

Assignment - 4

Assignment Date	02 November 2022
Student Name	Ms.M.Suganya
Student Roll Number	821919104028
Maximum Marks	2 marks

Question -1:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100cms send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud.

Wowki Link:

<https://wokwi.com/projects/347845630485332562>

Code:

```
1  #include <WiFi.h>
2  #include <PubSubClient.h>
3  WiFiClient wifiClient;
4  String data3;
5  #define ORG "jrbl5n"
6  #define DEVICE_TYPE "Assignment4"
7  #define DEVICE_ID "12345"
8  #define TOKEN "12345678"
9  #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/data/fmt/json";
13 char topic[] = "iot-2/cmd/home/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wifiClient);
18 void publishData();
19 const int trigpin = 5;
20 const int echopin = 18;
21 String command;
22 String data = "";
23 long duration;
24 float dist;
25 void setup()
26
27 {
28   Serial.begin(115200);
```

```
29 pinMode(led, OUTPUT);
30 pinMode(trigpin, OUTPUT);
31 pinMode(echopin, INPUT);
32 wifiConnect();
33 mqttConnect();
34 }
35 void loop() {
36   bool isNearby = dist < 100;
37   digitalWrite(led, isNearby);
38   publishData();
39   delay(500);
40   if (!client.loop()) {
41     mqttConnect();
42   }
43 }
44 void wifiConnect() {
45   Serial.print("Connecting to "); Serial.print("Wifi");
46   WiFi.begin("Wokwi-GUEST", "", 6);
47   while (WiFi.status() != WL_CONNECTED) {
48     delay(500);
49     Serial.print(".");
50   }
51   Serial.print("WiFi connected, IP address: ");
52   Serial.println(WiFi.localIP());
53 }
54 void mqttConnect() {
55   if (!client.connected()) {
56     Serial.print("Reconnecting MQTT client to "); Serial.println(server);
```

```
57 while (!client.connect(clientId, authMethod, token)) {
58   Serial.print(".");
59   delay(500);
60 }
61 initManagedDevice();
62 Serial.println();
63 }
64 }
65 void initManagedDevice() {
66   if (client.subscribe(topic)) {
67     // Serial.println(client.subscribe(topic));
68     Serial.println("IBM subscribe to cmd OK");
69   } else {
70     Serial.println("subscribe to cmd FAILED");
71   }
72 }
73 void publishData()
74 {
75
76   digitalWrite(trigpin, LOW);
77   digitalWrite(trigpin, HIGH);
78   delayMicroseconds(10);
79   digitalWrite(trigpin, LOW);
80   duration = pulseIn(echopin, HIGH);
81   dist = duration * speed / 2;
82   if (dist < 100) {
83     String payload = "{\"Normal Distance\":\"";
84     payload += dist;
```

