

ASSIGNMENT 4

STUDENT NAME: MADHUMITHA K

DATE	25-10-2022
TEAM ID	PNT2022TMID15198
PROJECT NAME	IoT Based Safety Gadget for Child Safety Monitoring and Notification

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

```
#include <WiFi.h> #include
<PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "c0mbt9"
#define DEVICE_TYPE "Node"
#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, 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");
    }
}
}
```

OUTPUT:

i) When distance greater than 100 cm

The screenshot shows the Wokwi IoT simulator interface. On the left, the sketch code for an ESP32 is displayed, which includes the necessary libraries and defines the MQTT broker details and sensor pins. The code is as follows:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "c0mbt9"
6 #define DEVICE_TYPE "Node"
7 #define DEVICE_ID "1234"
8 #define TOKEN "987654321"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/shanmugam_assignment4/fmt/json";
13 char topic[] = "iot-2/cmd/home/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;
```

On the right, the simulation output shows the device sending a payload to the MQTT broker:

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

Sending payload: {"Distance":399.96}
Publish OK
Reconnecting MQTT client to
x0fxss.messaging.internetofthings.ibmcloud.com
```

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

Event	Value	Format	Last Received
Node	{"distance":144}	json	a few seconds ago
Node	{"distance":182}	json	a few seconds ago
Node	{"distance":196}	json	a few seconds ago
Node	{"distance":165}	json	a few seconds ago
Node	{"distance":164}	json	a few seconds ago

At the bottom of the dashboard, a status message indicates: 1 Simulation running

ii)When distance less than 100

The screenshot shows the Wokwi IDE interface. On the left, the sketch.ino file is open, displaying C++ code for an IoT project. The code includes libraries for WiFi and PubSubClient, defines constants for the organization, device type, device ID, token, and LED pin, and sets up a server and client for MQTT communication. It also defines pins for a trigger and echo. The main loop publishes the distance value to a specific topic when the distance is less than 100 cm.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "c0mbt9"
6 #define DEVICE_TYPE "Node"
7 #define DEVICE_ID "1234"
8 #define TOKEN "987654321"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/shammugam_assignment4/fmt/json";
13 char topic[] = "iot-2/cmd/home/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 triggerpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31
```

On the right, the simulation window shows a breadboard circuit with an ultrasonic sensor, a red LED, and a buzzer. The sensor is connected to a microcontroller. The distance is currently 55cm. Below the simulation, the console shows the following output:

```
Publish OK
Sending payload: {"Alert Distance":54.94}
Publish OK
Sending payload: {"Alert Distance":54.94}
Publish OK
```

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes links to Browse, Action, Device Types, and Interfaces. The main content area displays a table of device data. The table has four columns: Event, Value, Format, and Last Received. The data shows five events, all with a value of {"distance":8} and a format of json. The last received time for all events is a few seconds ago.

Event	Value	Format	Last Received
Node	{"distance":8}	json	a few seconds ago
Node	{"distance":8}	json	a few seconds ago
Node	{"distance":5}	json	a few seconds ago
Node	{"distance":9}	json	a few seconds ago
Node	{"distance":11}	json	a few seconds ago

At the bottom of the dashboard, there is a status bar indicating "1 Simulation running".

WOKWI LINK –

<https://wokwi.com/projects/348408376977261139>