

### Assignment -3

Assignment Date	27 October 2022
Student Name	Mr. Vishva
Student Roll Number	621319205059
Maximum Marks	2 Marks

#### Question-1:

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.

#### Program:

```

#include <WiFi.h>

#include <PubSubClient.h>

//----- credentials of IBM Accounts -----

#define ORG "qwErv"

#define DEVICE_TYPE "device_1"

#define DEVICE_ID "iot"

#define TOKEN "1234567"

#define speed 0.034

#define led 14

String data3;

int LED = 4;

//----- customise above values -----

char server[] = ORG
".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-
2/evt/sreedhar/fmt/json";          //
topic name and type of event perform and
format in which data to be send

char topic[] = "iot-2/cmd/led/fmt/String";
// cmd Represent type and command is test
format of strings

char authMethod[] = "use-token-auth";
// authentication method

char token[] = TOKEN;
```

```

char clientId[] = "d:" ORG ":" DEVICE_TYPE
":" DEVICE_ID;

//-----

WiFiClient wifiClient;
// creating instance for wificlient

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();
// function call to connect to ibm

    }

}

```

```

/* -----retrieving to cloud-----*/
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("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)
    {
        digitalWrite(LED,HIGH);
        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)
{
    digitalWrite(LED,HIGH);

    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
{
    digitalWrite(LED,LOW);
    Serial.println("Publish FAILED");
}
}
}

```

WOKWI

SAVE

SHARE

esp32-dht22.ino

diagram.json

libraries.txt

Library Manager

```
1 #include <WiFi.h> //
2 #include <PubSubClient.h> // library
3
4 //----- credentials of IBM Accounts -----
5
6 #define ORG "querv" // IBM orga
7 #define DEVICE_TYPE "device_1" // Device type mentioned
8 #define DEVICE_ID "iot" // Device ID mentioned in
9 #define TOKEN "1234567" // Token
10 #define speed 0.034
11 #define led 14
12 String data;
13 int LED = 4;
14
15 //----- customise above values -----
16
17 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server
18 char publishTopic[] = "iot-2/evt/sreedhar/fmt/json";
19 char topic[] = "iot-2/cmd/led/fmt/String";
20 char authMethod[] = "use-token-auth";
21 char token[] = TOKEN;
22 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id
23
24 //-----
25
26 WiFiClient wifiClient;
27 PubSubClient client(server, 1883, wifiClient); //
28
29 const int trigpin=5;
30 const int echopin=18;
31 String command;
32 String data="";
33
34
```

Simulation

ESP32

HC-SR04

Sending payload: {"Distance":399.96}

Publish OK

Sending payload: {"Distance":399.96}

Publish OK

Sending payload: {"Distance":399.96}

Publish OK

Sending payload: {"Distance":399.96}

Publish OK