

Assignment-4

Distance Detection Using Ultrasonic Sensor

Date	26 October 2022
Team ID	PNT2022MID48179
Name	Sasika.K
Student Roll Number	912619106011
Maximum Marks	2 Marks

Question 1:

Write code and connections in wokwi for ultrasonic sensor.

Whenever distance is less than 100 centimeters it should 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  void callback(char* topic, byte* payload, unsigned int payloadLength);
4
5
6  #define ORG "4h@p" //IBM ORGANIZATION ID
7  #define DEVICE_TYPE "ULTRASON"
8  #define DEVICE_ID "DISTANCEDETECT"
9  #define TOKEN "uo0s7P8jZSegu&R"
10 String data;
11 float dist;
12 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
13 char publishTopic[] = "iot-2/evata/fwt/json";
14 char authMethod[] = "use-token-auth"; // authentication method
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
17
18 int LED = 4;
19 int trig = 5;
20 int echo = 18;
21 void setup()
22 {
23   Serial.begin(115200);
24 }
```

```

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 += ", \"object\": \"";
89     payload += object;
90     payload += "\"}";
91
92
93     Serial.print("Sending payload: ");
94     Serial.println(payload);
95
96
97
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 wificonnect
122 {
123     Serial.println();
124     Serial.print("Connecting to ");
125
126     WiFi.begin("Wokwi-GUEST", "", 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 }

```

```

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     // }
159 }

```

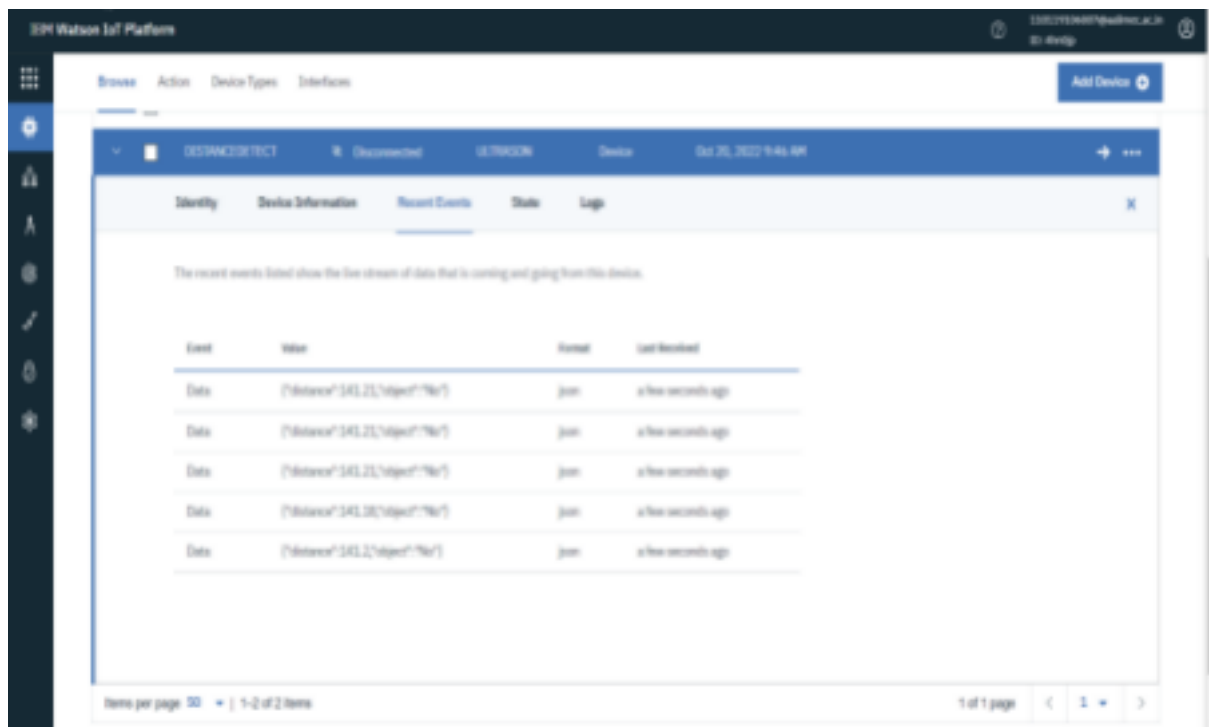
```

