

ASSIGNMENT-4

Date : 23 October 2022

Team ID : PNT2022TMID11580

Name : Madhumitha M

Maximum Marks : 2Marks

Question1: 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 <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for mqtt
3
4
5 void callback(char* topic, byte* payload, unsigned int payloadLength);
6
7 //----- credentials of IBM accounts -----
8
9 #define QMS "qmsqj" //IBM ORGANIZATION ID
10 #define DEVICE_TYPE "ULTRASONIC" //device type mentioned in the Watson IoT Platform
11 #define DEVICE_ID "DISTANCEDETECT" //device ID mentioned in the Watson IoT Platform
12 #define TOKEN "aawts7PMj7agvWdx" //token
13 String data[];
14 float dist;
15
16
17 //----- configure the above values -----
18 char server[] = QMS ".messaging.internetofthings.ibmcloud.com"; // server name
19 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform and format in which data to be send
20 char subscribeTopic[] = "iot-2/cmd/test/fmt/string"; // cmd REPRESENT command type and COMMAND IS TEST OF FORMAT STRING
21 char authMethod[] = "use-token-auth"; // authentication method
22 char token[] = TOKEN;
23 char clientId[] = "0." QMS "." DEVICE_TYPE "." DEVICE_ID //client id
24
25
26 //-----
27 WiFiClient wifiClient; // creating the instance for wifiClient
28 PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client id by passing parameter like server id, port and wifiClient
29
30 int LED = 4;
31 int trig = 5;
32 int echo = 10;
33 void setup() {
34 {
35   Serial.begin(115200);
```

```
esp32-4link.ino • diagram.json • libraries.txt • Library Manager
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.connected()) {
59     mqttConnect();
60   }
61 }
62
63
64 //----- Trying to Connect to Cloud -----
65
66 void PublishData(float dist) {
67   mqttConnect(); //function call for connecting to the
68   // creating the string in its form that to update the data to the cloud
```

```
69
70 // creating the string to its form that to update the data to the 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 = "[" + object + "," + dist + "," + "subject" + "]";
87 payload += "\n";
88 payload += object;
89 payload += "\n";
90
91 Serial.print("Sending payload: ");
92 Serial.println(payload);
93 }
```

```

124 // connect_wifi_credentials, ssid, password, pin
125 Serial.print("Waiting for WiFi...");
126 while (WiFi.status() != WL_CONNECTED) {
127   delay(500);
128   Serial.print(".");
129 }
130 Serial.println("");
131 Serial.println("WiFi connected");
132 Serial.println("IP address: ");
133 Serial.println(WiFi.localIP());
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.println("callback invoked for topic: ");
147   Serial.println(subscribetopic);
148   for (int i = 0; i < payloadlength; i++) {
149     //Serial.print((char)payload[i]);
150     data += (char)payload[i];
151   }
152   // Serial.println("data: " + data);
153   // if(data=="new")
154   // {
155   //   Serial.println(data);
156   //   digitalWrite(LED, HIGH);
157   // }
158   // else
159   // {
160   //   Serial.println(data);
161   //   digitalWrite(LED, LOW);
162   // }
163   data="";
164 }

```

```

124 WiFi.begin("ssid", "password"); //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 void initManagedDevice() {
135   if (client.subscribe(subscribetopic)) {
136     Serial.println(subscribetopic);
137     Serial.println("subscribe to cmd OK");
138   } else {
139     Serial.println("subscribe to cmd FAILED");
140   }
141 }
142
143 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength)
144 {
145   Serial.println("callback invoked for topic: ");
146   Serial.println(subscribetopic);
147   for (int i = 0; i < payloadlength; i++) {
148     //Serial.print((char)payload[i]);
149     data += (char)payload[i];
150   }
151   // Serial.println("data: " + data);
152   // if(data=="new")
153   // {
154   //   Serial.println(data);
155   //   digitalWrite(LED, HIGH);
156   // }
157   // else
158   // {
159   //   Serial.println(data);
160   //   digitalWrite(LED, LOW);
161   // }
162   data="";
163 }

```

```

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

```



