

ASSIGNMENT- 4

DISTANCE DETECTION USING ULTRASONIC SENSOR

Date	20 October 2022
Team ID	PNT2022TMID01694
Name	DIVYASREETHA M
Student Roll Number	7376191EC137
Maximum Marks	2 Marks

Question1 :

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events.

WOKWI LINK :

<https://wokwi.com/projects/305566932847821378>

CODE :

```
1 #include <Arduino.h> //library for arduino
2 #include <Wire.h> //library for i2c
3
4
5 void callback(char* subscription, byte* payload, unsigned int payloadLength);
6
7 //-----credentials of IBM Account-----
8
9 #define IBM_HOST "mqtts://iot-000000000000.us-east-1.amazonaws.com" //Server Name
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 "mqtts://mqtts://mqtts://mqtts" //Token
13 #define DATA;
14 #define ALERT;
15
16
17 //-----Customize the above values-----
18
19 char server[] = IBM_HOST; //messaging.internetofthings.us-east-1.amazonaws.com // Server Name
20 char publishTopic[] = "iot-000000000000.us-east-1.amazonaws.com" // Topic Name and type of event perform and format in which data to be send
21 char subscribeTopic[] = "iot-000000000000.us-east-1.amazonaws.com" // Topic Name and type of event perform and format in which data to be send
22 char authMethod[] = "aws-iot-token" // authentication method
23 char token[] = "token";
24 char clientId[] = "id:" IBM_HOST "/" DEVICE_TYPE "/" DEVICE_ID; //client id
25
26
27 //-----
28 //MQTTClient initialized; //creating the instance for MQTTClient
29 MQTTClient client(server, token, callback, clientId); //creating the MQTTClient by passing parameter like server id, token and authentication
30
31 int led = 4;
32 int trig = 5;
33 int echo = 6;
34 void setup()
35 {
36   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 the 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 • chapters_3.ino • libraries • library manager

