

ASSIGNMENT-4

Date	23 October 2022
TeamID	PNT2022TMID49546
Name	Renuka P
MaximumMarks	2Marks

Question 1:

Write code and connections in work for ultrasonic sensor. Whenever distance is less than 100cms send "alert" to ibm cloud and display in device recent events.

CODE:

```
1 #include <Arduino.h> // Arduino IDE library
2 #include <PubSubClient.h> // library for mqtt
3
4
5 void callback(char* topic, byte* payload, unsigned int payloadlength);
6
7 // ... connect calls set IDB Account
8
9 #define DMC "dmcc" // IBM Organization ID
10 #define DEVICE_TYPE "ULTRASONIC" // Device Type defined in the Watson IoT Platform
11 #define DEVICE_ID "DEVICEID12345" // Device ID defined in the Watson IoT Platform
12 #define TOKEN "token7890123456789" // Token
13 #define delay 1000;
14 #define dist 100;
15
16
17 // ... initialize the above system
18 char server[] = DMC "messaging.internetofthings.ibmcloud.com"; // server name
19 char publishTopic[] = "iot2/telemetry/ultrasonic"; // topic name and type of event perform and format in which data to be send
20 char subscribeTopic[] = "iot2/alert/test/ultrasonic"; // no. subscribe command type and command in test in format string
21 char authMethod[] = "bearer token auth"; // authentication method
22 char token[] = TOKEN;
23 char clientID[] = "id" DMC "/" DEVICE_TYPE "/" DEVICE_ID; // client id
24
25
26
27 #if !defined(MQTT_CLIENT) // creating the instance for mqtt client
28 #define MQTT_CLIENT
29 #endif
30 #include <MQTTClient.h> // creating the instance for mqtt client
31 #include <PubSubClient.h> // calling the predefined client id by passing parameter like server id, port and authentication
32
33 int val = 0;
34 int trig = 0;
35 int echo = 0;
36 void setup() {
37 {
38 Serial.begin(115200);
```

```

36 pinMode(trig,OUTPUT);
37 pinMode(echo,INPUT);
38 pinMode(LED, OUTPUT);
39 delay(10);
40 wifiConnect();
41 mqttConnect();
42 }
43 void loop()// Recursive Function
44 {
45
46     digitalWrite(trig,LOW);
47     digitalWrite(trig,HIGH);
48     delayMicroseconds(10);
49     digitalWrite(trig,LOW);
50     float dur = pulseIn(echo,HIGH);
51     float dist = (dur * 0.0343)/2;
52     Serial.print ("Distance in cm");
53     Serial.println(dist);
54
55
56     PublishData(dist);
57     delay(1000);
58     if (!client.loop()) {
59         mqttConnect();
60     }
61 }
62
63
64
65 /*.....retrieving to Cloud.....*/
66
67 void PublishData(float dist) {
68     mqttConnect();//function call for connecting to ibm
69     /*
70     | creating the String in in form JSon to update the data to ibm cloud

```

```

70     | creating the String in in form JSon to update the data to ibm cloud
71     */
72     String object;
73     if (dist < 100)
74     {
75         digitalWrite(LED,HIGH);
76         Serial.println("object is near");
77         object = "near";
78     }
79     else
80     {
81         digitalWrite(LED,LOW);
82         Serial.println("no object found");
83         object = "no";
84     }
85
86     String payload = "{\"distance\":";
87     payload += dist;
88     payload += "," " \"object\":";
89     payload += object;
90     payload += "\"}";
91
92
93     Serial.print("Sending payload: ");
94     Serial.println(payload);
95
96
97
98

```

```

180 if (client.publish(subscribetopic, (char*) payload.c_str())) {
181     Serial.println("publish ok"); // if it successfully upload data on the cloud then it will print publish ok in serial monitor or else it will print publish failed
182 } else {
183     Serial.println("publish failed");
184 }
185 }
186
187 void mqttconnect() {
188     if (!client.connected()) {
189         Serial.print("Attempting client to ");
190         Serial.println(server);
191         while (!client.connect(clientid, authMethod, token)) {
192             Serial.print(".");
193             delay(500);
194         }
195     }
196     initManagedDevice();
197     Serial.println();
198 }
199
200 void wificonnect() //function definition for wifi connect
201 {
202     Serial.println();
203     Serial.print("Connecting to ");
204     Serial.println(ssid);
205     WiFi.begin(ssid, password, "", 6); //passing the wifi credentials to establish the connection
206     while (WiFi.status() != WL_CONNECTED) {
207         delay(500);
208         Serial.print(".");
209     }
210     Serial.println("");
211     Serial.println("Wifi connected");
212     Serial.println("IP address: ");
213     Serial.println(WiFi.localIP());

```

```

124 WiFi.begin("Mokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
125 while (WiFi.status() != WL_CONNECTED) {
126     delay(500);
127     Serial.print(".");
128 }
129 Serial.println("");
130 Serial.println("Wifi connected");
131 Serial.println("IP address: ");
132 Serial.println(WiFi.localIP());
133 }
134
135 void initManagedDevice() {
136     if (client.subscribe(subscribetopic)) {
137         Serial.println((subscribetopic));
138         Serial.println("subscribe to cmd OK");
139     } else {
140         Serial.println("subscribe to cmd FAILED");
141     }
142 }
143
144 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength)
145 {
146     Serial.print("callback invoked for topic: ");
147     Serial.println(subscribetopic);
148     for (int i = 0; i < payloadlength; i++) {
149         //Serial.print((char)payload[i]);
150         data3 += (char)payload[i];
151     }
152 }
153
154 // Serial.println("data: "+ data3);
155 // if(data3=="Near")
156 // {
157 // Serial.println(data3);
158 // }

```

```

142 }
143
144 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength)
145 {
146
147     Serial.print("callback invoked for topic: ");
148     Serial.println(subscribetopic);
149     for (int i = 0; i < payloadlength; i++) {
150         //Serial.print((char)payload[i]);
151         data3 += (char)payload[i];
152     }
153
154     // Serial.println("data: "+ data3);
155     // if(data3=="near")
156     // {
157     // Serial.println(data3);
158     // digitalWrite(LED,HIGH);
159
160     // }
161
162     // else
163     // {
164     // Serial.println(data3);
165     // digitalWrite(LED,LOW);
166
167     // }
168     data3="";
169
170
171 }

```

OUTPUT:

[illegible]

