

ASSIGNMENT-4

Assignment Date	1 november 2022
Student Name	K.RAJALAKSHMI
Student Roll Number	711619106021
Maximum Marks	2 Marks

QUESTION:

Write code and connections in wokwi for ultrasonic sensor. 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 of ibm cloud.

SOLUTION:

PROGRAM:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;

#define DEVICE_TYPE "Raji"
#define DEVICE_ID "ijar"
#define TOKEN "03261924"
#define speed 0.034
#define led 14
#define ORG "5i4ylx"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status/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(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 = "{\"Normal 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>101 && dist<111){
        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("Warning crosses 110cm -- it automatically of the loop");
            digitalWrite(led,HIGH);
        }else {

```

```

        Serial.println("Publish FAILED");
    }

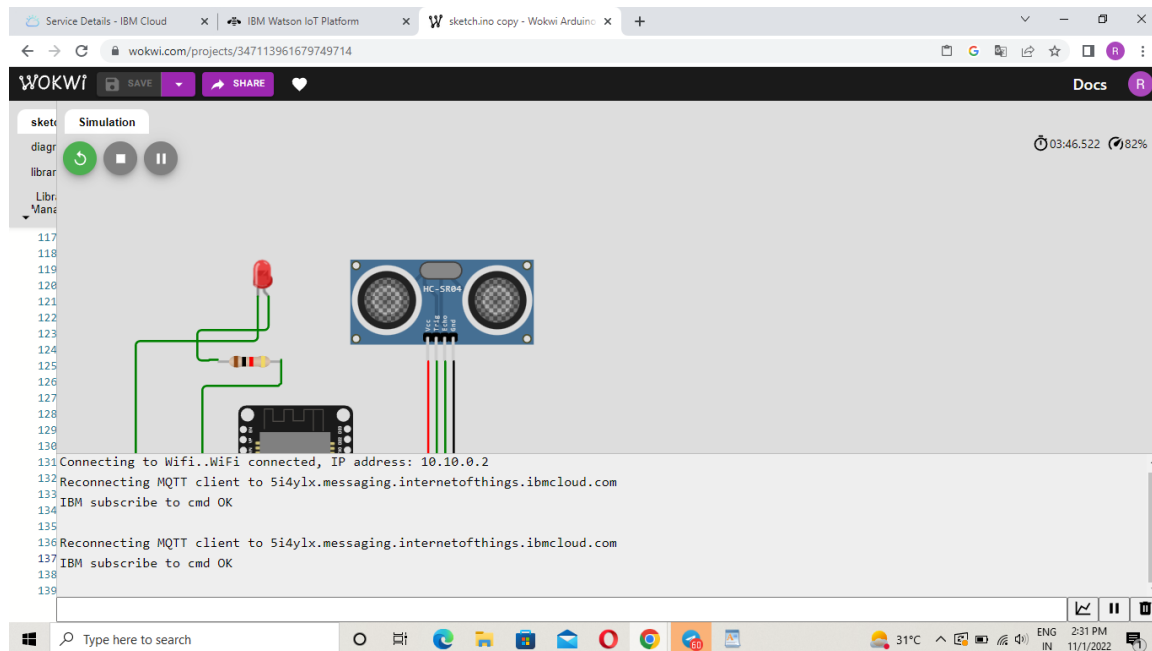
}

}

void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength){
    Serial.print("callback invoked for topic:");
    Serial.println(subscribeTopic);
    for(int i=0; i<payloadLength; i++){
        dist += (char)payload[i];
    }
    Serial.println("data:"+ data3);
    if(data3=="lighton"){
        Serial.println(data3);
        digitalWrite(led,HIGH);
    }
    data3="";
}
}

```

SIMULATION SCREENSHOTS:



The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main heading is 'Browse Devices'. Below this, there's a search bar and a table of devices. The table has columns for Device ID, Status, Device Type, Class ID, and Date Added. A single device, 'Raji', is listed with a status of 'Connected'. Below the table, there's a section for 'Recent Events' showing a stream of data. The events are listed in a table with columns for Event, Value, Format, and Last Received. The events show a status of 'Alert distance:103.96' in JSON format, received a few seconds ago. The bottom of the screen shows a Windows taskbar with various application icons and system information.

IBM Watson IoT Platform

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

Device ID	Status	Device Type	Class ID	Date Added
ijar	Connected	Raji	Device	Oct 31, 2022 11:26 AM

Items per page 50 | 1-1 of 1 item

1 of 1 page

0 Simulations running

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Device

ijar Connected Raji Device Oct 31, 2022 11:26 AM

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
status	{"Alert distance":103.96}	json	a few seconds ago
status	{"Alert distance":103.96}	json	a few seconds ago
status	{"Alert distance":103.96}	json	a few seconds ago
status	{"Alert distance":103.96}	json	a few seconds ago
status	{"Alert distance":103.96}	json	a few seconds ago

0 Simulations running

Wokwi Link:

<https://wokwi.com/projects/347113961679749714>