

## Assignment-4

### Question-1:

Write code and connections in wokwi for ultrasonic. When ever distance is less than 100 cm  
end "alert" to ibm cloud and display in device recent events.

Solution:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#define EchoPIN 4 // what pin we're connected to
#define TrigPIN 2
#define LED 5
//DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of
dht connected

void callback(char* subscribtopic, byte* payload, unsigned int payloadLength);

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

#define ORG "d142sf" //IBM ORGANITION ID
#define DEVICE_TYPE "abcd" //Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //Token
String data3;

//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
perform and format in which data to be send
char subscribtopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT
command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth"; // authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
float dist, dur;
String data;
//-----
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback, wifiClient); //calling the
predefined client id by passing parameter like server id, port and wificredential

void setup() // configuring the ESP32
{
```

```

Serial.begin(115200);
pinMode(TrigPIN, OUTPUT);
digitalWrite(TrigPIN, LOW);
pinMode(EchoPIN, INPUT);
pinMode(LED,OUTPUT);
delay(10);
Serial.println();
wificonnect();
mqttconnect();
}

void loop()// Recursive Function
{

    digitalWrite(TrigPIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TrigPIN, LOW);

    dur = pulseIn(EchoPIN,HIGH);

    dist= dur *0.034 / 2;
    if(dist<100)
    {
        data="alert";
        digitalWrite(LED,HIGH);

    }
    else{
        data="safe";
        digitalWrite(LED,LOW);

    }

    PublishData(dist);
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

/*.....retrieving to
Cloud.....*/

void PublishData(float dist) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form Json to update the data to ibm cloud
    */

```

```

String payload = "{\"distance\": ";
payload += dist;
payload += ", \"msg\": \"";
payload += data;
payload += "\"}";

Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it successfully upload data on the cloud
    then it will print publish ok in Serial monitor or else it will print publish
    failed
} else {
    Serial.println("Publish failed");
}

}

void mqttconnect() {
    if (!client.connected()) {
        Serial.print("Reconnecting client to ");
        Serial.println(server);
        while (!client.connect(clientId, authMethod, token)) {
            Serial.print(".");
            delay(500);
        }

        initManagedDevice();
        Serial.println();
    }
}

void wificonnect() //function defination for wificonnect
{
    Serial.println();
    Serial.print("Connecting to ");

    WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish
    the connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
}

```

```

    Serial.println(WiFi.localIP());
}

void initManagedDevice() {
    if (client.subscribe(subscribetopic)) {
        Serial.println((subscribetopic));
        Serial.println("subscribe to cmd OK");
    } else {
        Serial.println("subscribe to cmd FAILED");
    }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
}

```

WOKWI
SAVE
SHARE
sketch.ino
Docs
SIGN IN

sketch.ino
diagram.json
libraries.txt
Library Manager

```

117     }
118 }
119 void wificonnect() //function definition for wificonnect
120 {
121     Serial.println();
122     Serial.print("Connecting to ");
123
124     WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish
125     while (WiFi.status() != WL_CONNECTED) {
126         delay(500);
127         Serial.print(".");
128     }
129     Serial.println("");
130     Serial.println("WiFi connected");
131     Serial.println("IP address: ");
132     Serial.println(WiFi.localIP());
133 }
134
135 void initManagedDevice() {
136     if (client.subscribe(subscribetopic)) {
137         Serial.println((subscribetopic));
138         Serial.println("subscribe to cmd OK");
139     } else {
140         Serial.println("subscribe to cmd FAILED");
141     }
142 }
143
144 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
145 {

```

Simulation

```

Sending payload: {"distance":399.92,"msg":"safe"}
Publish ok
Sending payload: {"distance":399.94,"msg":"safe"}
Publish ok
Sending payload: {"distance":399.99,"msg":"safe"}
Publish ok
Sending payload: {"distance":399.92,"msg":"safe"}

```

IBM Watson IoT Platform

311519106111@smartinternz.com  
ID: dl42sf

← Back

Device Drilldown - 12

Connection Information

Recent Events

State

Device Information

Metadata

Diagnostics

Connection Logs

Device Actions

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"distance":399.92,"msg":"safe"}	json	a few seconds ago
Data	{"distance":399.92,"msg":"safe"}	json	a few seconds ago
Data	{"distance":399.92,"msg":"safe"}	json	a few seconds ago
Data	{"distance":399.91,"msg":"safe"}	json	a few seconds ago
Data	{"distance":399.92,"msg":"safe"}	json	a few seconds ago

0 Simulations running