```

99
100     if (client.publish(topic, (char*) payload.c_str())) {
101         Serial.println("publish ok"); // if it successfully send data to the cloud then it will print publish ok in serial monitor or else it will print publish failed
102     } else {
103         Serial.println("publish failed");
104     }
105 }
106
107 void setup() {
108     if (!client.connected()) {
109         Serial.println("connecting client to ");
110         Serial.println(server);
111         while (!client.connect(clientID, authMethod, token)) {
112             Serial.println("");
113             delay(100);
114         }
115
116         initMessageDevice();
117         Serial.println();
118     }
119 }
120
121 void wifiConnect() //function definition for wifi connect
122 {
123     Serial.println();
124     Serial.println("connecting to ");
125
126     WiFi.begin("ssid-SSID", "", 0); //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());
135 }

```

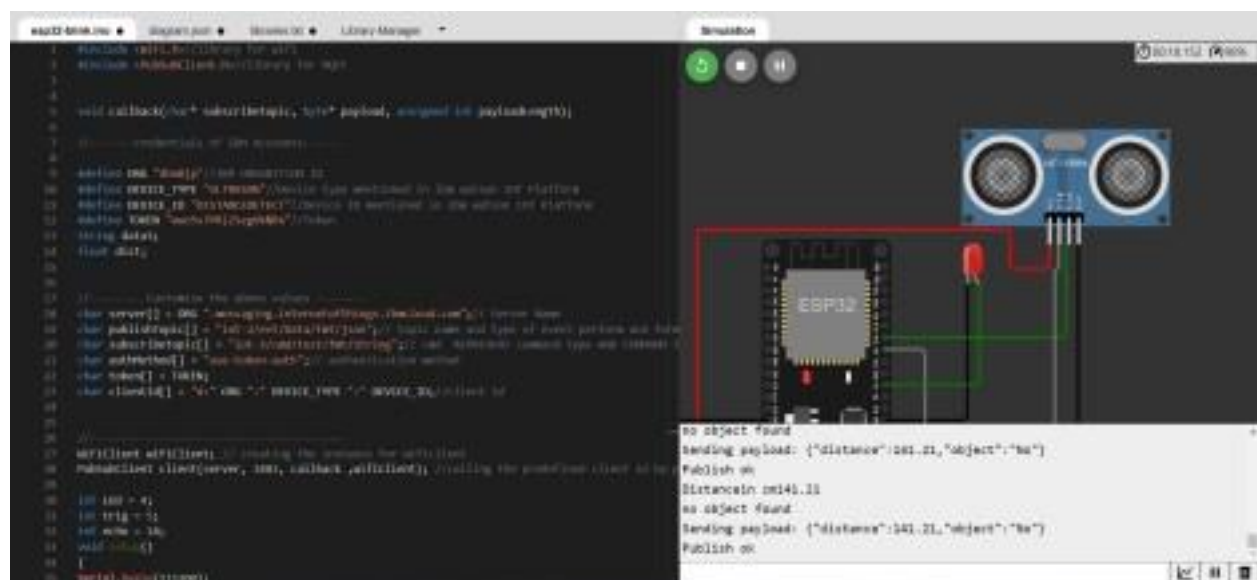
```
123
124   WiFi.begin("Wokwi-GUEST", "", 0); //passing the wifi credentials to establish the connection
125   while (WiFi.status() != WL_CONNECTED) {
126       delay(500);
127       Serial.print(".");
128   }
129   Serial.println("");
130   Serial.println("WiFi connected");
131   Serial.println("IP 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
162     // else
163     // {
164     // Serial.println(data3);
165     // digitalWrite(LED,LOW);
166     // }
167     data3="";
168 }
169
170
171

```

OUTPUT:



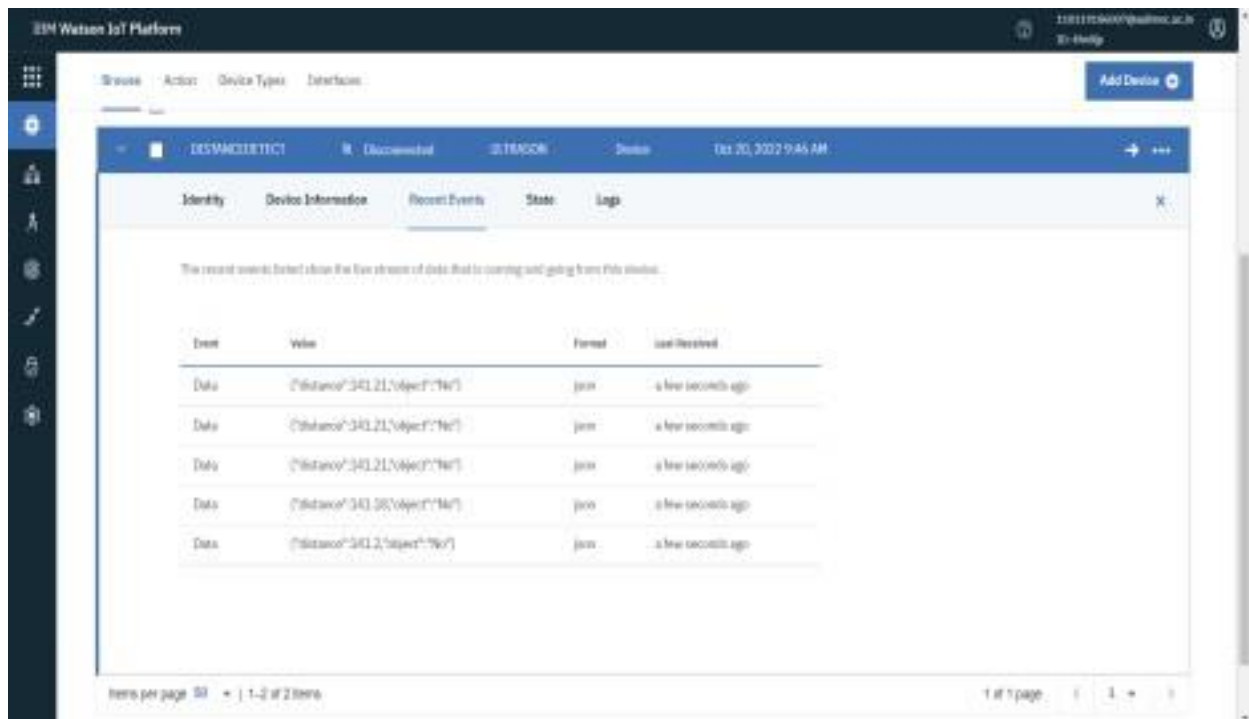
```

