

Assignment-4

Date	24 October 2022
Name	AISHWARYA V
Roll Number	620119106004
Team ID	PNT2022TMID30849
Project Name	Industry-Specific intelligent fire management system

Question :

Write code and connections in wokwi for ultrasonic sensors. That whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events.

Upload document with wokwi share link and images.

Wokwi:

<https://wokwi.com/projects/348219713565753938>

Code:

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

WiFiClient wifiClient;

#define ORG "lrlttd"
#define DEVICE_TYPE "Aishu"
#define DEVICE_ID "Ece"
#define TOKEN "vRC04S*H6ZudrVdWsp"
#define speed 0.034

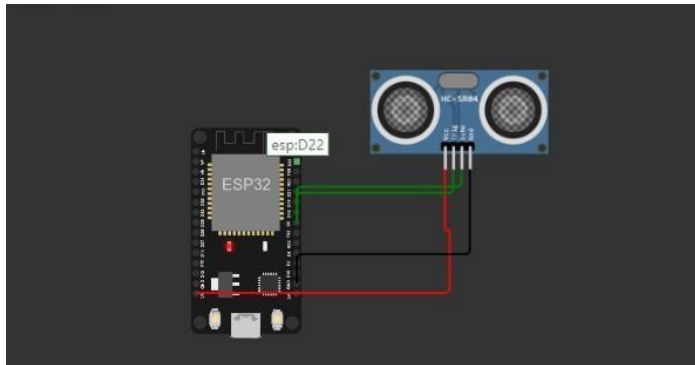
char server[] = ORG".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json"; char
topic[] = "iot-2/cmd/home/fmt/String"; char authMethod[] =
"use-token-auth"; char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient); void
publishData(); const int trigpin=5; const int echopin=18;
String command;
String data=""; long
duration; float
dist;
void
setup()
{
    Serial.begin(115200);
    pinMode(trigpin, OUTPUT);
```

```

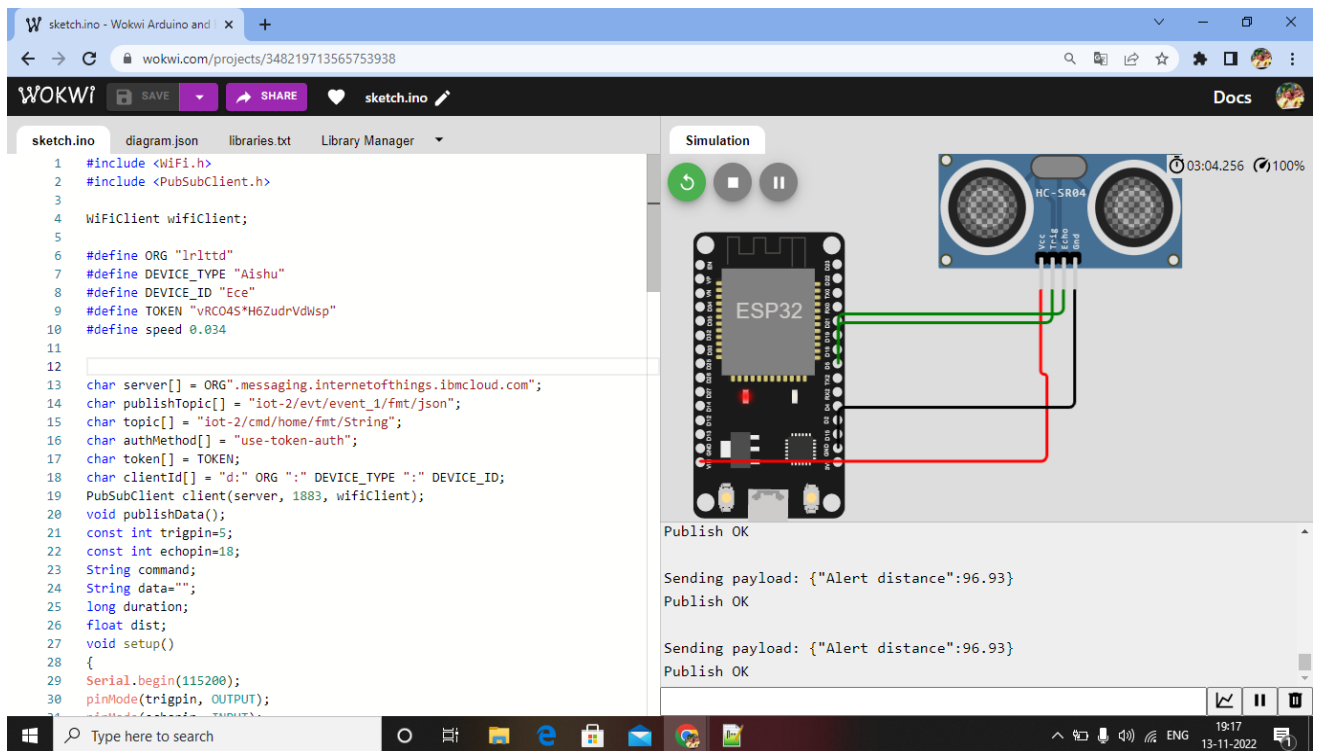
        pinMode(echopin, INPUT);
        wifiConnect(); mqttConnect();
    } void loop() {
        publishData(); delay(500);
        if (!client.loop()) {
            mqttConnect();
        }
    }
    void wifiConnect() {
        Serial.print("Connecting to "); Serial.print("Wifi");
        WiFi.begin("Wokwi-GUEST", "", 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(500);
        }
        initManagedDevice();
        Serial.println();
    } }
    void initManagedDevice() { if
        (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){
            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");
            }
        }
    }

```

Diagram:



Wokwi Output:



IBM Watson IoT Platform

lrlttd.internetofthings.ibmcloud.com/dashboard/devices/browse

viaace@gmail.com
ID: lrlttd

Browse Action Device Types Interfaces

Add Device +

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 ☒

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added
> <input type="checkbox"/>	Ece	Connected	Aishu	Device	Nov 13, 2022 6:54 PM

Items per page 50 | 1-1 of 1 item

1 of 1 page

1 Simulation running

IBM cloud output:

IBM Watson IoT Platform

lrlttd.internetofthings.ibmcloud.com/dashboard/devices/browse

viaace@gmail.com
ID: lrlttd

Browse Action Device Types Interfaces

Add Device +

Device ID: Ece Status: Disconnected Device Type: Aishu Class ID: Device Date Added: Nov 13, 2022 6:54 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
event_1	{"randomNumber":87}	json	a few seconds ago
event_1	{"randomNumber":19}	json	a few seconds ago
event_1	{"randomNumber":37}	json	a few seconds ago
event_1	{"randomNumber":31}	json	a few seconds ago
event_1	{"randomNumber":92}	json	a few seconds ago

1 Simulation running