

ASSIGNMENT-4

Date : 23 October 2022

Team ID : PNT2022TMID11580

Name : Mutheeswari 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 IBM_ORG "ibm" //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 "a00ts7PWJ7agvW8a" //token
13 String data[];
14 float dist;
15
16
17 //----- customize the above values -----
18 char server[] = "MQTT.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" IBM " " 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);
36 }
```

```
esp32-4link.ino • diagram.json • libraries.txt • Library Manager
37 pinMode(trig, OUTPUT);
38 pinMode(echo, INPUT);
39 pinMode(LED, OUTPUT);
40 delay(10);
41 wifiConnect();
42 mqttConnect();
43
44 void loop() // Recursive Function
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   PublishData(dist);
56   delay(1000);
57   if (!client.connected()) {
58     mqttConnect();
59   }
60 }
61
62
63 //----- Trying to Connect to Cloud -----
64
65 void PublishData(float dist) {
66   mqttConnect(); //function call for connecting to the
67   // creating the string in its form that to update the data to the cloud
68 }
```

```
69 // creating the string in its form that to update the data to the cloud
70
71 String object;
72 if (dist < 100)
73 {
74   digitalWrite(LED, HIGH);
75   Serial.println("Object is near");
76   object = "near";
77 }
78 else
79 {
80   digitalWrite(LED, LOW);
81   Serial.println("No object found");
82   object = "no";
83 }
84
85 String payload = "[" + "distance" + ",";
86 payload += dist;
87 payload += "," + "subject" + ",";
88 payload += object;
89 payload += "]";
90
91 Serial.print("sending payload: ");
92 Serial.println(payload);
93 }
```

```

124 // connect_wifi_credentials, const char* ssid, const char* pass) {
125     Serial.print("Connecting to Wi-Fi: ");
126     while (WiFi.status() != WL_CONNECTED) {
127         delay(500);
128         Serial.print(".");
129     }
130     Serial.println("");
131     Serial.println("Wi-Fi connected");
132     Serial.println("IP address: ");
133     Serial.println(WiFi.localIP());
134 }
135
136 void initManagedDevice() {
137     if (client.subscribe(subscribetopic)) {
138         Serial.println(subscribetopic);
139         Serial.println("subscribe to cmd OK");
140     } else {
141         Serial.println("subscribe to cmd FAILED");
142     }
143 }
144
145 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength)
146 {
147     Serial.println("callback invoked for topic: ");
148     Serial.println(subscribetopic);
149     for (int i = 0; i < payloadlength; i++) {
150         //Serial.print((char)payload[i]);
151         data += (char)payload[i];
152     }
153     // Serial.println("data: " + data);
154     // if(data=="new")
155     // {
156     //     Serial.println(data);
157     //     digitalWrite(LED, HIGH);
158     // }
159     // else
160     // {
161     //     Serial.println(data);
162     //     digitalWrite(LED, LOW);
163     // }
164     data="";
165 }
166
167 void setup() {
168     // Initialize serial communication for debug
169     Serial.begin(115200);
170     while (!Serial) {
171         ; // wait for serial port to connect
172     }
173     Serial.println("ESP32-Blink.ino");
174     // Initialize the WiFi module
175     WiFi.mode(WIFI_STA);
176     WiFi.begin(ssid, pass);
177     while (WiFi.status() != WL_CONNECTED) {
178         delay(500);
179         Serial.print(".");
180     }
181     Serial.println("");
182     Serial.println("Wi-Fi connected");
183     Serial.println("IP address: ");
184     Serial.println(WiFi.localIP());
185 }
186
187 void loop() {
188     initManagedDevice();
189     callback(subscribetopic, payload, payloadlength);
190 }

```

```

124 // connect_wifi_credentials, const char* ssid, const char* pass) {
125     Serial.print("Connecting to Wi-Fi: ");
126     while (WiFi.status() != WL_CONNECTED) {
127         delay(500);
128         Serial.print(".");
129     }
130     Serial.println("");
131     Serial.println("Wi-Fi connected");
132     Serial.println("IP address: ");
133     Serial.println(WiFi.localIP());
134 }
135
136 void initManagedDevice() {
137     if (client.subscribe(subscribetopic)) {
138         Serial.println(subscribetopic);
139         Serial.println("subscribe to cmd OK");
140     } else {
141         Serial.println("subscribe to cmd FAILED");
142     }
143 }
144
145 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength)
146 {
147     Serial.println("callback invoked for topic: ");
148     Serial.println(subscribetopic);
149     for (int i = 0; i < payloadlength; i++) {
150         //Serial.print((char)payload[i]);
151         data += (char)payload[i];
152     }
153     // Serial.println("data: " + data);
154     // if(data=="new")
155     // {
156     //     Serial.println(data);
157     //     digitalWrite(LED, HIGH);
158     // }
159     // else
160     // {
161     //     Serial.println(data);
162     //     digitalWrite(LED, LOW);
163     // }
164     data="";
165 }
166
167 void setup() {
168     // Initialize serial communication for debug
169     Serial.begin(115200);
170     while (!Serial) {
171         ; // wait for serial port to connect
172     }
173     Serial.println("ESP32-Blink.ino");
174     // Initialize the WiFi module
175     WiFi.mode(WIFI_STA);
176     WiFi.begin(ssid, pass);
177     while (WiFi.status() != WL_CONNECTED) {
178         delay(500);
179         Serial.print(".");
180     }
181     Serial.println("");
182     Serial.println("Wi-Fi connected");
183     Serial.println("IP address: ");
184     Serial.println(WiFi.localIP());
185 }
186
187 void loop() {
188     initManagedDevice();
189     callback(subscribetopic, payload, payloadlength);
190 }

```

```

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 void setup() {
167     // Initialize serial communication for debug
168     Serial.begin(115200);
169     while (!Serial) {
170         ; // wait for serial port to connect
171     }
172     Serial.println("ESP32-Blink.ino");
173     // Initialize the WiFi module
174     WiFi.mode(WIFI_STA);
175     WiFi.begin(ssid, pass);
176     while (WiFi.status() != WL_CONNECTED) {
177         delay(500);
178         Serial.print(".");
179     }
180     Serial.println("");
181     Serial.println("Wi-Fi connected");
182     Serial.println("IP address: ");
183     Serial.println(WiFi.localIP());
184 }
185
186 void loop() {
187     initManagedDevice();
188     callback(subscribetopic, payload, payloadlength);
189 }

```



