

## ASSIGNMENT-4

Date	23 October 2022
TeamID	PNT2022TMID49546
Name	Sayee shakshi N D
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 <Wire.h> //library for i2c
2 #include <PubSubClient.h> //library for mqtt
3
4
5 void callback(char* topic, byte* payload, unsigned int payloadLength){
6
7 // ..... conditions of the payload .....
8
9 #define MSG "dmesg" //var declaration
10 #define DEVICE_TYPE "ULTRASONIC" //Device type mentioned in the output of the platform
11 #define DEVICE_ID "PNT2022TMID49546" //Device ID mentioned in the output of the platform
12 #define TOKEN "xxxxxx/xxxxxx" //Token
13
14 String delay;
15 float dist;
16
17 // ..... initialize the above values .....
18 char server[] = "mqtts://messaging.internetofthings.ibmcloud.com"; // server name
19 char publishTopic[] = "iot/1/test/data/test/json"; // topic name and type of event perform and format in which data to be send
20 char subscribeTopic[] = "iot/1/test/test/string"; // var represent common type and format in test or sensor output
21 char authMethod[] = "user-token-auth"; // authentication method
22 char token[] = "xxxxxx";
23 char clientId[] = "d" MSG "-" DEVICE_TYPE "-" DEVICE_ID; //client id
24
25
26 // ..... create the instance of the client .....
27 PubSubClient client(server, 1883, callback, clientId); // creating the instance for mqtt client
28 // ..... initialize the client .....
29
30 int led = A1;
31 int trig = S4;
32 int echo = S8;
33 void setup()
34 {
35   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 += ",";
89      payload += "\"object\":";
90      payload += object;
91      payload += "\"}";
92
93      Serial.print("Sending payload: ");
94      Serial.println(payload);
95
96
97
98

```

```

esp32-blink.ino • diagram.json • libraries.txt • Library Manager
180
181 if (client.publish(publishTopic, (char*) payload_c_str)) {
182     Serial.println("Publish OK"); // if it successfully posted data on the cloud then it will print publish ok in serial monitor or else it will print publish failed
183 } else {
184     Serial.println("Publish failed");
185 }
186
187 }
188
189 void wifiConnect() {
190     if (!client.connected()) {
191         Serial.print("Reconnecting client to ");
192         Serial.println(server);
193         while (!client.connect(server, 80, httpMethod, token)) {
194             Serial.print(".");
195             delay(500);
196         }
197
198         initManagedDevice();
199         Serial.println();
200     }
201 }
202
203 void wifiConnect() //function definition for wifiConnect
204 {
205     Serial.println();
206     Serial.print("Connecting to ");
207
208     WiFi.begin("Mokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
209     while (WiFi.status() != WL_CONNECTED) {
210         delay(500);
211         Serial.print(".");
212     }
213     Serial.println("");
214     Serial.println("Wifi connected");
215     Serial.println("IP address: ");
216     Serial.println(WiFi.localIP());
217 }

```

```

esp32-blink.ino • diagram.json • libraries.txt • Library Manager
124
125 WiFi.begin("Mokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
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
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=="None")
156     // {
157     // Serial.println(data3);
158     // }
159 }

```

```

esp32-blink.ino • diagram.json • libraries.txt • Library Manager
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:

The screenshot shows the IBM Cloud IoT Platform console. At the top, there are tabs for 'Browse', 'Action', 'Device Types', and 'Overview'. A blue button 'Add Device' is in the top right. Below the tabs, a device card for 'DISTANCE DETECT' is shown, with status 'Disconnected', type 'ULTRASONIC', and a timestamp 'Oct 20, 2022 9:46 AM'. A dropdown menu is open, showing options: 'Identity', 'Device Information', 'Recent Events' (selected), 'State', and 'Logs'. Below the menu, a message states: 'The recent events table shows the live stream of data that is coming and going from this device.' A table with 4 columns (Event, Value, Format, Last Received) displays 5 rows of data. At the bottom, there is a pagination bar showing 'Items per page: 10' and '1-2 of 2 items'.

Event	Value	Format	Last Received
Data	["distance":141.22,"object":"No"]	json	a few seconds ago
Data	["distance":141.22,"object":"No"]	json	a few seconds ago
Data	["distance":141.22,"object":"No"]	json	a few seconds ago
Data	["distance":141.28,"object":"No"]	json	a few seconds ago
Data	["distance":141.2,"object":"No"]	json	a few seconds ago

Data send to the IBMcloud device when the objectics far