1 // Include the ESP32 library for WiFi
2 // Include the Arduino library for SPI
3
4 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
5 {
6     // Serial.println("callback invoked for topic: ");
7     // Serial.println(subscribetopic);
8     // for (int i = 0; i < payloadLength; i++) {
9         // Serial.print((char)payload[i]);
10        data3 += (char)payload[i];
11    }
12
13    // Serial.println("data: "+ data3);
14    // if(data3=="Near")
15    // {
16    // Serial.println(data3);
17    // digitalWrite(LED,HIGH);
18    // }
19
20
21    // else
22    // {
23    // Serial.println(data3);
24    // digitalWrite(LED,LOW);
25    // }
26    data3="";
27 }
28
29
30

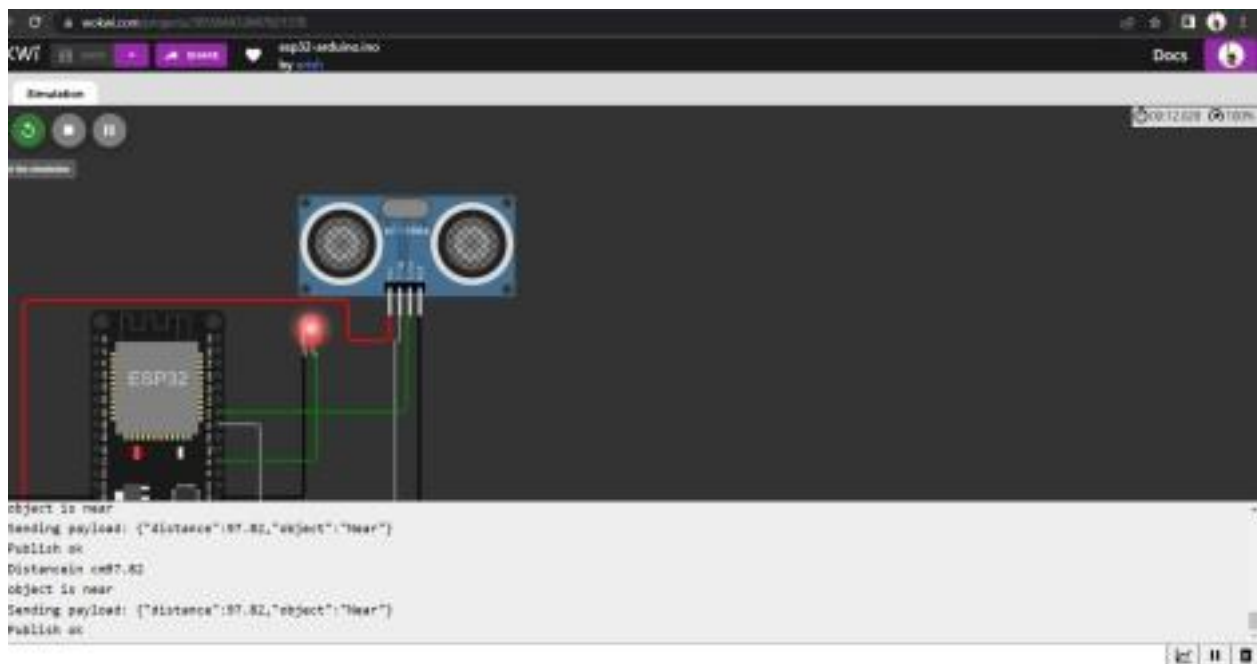
```

no object found
 Sending payload: {"distance":241.21,"object":"No"}
 Publish OK
 Distance in cm:241.21
 no object found
 Sending payload: {"distance":241.21,"object":"No"}
 Publish OK

Data send to the IBM cloud device when the object is far



when object is near to the ultrasonic sensor



Data sent to the IBM Cloud Device when the object is near

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Devices', 'Actions', 'Device Types', and 'Interactions'. The main content area is titled 'TESTING TEST' and shows a list of recent events for a device. The events are listed in a table with columns: Event, Value, Format, and Last Received. The events are all 'Data' type and are received 'a few seconds ago'.

Event	Value	Format	Last Received
Data	{"distance": 79.66, "object": "None"}	json	a few seconds ago
Data	{"distance": 79.64, "object": "None"}	json	a few seconds ago
Data	{"distance": 79.66, "object": "None"}	json	a few seconds ago
Data	{"distance": 79.64, "object": "None"}	json	a few seconds ago
Data	{"distance": 79.65, "object": "None"}	json	a few seconds ago

Items per page: 50 | 1-2 of 2 items | 1 of 1 page

<https://wokwi.com/projects/305566932847821378>