

**ASSIGNMENT 4**  
**WOKWI PROGRAM**

|                     |              |
|---------------------|--------------|
| Assignment Date     | 23 OCT 2022  |
| Student Name        | NAVEEN P     |
| Student Roll Number | 920219104303 |
| Maximum Marks       | 2 Marks      |

**Team ID : PNT2022TMID48488**

**PROGRAM**

**Smart Waste Management System for Metropolitan Cities**

**ASSIGNMENT 4:**

Write code and connections in wokwi for ultrasonic sensors. Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events. Uplode document with wokwi share link and images of ibm cloud.

**CODE:**

```
#include <WiFi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG "ztcz45"

#define DEVICE_TYPE "naveen"

#define DEVICE_ID "naveen123"

#define TOKEN "123456789"

#define speed 0.034

#define led 14

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

char publishTopic[] = "iot-2/evt/Data/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);
```

```
void publishData();
```

```
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);
```

```
    ...
```

```
[10:32 pm, 23/10/2022] Gogul B.E CSE: }
```

```
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 = "{ \"Normal 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>101 && dist<111){
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("Warning crosses 110cm -- it automaticaly of the loop");
    digitalWrite(led,HIGH);
}else {
    Serial.println("Publish 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++){
    dist += (char)payload[i];
}
Serial.println("data:"+ data3);
if(data3=="lighton"){
    Serial.println(data3);
}
}

```

```

digitalWrite(led,HIGH);
}

data3="";
}

```

out put:

The screenshot displays two side-by-side windows. The left window is WOKWI, an IDE for simulating IoT projects. It shows a C++ code file named 'esp32\_dht22.ino' with the following code:

```

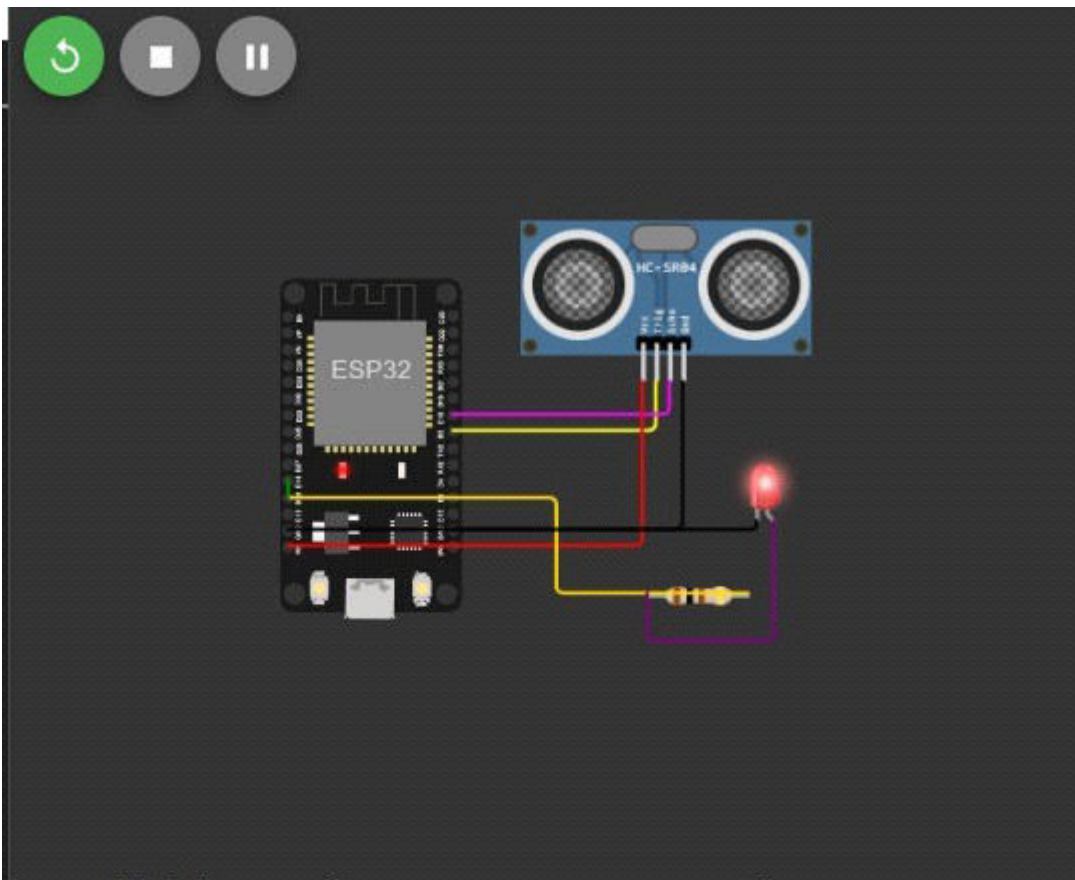
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 String data;
5 #define ONG "itc40"
6 #define DEVICE_TYPE "naven"
7 #define DEVICE_ID "naven123"
8 #define TOKEN "22456789"
9 #define SPEED 0.014
10 #define led 14
11 char server[] = "mqgw.messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "/iot-2/evt/Data/+/json";
13 char topic[] = "/iot-2/cmd/home/test/string";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "44-ONG-";
17 PubSubClient client(server, 1883, wifiClient);
18 void publishData();
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;
28
29
30 void setup()
31 {
32   Serial.begin(115200);
33   pinMode(led, OUTPUT);
34   pinMode(trigpin, OUTPUT);
35   pinMode(echopin, INPUT);
36   wifiConnect();
37   mqttConnect();
38 }
39
40 void loop() {
41   bool isNearby = dist < 100;
42   digitalWrite(led, isNearby);

```

The right window is the IBM Watson IoT Platform interface. It shows a device named 'naven123' with a status of 'Connected'. Below the device information, there is a table of recent events:

| Event | Value                     | Format | Last Received |
|-------|---------------------------|--------|---------------|
| Data  | {"Alert distance":110.94} | json   | a few second  |
| Data  | {"Alert distance":110.96} | json   | a few second  |
| Data  | {"Alert distance":110.98} | json   | a few second  |
| Data  | {"Alert distance":110.98} | json   | a few second  |
| Data  | {"Alert distance":110.96} | json   | a few second  |

1.When distance under 100 cm it wil show normal distance.



