

## ASSIGNMENT – 4

Date	05-11-2022
Team ID	
Project name	Gas leakage monitoring and alerting system

### Objective:

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

### Circuit diagram:

The screenshot displays the Wokwi simulation interface. On the left, the 'Library Manager' tab is active, showing a list of libraries. The 'Sketch' tab is selected, displaying the following code:

```
2 #include <PubSubClient.h> //library for MQTT
3 #define TRIG_PIN 13
4 #define ECHO_PIN 12
5
6
7
8 void callback(char* subscribetopic, byte
9
10 //-----credentials of IBM Accounts-----
11
12 #define ORG "hg0hl1" //IBM ORGANITION ID
13 #define DEVICE_TYPE "123" //Device type m
14 #define DEVICE_ID "abcd" //Device ID ment
15 #define TOKEN "12345678" //Token
16
17
18 //----- Customise the above values --
19
20 char server[] = ORG ".messaging.internet
21 char publishTopic[] = "iot-2/evt/Data/fm
22 char subscribetopic[] = "iot-2/cmd/comma
23 char authMethod[] = "use-token-auth"; //
24 char token[] = TOKEN;
25 char clientId[] = "d:" ORG ":" DEVICE_TY
26
27
28 //-----
29 WiFiClient wifiClient; // creating the i
30 PubSubClient client(server, 1883, callba
```

On the right, the 'Simulation' tab is active, showing a circuit diagram. An ESP32 microcontroller is connected to an HC-SR04 ultrasonic sensor. The connections are as follows:

- ESP32 Pin 13 (TRIG) is connected to the TRIG pin of the HC-SR04 sensor.
- ESP32 Pin 12 (ECHO) is connected to the ECHO pin of the HC-SR04 sensor.
- ESP32 Pin 5V is connected to the VCC pin of the HC-SR04 sensor.
- ESP32 Pin GND is connected to the GND pin of the HC-SR04 sensor.

WOKWI

SAVE

SHARE

Docs

SIGN UP

sketch.ino

diagram.json

libraries.txt

Library Manager

Simulation

00:17.262 53%

```

2 #include <PubSubClient.h> //library for MQTT
3 #define TRIG_PIN 13
4 #define ECHO_PIN 12
5
6
7
8 void callback(char* subscribetopic, byte
9
10 //-----credentials of IBM Accounts-----
11
12 #define ORG "hg0hl1" //IBM ORGANITION ID
13 #define DEVICE_TYPE "123" //Device type m
14 #define DEVICE_ID "abcd" //Device ID ment
15 #define TOKEN "12345678" //Token
16
17
18 //----- Customise the above values --
19
20 char server[] = ORG ".messaging.internet
21 char publishTopic[] = "iot-2/evt/Data/fm
22 char subscribetopic[] = "iot-2/cmd/commo
23 char authMethod[] = "use-token-auth"; //
24 char token[] = TOKEN;
25 char clientId[] = "d:" ORG ":" DEVICE_TY
26
27
28 //-----
29 WiFiClient wifiClient; // creating the
30 PubSubClient client(server, 1883, callba

```

Publish ok

Sending payload: {"Distance":84.98,"MESSAGE":"ALERT"}

Publish ok

Sending payload: {"Distance":84.98,"MESSAGE":"ALERT"}

Publish ok

Sending payload: {"Distance":84.98,"MESSAGE":"ALERT"}

Publish ok

## Output:

Browse

Action

Device Types

Interfaces

Add Device

12

Connected

abcd

Device

Oct 19, 2022 10:06 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	{"MESSAGE":"ALERT"}	json	a few seconds ago
Data	{"MESSAGE":"ALERT"}	json	a few seconds ago
Data	{"MESSAGE":"ALERT"}	json	a few seconds ago
Data	{"MESSAGE":"ALERT"}	json	a few seconds ago
Data	{"MESSAGE":"ALERT"}	json	a few seconds ago

>

12

Disconnected

ZXCV

Device

Oct 20, 2022 6:11 PM