

ASSIGNMENT-4

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.

Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "4yi0vc"
#define DEVICE_TYPE "measurement_assignment"
#define DEVICE_ID "node_mcu"
#define TOKEN "123456789"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Gayathri/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 = "{\"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);
    }

}

if(dist>101 && dist<111){
    String payload = "{\"Normal Distance\":";
    payload += dist;
    payload += "}";

    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);

}

}

```

```

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="";
}

```

IBM Watson IoT Platform

Navigation: Browse | Action | Device Types | Interfaces

Device: node_mcu | Disconnected | measurement_assignment | 6 Nov 2022 22:51

Buttons: Add Device

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":107}	json	a few seconds ago
event_1	{"randomNumber":174}	json	a few seconds ago
event_1	{"randomNumber":253}	json	a few seconds ago
event_1	{"randomNumber":35}	json	a few seconds ago
event_1	{"randomNumber":290}	json	a few seconds ago

1 Simulation running

WOKWI

Navigation: diagram.json | libraries.txt | Library Manager

Sketch Editor:

```

52
53 void wifiConnect() {
54   Serial.print("Connecting to "); Serial.print("Wifi");
55   WiFi.begin("Wokwi-GUEST", "", 6);
56   while (WiFi.status() != WL_CONNECTED) {
57     delay(500);
58     Serial.print(".");
59   }
60   Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
61 }
62
63 void mqttConnect() {
64   if (!client.connected()) {
65     Serial.print("Reconnecting MQTT client to "); Serial.println(server);
66     while (!client.connect(clientId, authMethod, token)) {
67       Serial.print(".");
68       delay(500);
69     }
70     initManagedDevice();
71     Serial.println();
72   }
73 }
74
75 void initManagedDevice() {
76   if (client.subscribe(topic)) {
77     // Serial.println(client.subscribe(topic));
78     Serial.println("IBM subscribe to cmd OK");
79   } else {

```

Simulation

HC-SR04

Editing Ultrasonic Distance Sensor

Distance: 87cm

Warning crosses 110cm -- it automatically of the loop

Sending payload: {"Alert Distance":86.96}

Warning crosses 110cm -- it automatically of the loop

Sending payload: {"Alert Distance":86.96}

Warning crosses 110cm -- it automatically of the loop