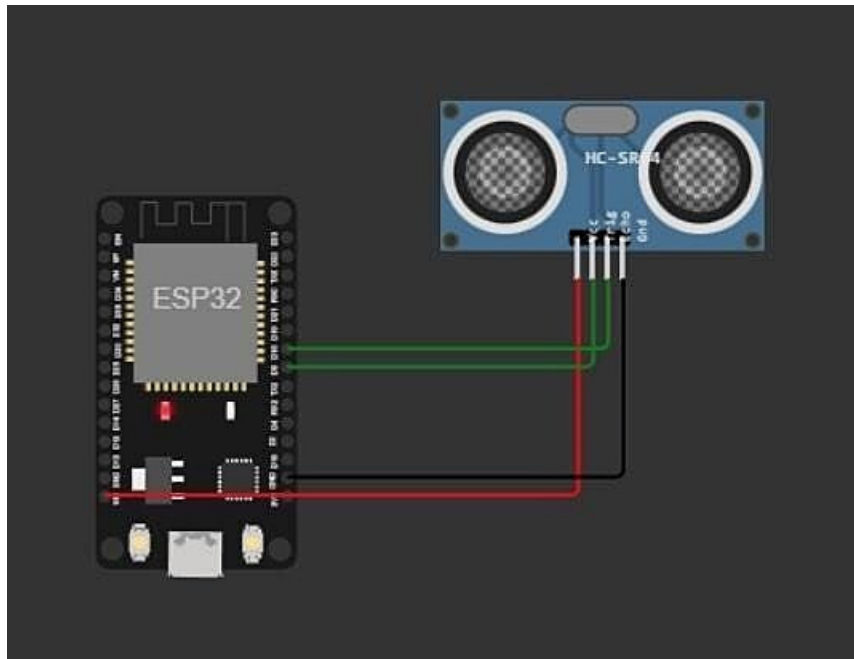
    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish OK");
    }
}

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");
    }
}
}
}

```

Connections:



Output:(wowki)

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

```
1 #include <wifi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 String data;
5 #define ORG "3yngbh"
6 #define DEVICE_TYPE "Assignment"
7 #define DEVICE_ID "1234"
8 #define TOKEN "234567890"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/shreedharen/fmt/json";
13 char topic[] = "iot-2/cmd/led/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=5;
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);
```

The simulation window on the right shows a visual representation of the ESP32 and the HC-SR04 sensor connected by wires. Below the visual, the serial output log displays the following messages:

```
Connecting to Wifi.....Wifi connected, IP address: 10.10.0.2
Reconnecting MQTT client to 3yngbh.messaging.internetofthings.ibmcloud.com
IBM subscribe to cmd OK

Sending payload: {"Distance":138.98}
Publish OK
```

The Wokwi interface also includes a top navigation bar with 'SAVE' and 'SHARE' buttons, and a right sidebar with a user profile and navigation links like 'Discord', 'My projects', 'The Club', 'Feature Roadmap', 'Language', and 'Logout'.

Link: <https://wokwi.com/projects/346405970317935188>

Output:(IBM Cloud)

Browse **Action** **Device Types** **Interfaces**

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

🔍 Search by Device ID

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
▼ <input checked="" type="checkbox"/>	12345	● Connected	NodemCU	Device	Oct 17, 2022 2:36 PM	

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
Data	{"Alert distance":93.96}	json	a few seconds ago
Data	{"Alert distance":93.96}	json	a few seconds ago
Data	{"Alert distance":93.96}	json	a few seconds ago
Data	{"Alert distance":93.96}	json	a few seconds ago
Data	{"Alert distance":93.96}	json	a few seconds ago

Items per page 100 ▼ | 1-1 of 1 item