

Assignment 4:

Title:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100cms send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud.

Roll No: CITC1904120

Name: T. Varun Karthik

Wowki Link: <https://wokwi.com/projects/348055074867511890>

Source Code:

```
#include <WiFi.h>//library for wifi
#include <WiFiClient.h>
#include <PubSubClient.h>//library for MQTT
#include <ArduinoJson.h>
// creating the instance by passing pin and typr of dht connected
float distance;
#define sound_speed 0.034
int trigpin=18;
int echopin=19;
int led=5;
int LED=9;
long duration;
String message;// creating the instance by passing pin and typr of
dht connected

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

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

#define ORG "10ilxy"//IBM ORGANITION ID
#define DEVICE_TYPE "ultrasonic_mvvh"//Device type mentioned in ibm
watson IOT Platform
#define DEVICE_ID "SenSor_mvvh"//Device ID mentioned in ibm watson
IOT Platform
#define TOKEN "eCHFw6NSwFct0yY8yw" //Token
String data3;
float h, t;
```

```
//----- 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 subscribetopic[] = "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
```

```
//-----
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling
the predefined client id by passing parameter like server id,portand
wificredential
```

```
void setup()// configureing the ESP32
```

```
{
    Serial.begin(115200);
    pinMode(trigpin,OUTPUT);
    pinMode(echopin,INPUT);
    pinMode(led,OUTPUT);
    delay(10);
    Serial.println();
    wificonnect();
    mqttconnect();
}
```

```
void loop()// Recursive Function
```

```
{
    digitalWrite(trigpin,LOW);
    digitalWrite(trigpin,HIGH);
    delay(1000);
    digitalWrite(trigpin,LOW);
    duration=pulseIn(echopin,HIGH);
    distance=duration*sound_speed/2;
    Serial.println("distance"+String(distance)+"cm");
    if(distance<100)
    {
        message="Alert";
        digitalWrite(led,HIGH);
    } else
    {
        message="No problem";
        digitalWrite(led,LOW);
    }
    delay(1000);
}
```

```

    PublishData(distance,message);
    // if (!client.loop()) {
    //     mqttconnect();
    // }
}

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

void PublishData(float d, String a) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSON to update the data to ibm
cloud
    */
    DynamicJsonDocument doc(1024);
    String payload;
    doc["Distance: "]=d;
    doc["Message: "]=a;
    serializeJson(doc, payload);

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

    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish ok");// if it sucessfully 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)
{
    Serial.print("callback invoked for topic: ");
    Serial.println(subscribetopic);
    for (int i = 0; i < payloadLength; i++) {
        //Serial.print((char)payload[i]);
        data3 += (char)payload[i];
    }
    Serial.println("data: "+ data3);
    if(data3=="lighton")
    {
        Serial.println(data3);
        digitalWrite(LED,HIGH);
    }
    else
    {
        Serial.println(data3);
        digitalWrite(LED,LOW);
    }
    data3=""; }

```

Output:

WOKWI

sketch.ino

```
1 #include <WiFi.h> //library for wifi
2 #include <WiFiClient.h>
3 #include <PubSubClient.h> //library for MQTT
4 #include <ArduinoJson.h>
5 // creating the instance by passing pin and typ of dht connected
6 float distance;
7 #define sound_speed 0.034
8 int trigpin=18;
9 int echopin=19;
10 int led=5;
11 int LED=9;
12 long duration;
13 String message; // creating the instance by passing pin and typ of dht connected
14
15 void callback(char* subscribtopic, byte* payload, unsigned int payloadlength);
16
17 //-----credentials of IBM Accounts-----
18
19 #define ORG "i0ilxy" //IBM ORGANITION ID
20 #define DEVICE_TYPE "ultrasonic_mvvh" //Device type mentioned in ibm watson IOT Platform
21 #define DEVICE_ID "SenSor_mvvh" //Device ID mentioned in ibm watson IOT Platform
22 #define TOKEN "eCHFwGNSwfcT0yY8yw" //Token
23 String data3;
24 float h, t;
25
26
27 //----- Customise the above values -----
28 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
29 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform a
30 char subscribtopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND
31 char authMethod[] = "use-token-auth"; // authentication method
32 char token[] = TOKEN;
33 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
```

Simulation

Editing Ultrasonic Distance Sensor

Distance: 79cm

Publish ok

distance78.95cm

Sending payload: {"Distance": "78.94799885", "Message": "Alert"}

Publish ok

distance78.96cm

Sending payload: {"Distance": "78.96499634", "Message": "Alert"}

Publish ok

WOKWI

sketch.ino

```
1 #include <WiFi.h> //library for wifi
2 #include <WiFiClient.h>
3 #include <PubSubClient.h> //library for MQTT
4 #include <ArduinoJson.h>
5 // creating the instance by passing pin and typ of dht connected
6 float distance;
7 #define sound_speed 0.034
8 int trigpin=18;
9 int echopin=19;
10 int led=5;
11 int LED=9;
12 long duration;
13 String message; // creating the instance by passing pin and typ of dht connected
14
15 void callback(char* subscribtopic, byte* payload, unsigned int payloadlength);
16
17 //-----credentials of IBM Accounts-----
18
19 #define ORG "i0ilxy" //IBM ORGANITION ID
20 #define DEVICE_TYPE "ultrasonic_mvvh" //Device type mentioned in ibm watson IOT Platform
21 #define DEVICE_ID "SenSor_mvvh" //Device ID mentioned in ibm watson IOT Platform
22 #define TOKEN "eCHFwGNSwfcT0yY8yw" //Token
23 String data3;
24 float h, t;
25
26
27 //----- Customise the above values -----
28 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
29 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform a
30 char subscribtopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND
31 char authMethod[] = "use-token-auth"; // authentication method
32 char token[] = TOKEN;
33 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
```

Simulation

Editing Ultrasonic Distance Sensor

Distance: 180cm

distance78.96cm

Sending payload: {"Distance": "78.96499634", "Message": "Alert"}

Publish ok

distance174.93cm

Sending payload: {"Distance": "174.9299927", "Message": "No problem"}

Publish ok

distance179.96cm

IBM Watson IoT Platform

Browse Action Device Types Interfaces

1904121ece@cit.edu.in ID: 10lky

Add Device

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
SenSor_mvvh	Connected	ultrasonic_mvvh	Device	Nov 11, 2022 11:43 PM	

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
Data	{"Distance": "179.9620056","Message": "No pro..."}	json	a few seconds ago
Data	{"Distance": "179.9620056","Message": "No pro..."}	json	a few seconds ago
Data	{"Distance": "174.9299927","Message": "No pro..."}	json	a few seconds ago
Data	{"Distance": "78.96499634","Message": "Alert!"}	json	a few seconds ago
Data	{"Distance": "78.96499634","Message": "Alert!"}	json	a few seconds ago

0 Simulations running

IBM Watson IoT Platform

Readings

Line chart

5 minutes now

Distance

0 Simulations running