

## ASSIGNMENT-4

---

Date : 23 October 2022

Team ID : PNT2022TMID11580

Name : Monika B

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 <stdio.h>/library for printf
2 #include <pubsubClient.h>/library for MQTT
3
4
5 void callback(char* topic, byte* payload, unsigned int payloadLength);
6
7 //-----Credentials of IBM Account-----
8
9 #define ONG "shopify"//IBM organization ID
10 #define DEVICE_TYPE "OUTRANCN"//device type mentioned in the Watson IoT Platform
11 #define DEVICE_ID "DISTANCEDETECT"//device ID mentioned in the Watson IoT Platform
12 #define TOKEN "watts7WZ5egv8dte"//token
13 String data;
14 float dist;
15
16
17 //-----Customise the above values-----
18 char server[] = ONG + "messaging.internetofthings.ibmcloud.com"; // Server Name
19 char pubSubClient[] = "iot-2/evt/data/req/req"; // topic name and type of event perform and format in which data to be send
20 char subScribTopic[] = "iot-2/cmd/test/req/string"; // cmd. HREF:iotfwd command type and COMMAND 25 TEXT or FORMAT STRING;
21 char authMethod[] = "use-token-auth"; // authentication method
22 char token[] = TOKEN;
23 char clientId[] = "dt" ONG "+" DEVICE_TYPE "+" DEVICE_ID"/client id
24
25
26 //-----
27 #ificient 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 wificredential
29
30 int LED = 4;
31 int trig = 5;
32 int echo = 10;
33 void setup()
34 {
35     Serial.begin(115200);

```

```
esp12-blink.ino • diagram • Libraries • Library Manager
36  pinMode(LED, OUTPUT);
37  pinMode(echo, INPUT);
38  pinMode(LED, OUTPUT);
39  delay(100);
40  myttnconnect();
41  myttnconnect();
42  }
43  void loop() // Recursive Function
44  {
45
46      digitalWrite(trig, HIGH);
47      digitalWrite(trig, LOW);
48      delayMicroseconds(100);
49      digitalWrite(trig, HIGH);
50      float dur = pulseIn(echo, HIGH);
51      float dist = (dur * 0.045) / 2;
52      Serial.print("Distance in cm");
53      Serial.println(dist);
54
55
56      PublishData(dist);
57      delay(1000);
58      if (!client.connected()) {
59          myttnconnect();
60      }
61  }
62
63  /*.....retrieving to cloud.....*/
64
65
66  void PublishData(float dist) {
67      myttnconnect(); //function call for connecting to sim
68
69      // create the String in in form this to update the data to the cloud
```

```

70 // Creating the string in an array form to update the data to the class
71 //
72 string object;
73 if (dist < 100)
74 {
75     digitalWrite(LED,LOW);
76     Serial.println("object is near");
77     object = "near";
78 }
79 else
80 {
81     digitalWrite(LED,HIGH);
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 Serial.print("sending payload: ");
93 Serial.println(payload);
94
95
96
97

```

```

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 WiFi
175     connect_wifi_credentials(ssid, pass);
176     // Initialize the client
177     initManagedDevice();
178     // Subscribe to the topic
179     subscribe(subscribetopic);
180     // Start the server
171     server.begin();
172     Serial.println("Server started");
173 }
174
175 void loop() {
176     // Check for incoming client connections
177     while (server.hasClient()) {
178         WiFiClient client = server.nextClient();
179         Serial.println("New client connected");
180         // Send the client the payload
181         client.print(data);
182     }
183 }

```

```

124 WiFi.begin(ssid, pass); //connecting 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("Wi-Fi 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 }
164
165 void setup() {
166     // Initialize serial communication for debug
167     Serial.begin(115200);
168     while (!Serial) {
169         ; // wait for serial port to connect
170     }
171     Serial.println("ESP32-Blink.ino");
172     // Initialize WiFi
173     connect_wifi_credentials(ssid, pass);
174     // Initialize the client
175     initManagedDevice();
176     // Subscribe to the topic
177     subscribe(subscribetopic);
178     // Start the server
179     server.begin();
180     Serial.println("Server started");
181 }
182
183 void loop() {
184     // Check for incoming client connections
185     while (server.hasClient()) {
186         WiFiClient client = server.nextClient();
187         Serial.println("New client connected");
188         // Send the client the payload
189         client.print(data);
190     }
191 }

```

```

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 WiFi
174     connect_wifi_credentials(ssid, pass);
175     // Initialize the client
176     initManagedDevice();
177     // Subscribe to the topic
178     subscribe(subscribetopic);
179     // Start the server
180     server.begin();
181     Serial.println("Server started");
182 }
183
184 void loop() {
185     // Check for incoming client connections
186     while (server.hasClient()) {
187         WiFiClient client = server.nextClient();
188         Serial.println("New client connected");
189         // Send the client the payload
190         client.print(data);
191     }
192 }

```



