

PROBLEM STATEMENT:

IOT BASED-REAL TIME RIVER WATER
QUALITY MONITORING AND CONTROL SYSTEM

DOMAIN:

INTERNET OF THINGS

ASSIGNMENT 4:

DISTANCE DETECTION USING ULTRASOINC SENSOR

BY

KEERTHI S- (623519106014)

BANUSINDHYA I-(623519106005)

GEETHA M-(623519106008)

SOWNTHARYA S-(623519106039)

Question-1:

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100cm send “alert” to IBM cloud and display in device recent events.

WOWKI LINK:

<https://wokwi.com/projects/348209730608431698>

Solution:

```
#include <WiFi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG “csrbex”

#define DEVICE_TYPE “esp32”

#define DEVICE_ID “geetha”

#define TOKEN “geetha08”

#define speed 0.034

#define led 14

Char server[] = ORG “.messaging.internetofthings.ibmcloud.com”;

Char publishTopic[] = “iot-2/evt/manimd/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");
}else {
Serial.println("Publish FAILED");
}
}

```

Output:

The WOKWI simulation interface displays an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The code on the left defines pins and sends distance data to an IBM Watson IoT Platform. The simulation output on the right shows the sensor's readings and the data being published.

```
1 #include <WiFi.h>
2 #include <WiFiClient.h>
3 #include <PubSubClient.h>
4 const int trigPin = 5;
5 const int echoPin = 18;
6 //define sound speed in cm/uS
7 #define SOUND_SPEED 0.034
8 #define CM_TO_INCH 0.393701
9 long duration;
10 float distanceCm;
11 float distanceInch;
12
13
14 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
15 //-----credentials of IBM Accounts-----
16
17 #define ORG "csrbex"//IBM ORGANIZATION ID
18 #define DEVICE_TYPE "esp32"//Device type mentioned in ibm watson IOT Platform
19 #define DEVICE_ID "geetha"//Device ID mentioned in ibm watson IOT Platform
20 #define TOKEN "geetha08" //Token
21 String data3;
22
23
24
25 //----- Customise the above values -----
26 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
27 char publishTopic[] = "iot-2/evt/Data/Fmt/json";// topic name and type of event
28 char subscribetopic[] = "iot-2/cmd/test/Fmt/String";// cmd REPRESENT command
29 char authMethod[] = "use-token-auth";// authentication method
```

Simulation output:

```
Distance (inch): 85.42
Sending payload: {"Distance (cm)":216.97}
Publish ok
Distance (cm): 216.94
Distance (inch): 85.41
Sending payload: {"Distance (cm)":216.94}
Publish ok
```

IBM Watson IoT Platform

623519106000@smarternz.com
ID: carbox

Browse Action Device Types Interfaces

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added
geetha	Connected	esp32	Device	13 Nov 2022 02:58

Items per page 50 | 1-1 of 1 item

1 of 1 page