

## ASSIGNMENT-4

Date	20 Oct 2022
Team ID	PNT2022TMID04713
Student Name & Roll number	SANDIYA S & 737819ECR154
Project Name	Smart Farmer-IOT Enabled Smart Farming Application
Maximum Marks	2 Marks

### Question1:

Write code and connections in work for ultrasonic sensors. Whenever distance is less than 100 cms 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 ORG "4kwojg"//IBM ORGANIZATION ID
10 #define DEVICE_TYPE "ultrasonic"//Device type mentioned in the Watson IOT Platform
11 #define DEVICE_ID "DGS148K1DETECI"//Device ID mentioned in the Watson IOT Platform
12 #define TOKEN "aawu7w3r9ag9d8kx"//token
13 String data;
14 float dist;
15
16
17 //----- Customise 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 FORMAT STRING
21 char authMethod[] = "use-token-auth"// authentication method
22 char token[] = TOKEN;
23 char clientId[] = "6:" 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 wifiCredential
29
30 int pin = A1;
31 int trig = 5;
32 int echo = 18;
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

```

```

esp32-blink.ino • diagram.json • libraries.txt • Library Manager
98
99 if (client.publish(publishTopic, (char*) payload_c_str)) {
100     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
101 } else {
102     Serial.println("Publish failed");
103 }
104
105 }
106
107 void mqttconnect() {
108     if (!client.connected()) {
109         Serial.print("Reconnecting client to ");
110         Serial.println(server);
111         while (!client.connect(clientId, authMethod, token)) {
112             Serial.print(".");
113             delay(500);
114         }
115
116         initManagedDevice();
117         Serial.println();
118     }
119 }
120
121 void wificonnect() //function definition for wifi connect
122 {
123     Serial.println();
124     Serial.print("Connecting to ");
125
126     WiFi.begin("Mokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
127     while (WiFi.status() != WL_CONNECTED) {
128         delay(500);
129         Serial.print(".");
130     }
131     Serial.println("");
132     Serial.println("WiFi connected");
133     Serial.println("IP address: ");
134     Serial.println(WiFi.localIP());

```

```

esp32-blink.ino • diagram.json • libraries.txt • Library Manager
123
124 WiFi.begin("Mokwi-GUEST", "", 6);//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
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=="Near")
156     // {
157     // Serial.println(data3);
158     // }

```

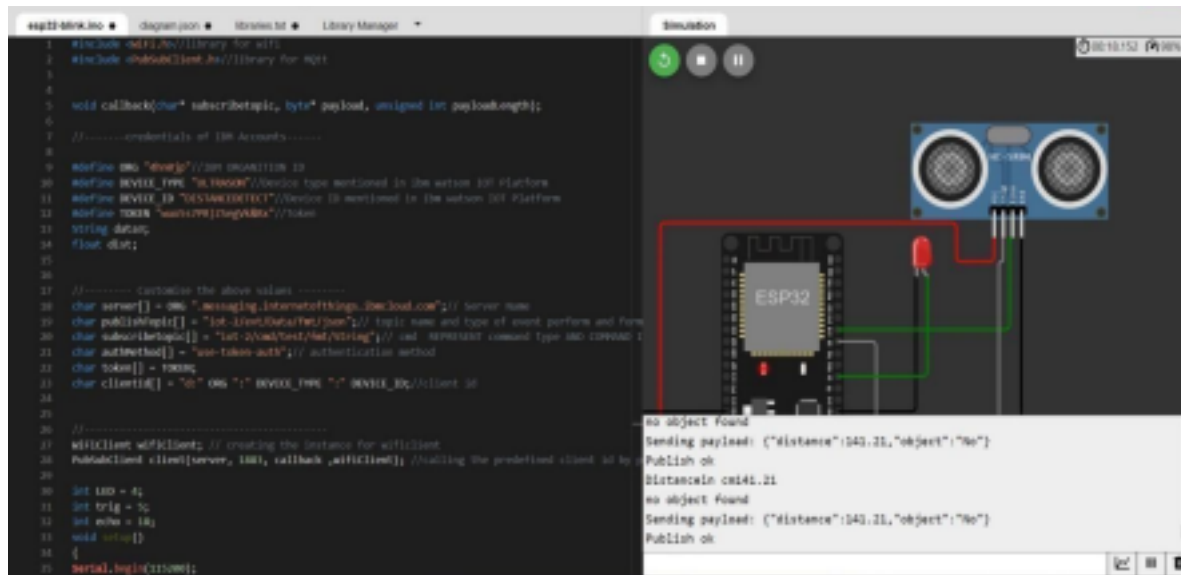
```
esp32-blink.ino • diagram.json • libraries.txt • Library Manager
142 }
143
144 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
145 {
146
147     Serial.print("callback invoked for topic: ");
148     Serial.println(subscribetopic);
149     for (int i = 0; i < payloadLength; 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 are tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A blue 'Add Device' button is in the top right. Below this, a header bar for a specific device shows 'DESIKARDE-11C2', 'Disconnected', 'ULTRASON', 'Device', and the timestamp 'Oct 20, 2022 9:46 AM'. A dropdown menu is open, showing 'Identity', 'Device Information', 'Recent Events' (which is selected), 'State', and 'Logs'. Below the menu, a message states: 'The recent events listed show the live stream of data that is coming and going from this device.' A table of recent events is displayed with columns: 'Event', 'Value', 'Format', and 'Last Received'. The table contains five rows of data, all with a value of '[{"distance":141.22,"object":"No"}]' and a format of 'json'. The 'Last Received' column shows 'a few seconds ago' for each entry. At the bottom, there is a pagination bar showing 'Items per page: 50' and '1 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":141.22,"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



Data sent to the IBMCloud Device when the object is near

Identity	Device Information	Recent Events	State	Logs
Event	Name	Format	Last Message	
Data	["distance":1043.21,"object":"No"]	json	2023-09-20 10:40:00	
Data	["distance":1043.21,"object":"No"]	json	2023-09-20 10:40:00	
Data	["distance":1043.21,"object":"No"]	json	2023-09-20 10:40:00	
Data	["distance":1043.21,"object":"No"]	json	2023-09-20 10:40:00	
Data	["distance":1043.21,"object":"No"]	json	2023-09-20 10:40:00	

When objects are near to the ultrasonic sensor

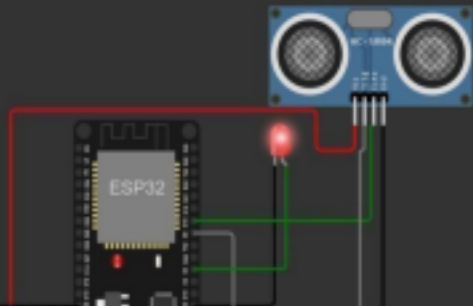
Wemos

esp32-arduino.ino  
by unish

Docs

Simulation

10:12:32 100%



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