

# ASSIGNMENT 4

## Ultrasonic sensor simulation in Wokwi

Team ID : PNT2022TMID29941

IBM ID : IBM-Project-31889-1660205917

Question :

Write a code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100cms send an "Alert" to IBM cloud and display in the device recent events.

## Code:

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

WiFiClient wifiClient;

#define ORG "vg9s67"
#define DEVICE_TYPE "divyaselvam"
#define DEVICE_ID "divya0014ass"
#define TOKEN "12345678"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/abcd_1/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="";
String lat="14.167589";
String lon="80.248510";
String name="point2";
String icon="";
long duration;
int dist;
void setup()
{
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
```

```

}
void loop() {
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(1000);
}
initManagedDevice();
Serial.println();
}
}
void initManagedDevice() {
if (client.subscribe(topic)) {
Serial.println(client.subscribe(topic));
Serial.println("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){
dist=100-dist;
icon="fa-trash";
}else{
dist=0;
icon="fa-trash-o";
}
}

```

```

DynamicJsonDocument doc(1024);
String payload;
doc["Name"]=name;
doc["Latitude"]=lat;
doc["Longitude"]=lon;
doc["Icon"]=icon;
doc["FillPercent"]=dist;
serializeJson(doc, payload);
delay(3000);
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");
}
}

```

diagram.json:

```

{
  "version": 1,
  "author": "Kishore Annadh S",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -4, "left": -104.67, "attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -40.7, "left": 53.83, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v101.24", "h-228.44" ] ],
    [ "ultrasonic1:TRIG", "esp:D5", "green", [ "v33.9", "h-138.33" ] ],
    [ "ultrasonic1:ECHO", "esp:D18", "green", [ "v25.24", "h-145.56" ] ],
    [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v88.57", "h-152.78" ] ]
  ]
}

```

# Program&Output: Simulation

The screenshot displays the Wokwi IDE interface. On the left, the C++ code for an ESP32 is shown, which includes libraries for WiFi, PubSubClient, and ArduinoJson. The code defines an MQTT server, topic, and token, and sets up an ultrasonic sensor (HC-SR04) to publish distance measurements. The right side of the IDE shows a 3D simulation of the ESP32 and the sensor. Below the simulation, the output log shows the device sending JSON payloads to the MQTT topic.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4
5 WiFiClient wifiClient;
6
7 #define ORG "vg9s67"
8 #define DEVICE_TYPE "divyasetvam"
9 #define DEVICE_ID "divya0014ass"
10 #define TOKEN "12345678"
11 #define speed 0.034
12
13 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/abcd_1/fmt/json";
15 char topic[] = "iot-2/cmd/home/fmt/String";
16 char authMethod[] = "use-token-auth";
17 char token[] = TOKEN;
18 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
19 PubSubClient client(server, 1883, wifiClient);
20 void publishData();
21
22 const int trigpin=5;
23 const int echopin=18;
24 String command;
25 String data="";
26 String lat="14.167589";
27 String lon="80.248510";
28 String name="point2";
29 String icon="";
30
31 long duration;
32 int dist;
33
34 void setup()
35 {
36   Serial.begin(115200);
37   pinMode(trigpin, OUTPUT);
38   pinMode(echopin, INPUT);
39   wifiConnect();
40 }
```

Simulation output log:

```
{ "Name": "point2", "Latitude": "14.167589", "Longitude": "80.248510", "Icon": "fa-trash-o", "FillPercent": 0 }
Publish OK

Sending payload:
{ "Name": "point2", "Latitude": "14.167589", "Longitude": "80.248510", "Icon": "fa-trash-o", "FillPercent": 0 }
Publish OK

Sending payload:
{ "Name": "point2", "Latitude": "14.167589", "Longitude": "80.248510", "Icon": "fa-trash-o", "FillPercent": 0 }
Publish OK
```

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	
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago	
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago	
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago	
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago	
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago	

wokwi link: <https://wokwi.com/projects/346858285644644946>

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added
▼ <input type="checkbox"/>	divya0014ass	<span>●</span> Connected	divyaselvam	Device	Nov 12, 2022 12:51 PM
Identity      Device Information      Recent Events      State      Logs					
Device ID		divya0014ass			
Device Type		divyaselvam			
Date Added		Nov 12, 2022 12:51 PM			
Added By		nspgipsy@gmail.com			
Connection Status		<b>Connected</b> Connection Time: Nov 12, 2022 1:03 PM Client Address: 50.31.197.64 Insecure			