

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.

Program:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "s8ov1q"
#define DEVICE_TYPE "pandies"
#define DEVICE_ID "pandies123"
#define TOKEN "123456789"
#define speed 0.034
#define led 14
char server[] = ORG
".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Pandies/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="";
}
```

SIMULATION:

The screenshot displays the Wokwi simulation environment. On the left, the code editor shows the following code:

```
32 {
33   Serial.begin(115200);
34   pinMode(led, OUTPUT);
35   pinMode(trigpin, OUTPUT);
36   pinMode(echopin, INPUT);
37   wifiConnect();
38   mqttConnect();
39 }
40
41 void loop() {
42   bool isNearby = dist < 100;
43   digitalWrite(led, isNearby);
44
45   publishData();
46   delay(500);
47
48   if (!client.loop()) {
49     mqttConnect();
50   }
51 }
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)) {
```

On the right, the simulation window shows an ESP32 microcontroller connected to an ultrasonic sensor (HC-SR04) and an LED. The simulation output console shows the following messages:

```
Warning crosses 110cm -- it automatically of the loop
Sending payload: {"Alert Distance":89.95}
Warning crosses 110cm -- it automatically of the loop
Sending payload: {"Alert Distance":89.95}
Warning crosses 110cm -- it automatically of the loop
```