Publish OK

Sending payload: {"Normal Distance":89.95}

Publish OK

Sending payload: {"Normal Distance":89.95}

Publish OK

Sending payload: {"Normal Distance":89.95}

Publish OK

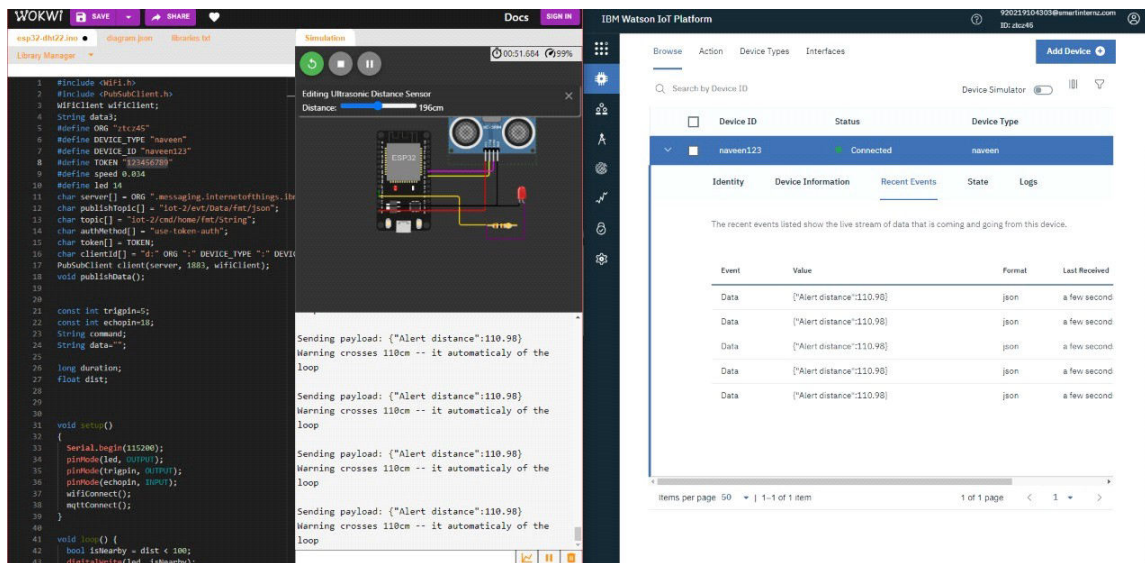
Sending payload: {"Normal Distance":89.95}

Publish OK

Sending payload: {"Normal Distance":89.95}

Publish OK

## 2. When distance cross 100 cm it wil show ALERT warning message distance



The screenshot displays two side-by-side windows. The left window is the Wokwi IDE, showing a C++ sketch for an ESP8266. The sketch includes libraries for WiFi and MQTT, and uses an ultrasonic sensor to measure distance. It sends JSON payloads to an MQTT broker when the distance is 110.98 cm. The right window is the IBM Watson IoT Platform, showing the device 'naveen123' in a 'Connected' state. The 'Recent Events' tab shows a list of events where the distance is 110.98 cm, each with a 'Warning crosses 110cm -- it automatically of the loop' message.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data;
5 #define ONG "ttcz45"
6 #define DEVICE_TYPE "naveen"
7 #define DEVICE_ID "naveen123"
8 #define TOKEN "103456789"
9 #define speed 0.004
10 #define led 14
11 char server[] = "ONG" * messaging.internetofthings.ibm
12 char publishTopic[] = "iot-2/evt/data/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:" * ONG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wificlient);
18 void publishData();
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;
28
29
30 void setup()
31 {
32   Serial.begin(115200);
33   pinMode(led, OUTPUT);
34   pinMode(trigpin, OUTPUT);
35   pinMode(echopin, INPUT);
36   wifiConnect();
37   mqttConnect();
38 }
39
40 void loop() {
41   bool isNearby = dist < 100;
42   digitalWrite(led, isNearby);
43 }
```

| Event | Value                     | Format | Last Received |
|-------|---------------------------|--------|---------------|
| Data  | {"Alert distance":110.98} | json   | a few second  |
| Data  | {"Alert distance":110.98} | json   | a few second  |
| Data  | {"Alert distance":110.98} | json   | a few second  |
| Data  | {"Alert distance":110.98} | json   | a few second  |

## 3. When it cross above 110 cm it today move to iff state once it reduce to 110 it on again

### Connection information:

Basic connnection information about this device.

Organization ID : ztcz45

Device Type : naveen

Device ID : naveen123

Authentication Method : use-token-auth

Authentication Token : 123456789

▼

naveen123

Connected

naveen

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  | {"Normal Distance":89.95} | json   | a few second: |
| Data  | {"Normal Distance":89.95} | json   | a few second: |
| Data  | {"Normal Distance":89.95} | json   | a few second: |
| Data  | {"Normal Distance":89.95} | json   | a few second: |
| Data  | {"Normal Distance":89.95} | json   | a few second: |

LINCK:

<https://wokwi.com/projects/346329704680129106>