

## ASSIGNMENT – 4

### WOKWI SIMULATION

Assignment Date	7 <sup>th</sup> November 2022
Student Name	Abina.S
Student Roll Number	960219106004
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 "hue0me"
#define DEVICE_TYPE "ibmb11m3e"
#define DEVICE_ID "ibmb11m3eid"
#define TOKEN "gUu&oxh1de07xf9l0!"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Abina/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**

Service Details - IBM Cloud | IBM Watson IoT Platform | sketchino - Wokwi Arduino and | +

wokwi.com/projects/347640643403645522

Gmail | YouTube | Maps | Untitled mural • Ajli... | IBM - ASSIGNMEN... | IBM Academic Initia... | My files - OneDrive | IBM-EPBL/IBM-Proj... | Internet of Things P... | Node-RED | Node-RED | IBM Cloud

WOKWI | SAVE | SHARE | sketchino | Docs | SIGN IN

sketchino | diagram.json | libraries.txt | Library Manager

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "hue0me"
6 #define DEVICE_TYPE "ibmb11m3e"
7 #define DEVICE_ID "ibmb11m3eid"
8 #define TOKEN "gUu8oxh1de07xf9l0!"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Abina/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   wifiConnect();
31   mqttConnect();
32 }
33 void loop() {
34   bool isNearbv = dist < 100;

```

Simulation

00:18.610 92%

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

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

Sending payload: {"Distance":399.96}

Type here to search

28°C Haze 1004 07-11-2022

## When Distance < 100:

Service Details - IBM Cloud | IBM Watson IoT Platform | sketchino - Wokwi Arduino and | +

wokwi.com/projects/347640643403645522

Gmail | YouTube | Maps | Untitled mural • Ajli... | IBM - ASSIGNMEN... | IBM Academic Initia... | My files - OneDrive | IBM-EPBL/IBM-Proj... | Internet of Things P... | Node-RED | Node-RED | IBM Cloud

WOKWI | SAVE | SHARE | sketchino | Docs | SIGN UP

sketchino | diagram.json | libraries.txt | Library Manager

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "hue0me"
6 #define DEVICE_TYPE "ibmb11m3e"
7 #define DEVICE_ID "ibmb11m3eid"
8 #define TOKEN "gUu8oxh1de07xf9l0!"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Abina/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   wifiConnect();
31   mqttConnect();
32 }
33 void loop() {
34   bool isNearbv = dist < 100;

```

Simulation

00:18.610 92%

Sending payload: {"Distance":106.98}  
Publish OK

Sending payload: {"Distance":106.98}  
Publish OK

Sending payload: {"Distance":106.98}  
Publish OK

Type here to search

28°C Haze 1004 07-11-2022

## When Distance > 100 :

The screenshot displays the Wokwi IoT Platform interface. On the left, the Arduino sketch is shown with the following code:

```
1 #include <wifi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "hue0me"
6 #define DEVICE_TYPE "ibmb11m3e"
7 #define DEVICE_ID "ibmb11m3eid"
8 #define TOKEN "glUu&oxh1de07xf9l0!"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Abina/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   wifiConnect();
31   mqttConnect();
32 }
33 void loop() {
34   bool isNearbv = dist < 100;
```

On the right, the simulation 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 status bar indicates a time of 00:18.610 and a battery level of 92%.

The console output shows the following messages:

```
Sending payload: {"Alert Distance":41.94}
Publish OK

Sending payload: {"Alert Distance":41.94}
Publish OK

Sending payload: {"Alert Distance":41.94}
Publish OK
```

## IBM CLOUD OUTPUT :

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Service Details - IBM Cloud', 'IBM Watson IoT Platform', and 'Sketchino - Wokwi Arduino and'. The main content area shows a list of devices under the 'Browse' tab. A device named 'ibmb11m3e' is selected, and its details are shown in a modal window. The modal window has tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a table of events.

Event	Value	Format	Last Received
Abina	{"Alert Distance":41.94}	json	a few seconds ago
Abina	{"Alert Distance":41.94}	json	a few seconds ago
Abina	{"Alert Distance":41.94}	json	a few seconds ago
Abina	{"Alert Distance":41.94}	json	a few seconds ago
Abina	{"Alert Distance":41.94}	json	a few seconds ago

At the bottom of the modal window, it says '0 Simulations running'.

**WOKWI LINK :**

<https://wokwi.com/projects/347639651525198420>