

ASSIGNMENT 4

Name : Keerthana.K

Reg No :113119UG07044

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

code:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifClient;
String data3;
#define ORG "x0fxss"
#define DEVICE_TYPE "Noder"
#define DEVICE_ID "1234"
#define TOKEN "987654321"
#define speed 0.034
#define led 14

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; char
publishTopic[] = "iot-2/evt/shanmugam_assignment4/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, wifClient);

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);  
    wifConnect();  
    mqttConnect();  
}
```

```
void loop() {
```

```
    bool isNearby = dist < 100;  
    digitalWrite(led, isNearby);
```

```
    publishData();  
    delay(500);
```

```
    if (!client.loop()) {  
        mqttConnect();  
    }
```

```
}
```

```
void wifConnect() {
```

```
    Serial.print("Connecting to "); Serial.print("Wif");  
    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:- i) When distance greater than 100 cm

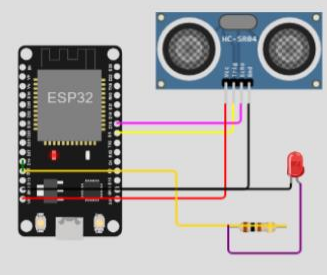
WOKWI

sketch.ino diagram.json libraries.txt Library Manager

```
1 {
2   "version": 1,
3   "author": "Keerthika J",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 92.67, "left": 45.3
7   },
8     { "type": "wokwi-led",
9       "id": "led1",
10      "top": 194.54,
11      "left": 309.26,
12      "attrs": { "color": "red" }
13    },
14    {
15      "type": "wokwi-hc-sr04",
16      "id": "ultrasonic1",
17      "top": 60.71,
18      "left": 185.64,
19      "attrs": { "distance": "139" }
20    },
21  ],
22  {
23    "type": "wokwi-resistor",
24    "id": "r1",
25    "top": 269.89,
26    "left": 260.39,
27    "attrs": { "value": "100" }
28  },
29  "connections": [
30    [ "esp:TX0", "$serialMonitor:RX", "", [ ] ],
31    [ "esp:GND", "r1:one end", "r1:other end", [ ] ],
32    [ "led1:anode", "r1:other end", "r1:one end", [ ] ],
33    [ "led1:cathode", "r1:one end", "r1:other end", [ ] ],
34    [ "ultrasonic1:VCC", "esp:5V", [ ] ],
35    [ "ultrasonic1:GND", "esp:GND", [ ] ],
36    [ "ultrasonic1:Trig", "esp:IO4", [ ] ],
37    [ "ultrasonic1:Echo", "esp:IO5", [ ] ],
38    [ "ultrasonic1:IO4", "led1:anode", [ ] ],
39    [ "ultrasonic1:IO5", "led1:cathode", [ ] ]
40  ]
41 }
```

Simulation

00:43.263 69%



Publish OK

Sending payload: {"Distance":138.96}

Publish OK

Sending payload: {"Distance":138.98}

Publish OK

IBM

IBM-Project-21602-1659785

IBM Watson IoT Platform

ASSIGNMENT 4

sketch.ino - Wokwi Arduino

9pjm1.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform

keerthikaj.cse19@veltechmultitech.org

ID: 9pjm1

Browse Action Device Types Interfaces

Add Device

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
Node	{"distance":17}	json	a few seconds ago
Node	{"distance":153}	json	a minute ago
Node	{"distance":154}	json	a minute ago
Node	{"distance":155}	json	a minute ago
Node	{"distance":170}	json	a minute ago

Items per page 50 | 1-1 of 1 item

1 Simulation running

ii) When distance less than 100 cms.

The screenshot shows the Wokwi IDE interface. On the left, the 'diagram.json' file is open, displaying a JSON configuration for a simulation. The configuration includes an ESP32 devkit, an LED, an ultrasonic sensor, and a resistor. The ultrasonic sensor is configured with a distance of 139 cm. The simulation window on the right shows a live simulation of the circuit. The 'Editing Ultrasonic Distance Sensor' dialog is open, showing a distance of 94 cm. The simulation output shows the following data:

```
Sending payload: {"Alert Distance":93.96}
Publish OK
```

```
Sending payload: {"Alert Distance":93.96}
Publish OK
```

The screenshot shows the IBM Watson IoT Platform dashboard. The 'Recent Events' tab is selected, displaying a table of events. The table has four columns: Event, Value, Format, and Last Received. The events are listed as follows:

Event	Value	Format	Last Received
Node	{"distance":91}	json	a few seconds ago
Node	{"distance":45}	json	a few seconds ago
Node	{"distance":4}	json	a few seconds ago
Node	{"distance":19}	json	a few seconds ago
Node	{"distance":94}	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

1 Simulation running