

ASSIGNMENT – 4

Date	15th October 2022
Name	Atchayapriya M
Project name	Sign with smart connectivity for better road safety using IOT
Maximum marks	2 marks

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
#define ORG "arkiki"
#define DEVICE_TYPE "iot"
#define DEVICE_ID "123"
#define TOKEN "12345678"
String data3;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/distance/fmt/json";
char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientID[] = "d:"ORG":DEVICE_TYPE:DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server,1883,callback,wifiClient);

#define ECHO_PIN 2
#define TRIG_PIN 4
#define led 5

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    pinMode(led, OUTPUT);
    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);
    wificonnect();
    mqttconnect();
}
float readDistanceCM() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
```

```

    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    int duration=random(1,200);
    //Serial.println(duration);
    //duration = pulseIn(ECHO_PIN, HIGH);
    return duration ;
    //Serial.println(duration);
}

void loop() {
    float distance = readDistanceCM();
    //Serial.println(distance);

    bool isNearby = distance < 100;
    digitalWrite(led, isNearby);

    Serial.print("Measured distance: ");
    Serial.println(distance);
    if(distance<100){
        PublishData2(distance);

    }else{
        PublishData1(distance);

    }
    //PublishData(distance);
    delay(1000);
    if(!client.loop()){
        mqttconnect();
    }

    //delay(2000);
}
void PublishData1(float dist){
    mqttconnect();
    String payload= "{\"distance\": ";
    payload += dist;
    payload+="}";

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

    if(client.publish(publishTopic,(char*)payload.c_str())){
        Serial.println("publish ok");
    } else{
        Serial.println("publish failed");
    }
}

```

```

}
void PublishData2(float dist){
  mqttconnect();
  String payload= "{\"ALERT\":";
  payload += dist;
  payload+="}";

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

  if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("publish ok");
  } else{
    Serial.println("publish failed");
  }
}
}
void mqttconnect(){
  if(!client.connected()){
    Serial.print("Reconnecting to ");
    Serial.println(server);
    while(!!!client.connect(clientID, authMethod, token)){
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void wificonnect(){
  Serial.println();
  Serial.print("Connecting to");

  WiFi.begin("Wokwi-GUEST","",6);
  while(WiFi.status()!=WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WIFI CONNECTED");
  Serial.println("IP address:");
  Serial.println(WiFi.localIP());
}

void initManagedDevice(){
  if(client.subscribe(subscribeTopic)){
    Serial.println((subscribeTopic));
    Serial.println("subscribe to cmd ok");
  }
}

```

```

    }else{
        Serial.println("subscribe to cmd failed");
    }
}

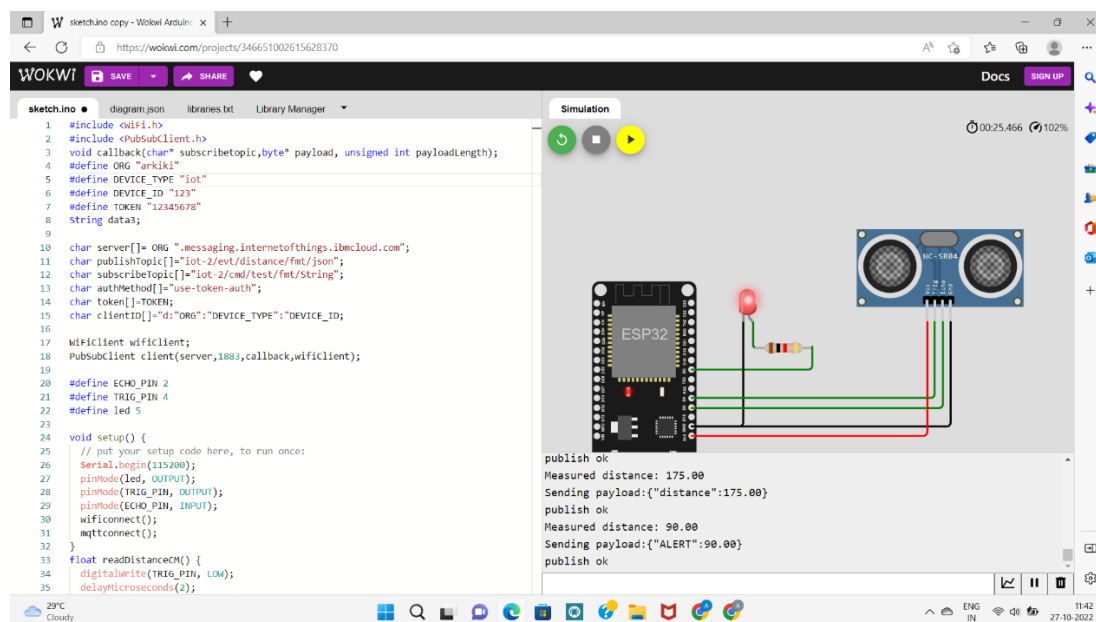
void callback(char* subscribeTopic, byte* payload, unsigned int
payloadLength){
    Serial.print("callback invoked for topic:");
    Serial.println(subscribeTopic);
    for(int i=0; i<payloadLength; i++){
        data3 += (char)payload[i];
    }
    Serial.println("data:" + data3);
    if(data3=="lighton"){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

```

Wokwi project link:

<https://wokwi.com/projects/346651002615628370>

Alert case:



Normal case:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* topic, byte* payload, unsigned int payloadLength);
4 #define ORG "arkiki"
5 #define DEVICE_TYPE "iot"
6 #define DEVICE_ID "123"
7 #define TOKEN "12345678"
8 String data3;
9
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/distance/fmt/json";
12 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;
16
17 WiFiClient wifiClient;
18 PubSubClient client(server, 1883, callback, wifiClient);
19
20 #define ECHO_PIN 2
21 #define TRIG_PIN 4
22 #define led 5
23
24 void setup() {
25   // put your setup code here, to run once:
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(TRIG_PIN, OUTPUT);
29   pinMode(ECHO_PIN, INPUT);
30   wifiConnect();
31   mqttConnect();
32 }
33
34 float readDistanceCM() {
35   digitalWrite(TRIG_PIN, LOW);
36   delayMicroseconds(2);
37
38   digitalWrite(TRIG_PIN, HIGH);
39   delayMicroseconds(10);
40   digitalWrite(TRIG_PIN, LOW);
41
42   while (digitalRead(ECHO_PIN) == LOW) {}
43   long duration = pulseIn(TRIG_PIN, HIGH);
44   float distance = duration * 0.034 / 2;
45   Serial.print("Measured distance: ");
46   Serial.print(distance);
47   Serial.println("cm");
48
49   String payload = "{\"ALERT\":\"" + String(distance) + "\"}";
50   Serial.print("Sending payload:");
51   Serial.println(payload);
52   client.publish(publishTopic, payload);
53   Serial.println("publish ok");
54 }
```

publish ok
Measured distance: 47.00
Sending payload:{"ALERT":47.00}
publish ok
Measured distance: 115.00
Sending payload:{"distance":115.00}
publish ok

Cloud storage:

IBM Watson IoT Platform

Device ID: 123, Status: Connected, Device Type: iot, Class ID: Device, Date Added: Oct 27, 2022 11:40 AM

Recent Events

Event	Value	Format	Last Received
distance	{"ALERT":42}	json	a few seconds ago
distance	{"distance":179}	json	a few seconds ago
distance	{"distance":188}	json	a few seconds ago
distance	{"ALERT":23}	json	a few seconds ago
distance	{"ALERT":55}	json	a few seconds ago