

ASSIGNMENT-4

Date	20 Oct 2022
Team ID	PNT2022TMID04692
Project Name	IoT Based Smart Crop Protection System For Agriculture
Maximum Marks	2 Marks

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 Account-----
8
9 #define ORG "ibmorg" //IBM ORGANIZATION ID
10 #define DEVICE_TYPE "ULTRASONIC" //device type mentioned in the Watson IoT Platform
11 #define DEVICE_ID "DETECT" //Device ID mentioned in the Watson IoT Platform
12 #define TOKEN "token" //token
13 String data;
14 float dist;
15
16
17 //----- Customize the above values -----
18 char server[] = ORG ".messaging.internetofthings.ibmcloud.com" // Server Name
19 char publishTopic[] = "iot-2/evt/data/fat/json" // topic name and type of event perform and format in which data to be send
20 char subscribeTopic[] = "iot-2/cmd/test/fat/String" // cmd REPRESENT command type AND COMMAND IS TEST OF PUBLISH STRING
21 char authMethod[] = "use-token-auth" // authentication method
22 char token[] = TOKEN;
23 char clientId[] = "0:" ORG ":" 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 pin = 4;
31 int trig = 5;
32 int echo = 10;
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 ("Distancein 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

```

```

185     if (client.publish(publishTopic, (char*) payload_c_str())) {
186         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
187     } else {
188         Serial.println("Publish failed");
189     }
190 }
191
192 void mqttconnect() {
193     if (!client.connected()) {
194         Serial.print("Reconnecting client to ");
195         Serial.println(server);
196         while (!client.connect(clientId, authMethod, token)) {
197             Serial.print(".");
198             delay(500);
199         }
200
201         initManagedDevice();
202         Serial.println();
203     }
204 }
205
206 void wificonnect() //function definition for wifi connect
207 {
208     Serial.println();
209     Serial.print("Connecting to ");
210
211     WiFi.begin("Mokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
212     while (WiFi.status() != WL_CONNECTED) {
213         delay(500);
214         Serial.print(".");
215     }
216     Serial.println("");
217     Serial.println("WiFi connected");
218     Serial.println("IP address: ");
219     Serial.println(WiFi.localIP());

```

```

220     WiFi.begin("Mokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
221     while (WiFi.status() != WL_CONNECTED) {
222         delay(500);
223         Serial.print(".");
224     }
225     Serial.println("");
226     Serial.println("WiFi connected");
227     Serial.println("IP address: ");
228     Serial.println(WiFi.localIP());
229 }
230
231 void initManagedDevice() {
232     if (client.subscribe(subscribetopic)) {
233         Serial.println(subscribetopic);
234         Serial.println("subscribe to cmd OK");
235     } else {
236         Serial.println("subscribe to cmd FAILED");
237     }
238 }
239
240 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
241 {
242     Serial.print("callback invoked for topic: ");
243     Serial.println(subscribetopic);
244     for (int i = 0; i < payloadLength; i++) {
245         //Serial.print((char)payload[i]);
246         data3 += (char)payload[i];
247     }
248
249     // Serial.println("data: "+ data3);
250     // if(data3=="Near")
251     // {
252     // Serial.println(data3);
253     // }

```

```
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     // else
162     // {
163     // Serial.println(data3);
164     // digitalWrite(LED,LOW);
165     // }
166     data3="";
167 }
168
169
170
171 }
```

OUTPUT:

The screenshot shows the IBM Cloud IoT Platform interface. At the top, there's a navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A blue 'Add Device' button is on the right. Below this, a device card for 'DISTANCE DETECT' is shown, indicating it is 'Disconnected' and last seen on 'Oct 20, 2022 9:45 AM'. The 'Recent Events' tab is selected, showing a table of events. The table has columns for 'Event', 'Data', 'Format', and 'Last Received'. Five events are listed, all with a 'Data' format and received 'a few seconds ago'. At the bottom, there's a pagination bar showing 'Items per page: 50' and '1 of 2 items'.

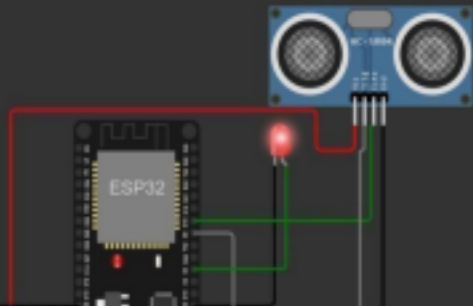
Event	Data	Format	Last Received
Data	[{"distance": 141.22, "object": "No"}]	json	a few seconds ago
Data	[{"distance": 141.22, "object": "No"}]	json	a few seconds ago
Data	[{"distance": 141.22, "object": "No"}]	json	a few seconds ago
Data	[{"distance": 141.22, "object": "No"}]	json	a few seconds ago
Data	[{"distance": 141.22, "object": "No"}]	json	a few seconds ago

Data send to the IBMcloud device when the objectics far

WIFI | [back](#) | [project](#) | [esp32-arduino.ino](#) | [Docs](#) | [by unish](#)

Simulation

30:12:32 100%



object is near
Sending payload: {"distance":97.82,"object":"Near"}
Publish ok
Distancein cm97.82
object is near
Sending payload: {"distance":97.82,"object":"Near"}
Publish ok