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 }

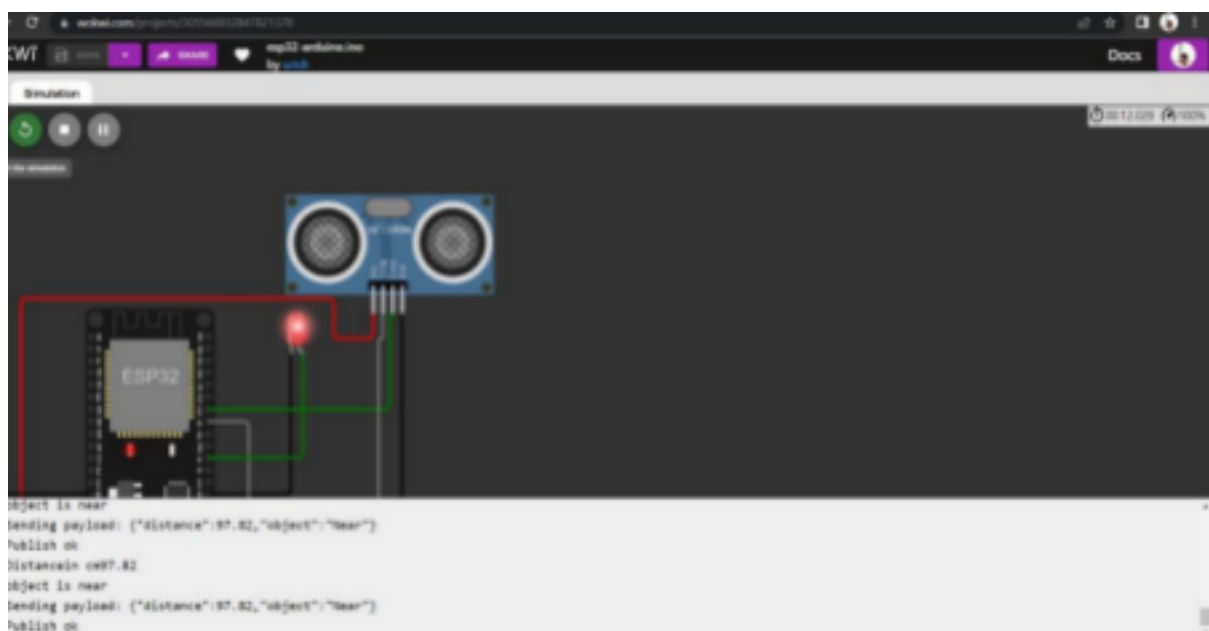
```



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



When object is near to the ultrasonic sensor



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

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area displays details for a device named 'SENSORCODETEST', which is 'Disconnected'. The 'Recent Events' tab is active, showing a table of events. The table has columns for 'Event', 'Value', 'Status', and 'Last Modified'. Five events are listed, all with a status of 'join' and a 'Last Modified' time of 'a few seconds ago'. The bottom of the page shows 'Items per page: 10' and '1 of 1 page'.

Event	Value	Status	Last Modified
Data	{"distance":79.66,"dpsd":"New"}	join	a few seconds ago
Data	{"distance":79.66,"dpsd":"New"}	join	a few seconds ago
Data	{"distance":79.66,"dpsd":"New"}	join	a few seconds ago
Data	{"distance":79.66,"dpsd":"New"}	join	a few seconds ago
Data	{"distance":79.66,"dpsd":"New"}	join	a few seconds ago