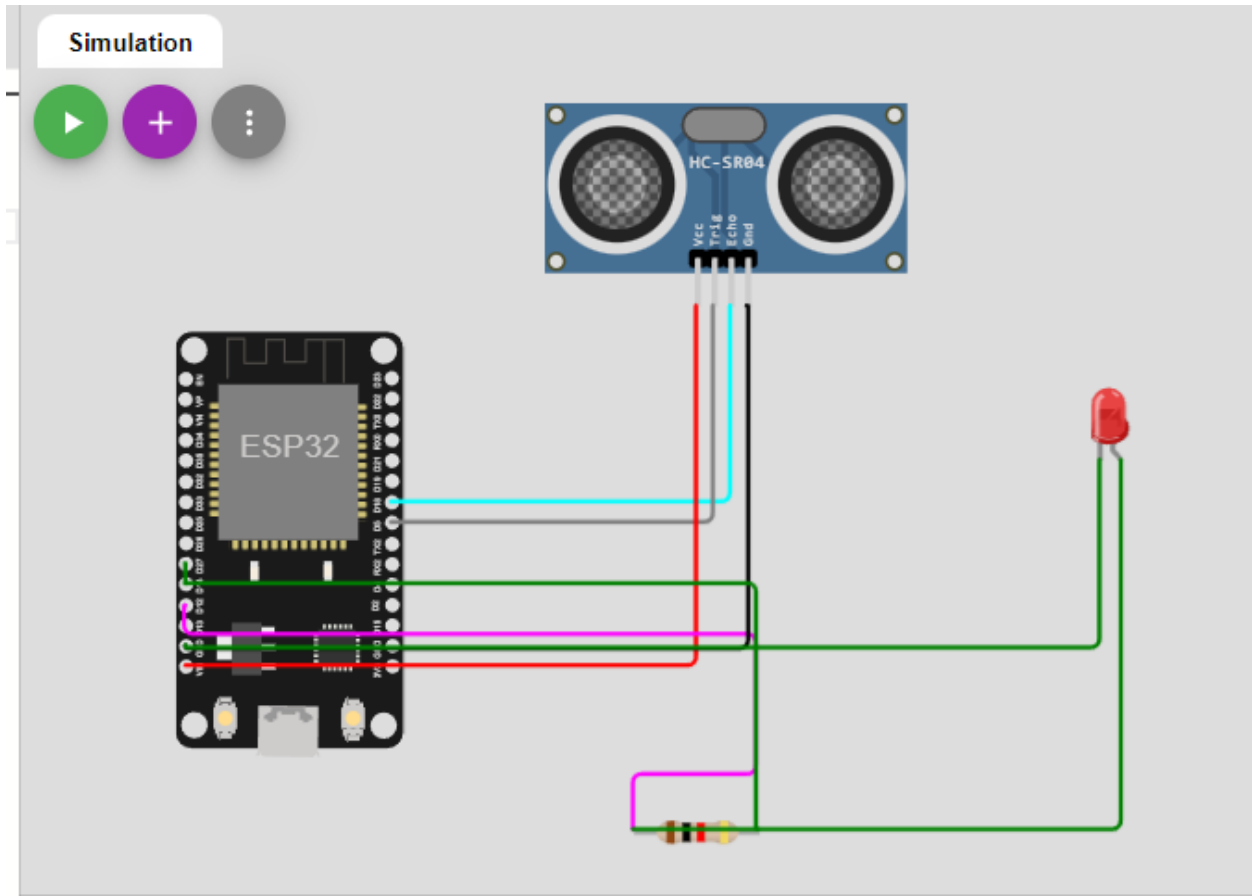
```

84     payload += "0000";
85     payload += "}";
86     Serial.print("\n");
87     Serial.print("Sending payload: ");
88     Serial.println(payload);
89     if (client.publish(publishTopic, (char*) payload.c_str())) {
90         Serial.println("Publish OK");
91     }
92 }
93 if (dist > 101) {
94     String payload = "{\\\"Alert distance\\\": ";
95     payload += dist;
96     payload += "}";
97     Serial.print("\n");
98     Serial.print("Sending payload: ");
99     Serial.println(payload);
100    if (client.publish(publishTopic, (char*) payload.c_str())) {
101        Serial.println("Warning crosses 110cm -- it automatically of the loop");
102        digitalWrite(led, HIGH);
103    } else {
104        Serial.println("Publish FAILED");
105    }
106 }
107 }
108 void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength)
109 {
110     Serial.print("callback invoked for topic:");
111     Serial.println(subscribeTopic);

112     for (int i = 0; i < payloadLength; i++) {
113         dist += (char)payload[i];
114     }
115     Serial.println("data:" + data3);
116     if (data3 == "lighton") {
117         Serial.println(data3);
118         digitalWrite(led, HIGH);
119     }
120     data3 = "";
121 }

```

Circuit Diagram:



Output:

The screenshot shows the Wokwi IDE interface with the code for the ESP32 project and the simulation output.

Code (esp32-dht22.ino):

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "zirkbl"
6 #define DEVICE_TYPE "Assignment4"
7 #define DEVICE_ID "12345"
8 #define TOKEN "12345678"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/data/fmt/json";
13 char topic[] = "iot-2/cmd/home/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wifiClient);
18 void publishData();
19 const int trigpin = 5;
20 const int echopin = 18;
21 String command;
22 String data = "";
23 long duration;
24 float dist;
25 void setup()
26 {
27   {
28     Serial.begin(115200);
29     pinMode(trigpin, OUTPUT);
30     pinMode(echopin, INPUT);
31   }
32 }
```

Simulation Output:

00:19.649 84%

Editing Ultrasonic Distance Sensor
Distance: 59cm

Sending payload: {"Normal Distance":58.99}
Publish OK

Sending payload: {"Normal Distance":58.99}
Publish OK

Reconnecting MQTT client to
zirkbl.messaging.internetofthings.ibmcloud.com

Identity
Device Information
Recent Events
State
Logs

X

Event	Value	Format	Last Received
data	{"Normal Distance":58.99}	json	a few seconds ago
data	{"Normal Distance":58.99}	json	a few seconds ago
data	{"Normal Distance":86.99}	json	a few seconds ago
data	{"Alert distance":163.95}	json	a few seconds ago
data	{"Alert distance":399.92}	json	a few seconds ago