

**Assignment -4**  
**Wokwi & IBM Cloud**

Assignment Date	28 October 2022
Student Name	Somnath Gorai P
Student Roll Number	732219CS103
Maximum Marks	2 Marks

**Question-1:**

Write code and connections in wokwi 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 "myox5t"
#define DEVICE_TYPE "ultrasonic_sensor"
#define DEVICE_ID "ultrasonic"
#define TOKEN "7708981161"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/manimd/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 += "\"}";

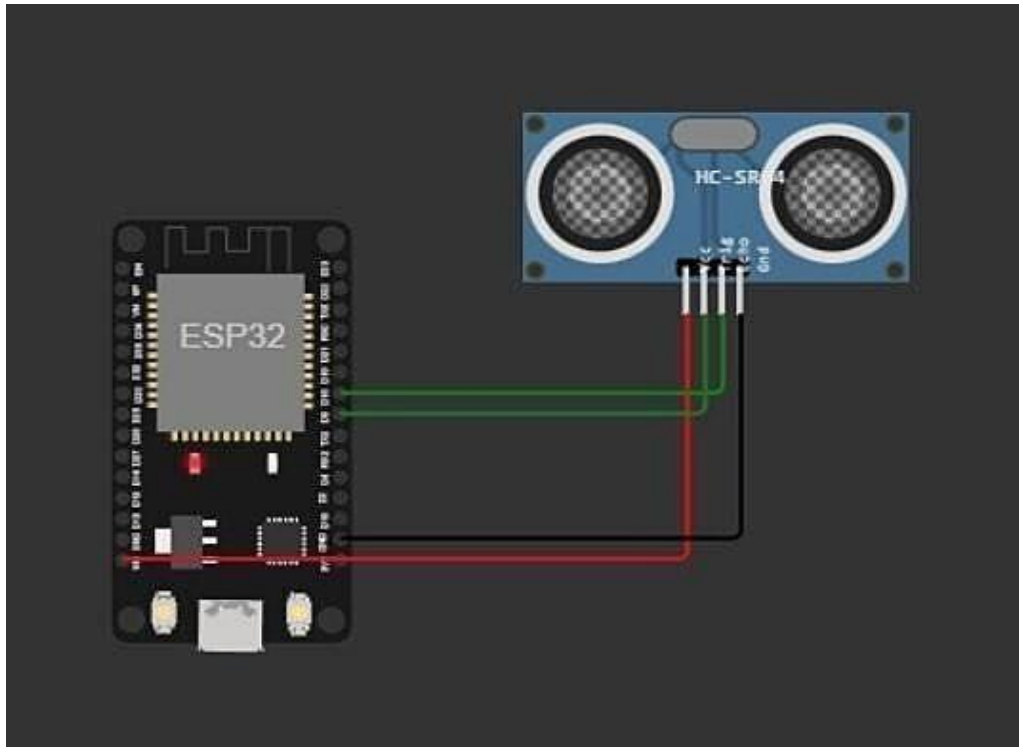
    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);
    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish OK");
    }
}

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:(wokwi):

The screenshot shows the Wokwi simulation interface. On the left, the code editor displays the following code:

```
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 }
40
41 void loop() {
42   bool isNearby = dist < 100;
43   digitalWrite(led, isNearby);
44   publishData();
45   delay(500);
46
47   if (!client.connected()) {
48     mqttConnect();
49   }
50 }
51 }
```

On the right, the simulation console shows the following output:

```
01:13.363 100%
Sending payload: {"Alert Distance":44.98}
Publish OK
Sending payload: {"Alert Distance":44.97}
Publish OK
Sending payload: {"Alert Distance":44.97}
Publish OK
Sending payload: {"Alert Distance":44.97}
Publish OK
Sending payload: {"Alert Distance":44.97}
Publish OK
Sending payload: {"Alert Distance":44.97}
Publish OK
Sending payload: {"Alert Distance":44.97}
Publish OK
Sending payload: {"Alert Distance":44.97}
Publish OK
```

Link: <https://wokwi.com/projects/347142981438931538>

Output:(IBM Cloud)

The screenshot displays the IBM Watson IoT Platform interface. At the top, there's a navigation bar with tabs for 'Service Details - IBM Cloud', 'IBM Watson IoT Platform', 'Your IBM Cloud verification code...', and 'IBM'. Below this, the browser address bar shows the URL 'myoxst.internetofthings.ibmcloud.com/dash-board/devices/browse'. The main header area includes the 'IBM Watson IoT Platform' logo and a user profile section for 'somonathgopal.19c103@nandhaengg.org' with ID 'myoxSt'. The left sidebar contains icons for various system functions. The main content area is titled 'Browse' and shows a table of devices. The first device listed is 'ultrasonic', which is 'Connected' and has a device type of 'ultrasonic\_sensor'. Below the device list, there's a section for 'Recent Events' for the selected device. It contains a message: 'The recent events listed show the live stream of data that is coming and going from this device.' followed by a table of events.

Event	Value	Format	Last Received
manimd	{"Alert Distance":44.98}	json	a few seconds ago
manimd	{"Alert Distance":45.03}	json	a few seconds ago
manimd	{"Alert Distance":44.97}	json	a few seconds ago
manimd	{"Alert Distance":44.97}	json	a few seconds ago
manimd	{"Alert Distance":44.97}	json	a few seconds ago