

**ASSIGNMENT-4**  
**WOKWI SIMULATION**

Assignment Date	2nd November 2022
Student Name	S.ARJITHA
Student Roll Number	960219106034
Maximum Marks	2 Marks

**QUESTION:**

Write a code and make a connection in WOKWI for ultrasonic sensor. Whenever distance is less than 100 , send "alert" to IBM cloud and display in device recent events.

**PROGRAM:**

```
#include <WiFi.h>
#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG "d5oxwa"
#define DEVICE_TYPE "ibm-device"
#define DEVICE_ID "ibmid"
#define TOKEN "vtn5w?t3s?vX-vn8Z8"
#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/led/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:

## WOKWI SIMULATION:

The screenshot displays the WOKWI simulation interface. On the left, the 'sketch.ino' file contains the following code:

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "d5oxwa"
6 #define DEVICE_TYPE "ibm-device"
7 #define DEVICE_ID "ibmid"
8 #define TOKEN "vtn5w?ts?vX-vn8Z8"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/data/fmt/json";
13 char topic[] = "iot-2/cmd/led/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 const int trigpin=5;
19 const int echopin=18;
20 String command;
21 String data="";
22 long duration;
23 float dist;
24 void setup()
25 {
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(trigpin, OUTPUT);
29   pinMode(echopin, INPUT);

```

On the right, the 'Simulation' window shows an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The sensor's output is connected to the ESP32's digital pins. The simulation output log shows the following messages:

```

Sending payload: {"Distance":399.92}
Publish OK

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

Sending payload: {"Distance":399.94}
Publish OK

```

## When distance>100

WOKWI SAVE SHARE sketch.ino Docs

sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "d5oxwa"
6 #define DEVICE_TYPE "ibm-device"
7 #define DEVICE_ID "ibmid"
8 #define TOKEN "vt5w?3s?vX-vn8Z8"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/data/fmt/json";
13 char topic[] = "iot-2/cmd/led/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 const int trigpin=5;
19 const int echopin=18;
20 String command;
21 String data="";
22 long duration;
23 float dist;
24 void setup()
25 {
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(trigpin, OUTPUT);
29   pinMode(echopin, INPUT);
30 }
```

Simulation

00:04.899 59%

Publish OK

Sending payload: {"Distance":174.95}

Publish OK

Sending payload: {"Distance":174.95}

Publish OK

## When distance<100

WOKWI SAVE SHARE sketch.ino Docs

sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "d5oxwa"
6 #define DEVICE_TYPE "ibm-device"
7 #define DEVICE_ID "ibmid"
8 #define TOKEN "vt5w?3s?vX-vn8Z8"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/data/fmt/json";
13 char topic[] = "iot-2/cmd/led/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 const int trigpin=5;
19 const int echopin=18;
20 String command;
21 String data="";
22 long duration;
23 float dist;
24 void setup()
25 {
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(trigpin, OUTPUT);
29   pinMode(echopin, INPUT);
30 }
```

Simulation

00:25.229 57%

Sending payload: {"Alert Distance":68.95}

Publish OK

Sending payload: {"Alert Distance":68.95}

Publish OK

Sending payload: {"Alert Distance":68.97}

Publish OK

## IBM CLOUD OUTPUT:

### 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	{"Normal Distance":92.99}	json	a few seconds ago
Data	{"Normal Distance":92.99}	json	a few seconds ago
Data	{"Normal Distance":92.99}	json	a few seconds ago
Data	{"Normal Distance":92.99}	json	a few seconds ago
Data	{"Normal Distance":92.99}	json	a few seconds ago

## WOKWI LINK:

<https://wokwi.com/projects/347414646073852499>