

ASSIGNMENT 4

DISTANCE DETECTION USING ULTRASONIC SENSOR

| | |
|---------------------|-----------------|
| Date | 20 October 2022 |
| Team ID | PNT2022TMID2481 |
| Name | SIVA POORANI M |
| Registration Number | 210419106105 |
| Maximum Marks | 4 Marks |

Question:

Write code and connections in wokwi for ultrasonic sensor. Whenever the distance is less than 100cm sent “alert” to IBM cloud and display in device recent.

WOKWI LINK:

<https://wokwi.com/projects/347469744763306579>

CODE:

```
WOKWI [SAVE] [SHARE] [ultraS] Docs S
sketch.ino diagram.json libraries.txt Library Manager
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include <Ultrasonic.h>
4
5 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
6
7 //-----credentials of IBM Accounts-----
8
9 #define ORG "4xmt8v" //IBM ORGANIZATION ID
10 #define DEVICE_TYPE "UltraS" //Device type mentioned in ibm watson IOT Platform
11 #define DEVICE_ID "Ultra1814" //Device ID mentioned in ibm watson IOT Platform
12 #define TOKEN "WztSaoT0q-3oWhk&u0" //Token
13 String data3;
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/fmt/json"; // topic name and type of event perform and format in which data to be send
20 char subscribetopic[] = "iot-2/cmd/test/fmt/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[] = "d:" 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 wifi credential
```

```
sketch.ino  diagram.json  libraries.txt  Library Manager  ▾
28  PublishClient(client(server, 1883, callback, wifiClient)); //calling the predefined client id by passing parameter like server id,port and wifi credential
29
30  int LED=4;
31  int trig=5;
32  int echo=18;
33  void setup()// configureing the ESP32 Z
34  {
35      Serial.begin(115200);
36
37      pinMode(trig,OUTPUT);
38      pinMode(echo,INPUT);
39      pinMode(LED,OUTPUT);
40      delay(10);
41      Serial.println();
42      wifiConnect();
43      mqttConnect();
44  }
45
46  void loop()// Recursive Function
47  {
48
49      digitalWrite(trig,LOW);
50      digitalWrite(trig,HIGH);
51      delayMicroseconds(10);
52      digitalWrite(trig,LOW);
53      float dur=pulseIn(echo,HIGH);
54      float dist= (dur*0.0343)/2;
55      Serial.print("Distance in centimeter:");
56      Serial.println(dist);
57
```

```
sketch.ino  diagram.json  libraries.txt  Library Manager  ▾
55      Serial.print("Distance in centimeter:");
56      Serial.println(dist);
57
58      PublishData(dist);
59      delay(1000);
60      if (!client.loop()) {
61          mqttConnect();
62      }
63  }
64
65
66
67  /*.....retrieving to Cloud.....*/
68
69  void PublishData(float dist) {
70      mqttConnect();//function call for connecting to ibm
71      /*
72      | | creating the String in in form JSon to update the data to ibm cloud
73      */
74
75      String object;
76      if(dist<100)
77      {
78          digitalWrite(LED,HIGH);
79          Serial.println("object is near");
80          object="Near";
81      }
82      else
83      {
84          digitalWrite(LED,LOW);
85      }
86  }
```

```
sketch.ino • diagram.json libraries.txt Library Manager
81 }
82 else
83 {
84     digitalWrite(LED, LOW);
85     Serial.println("no object found");
86     object="No";
87 }
88 String payload = "{\"distance\": ";
89 payload += dist;
90 payload += ", " "object\": \"";
91 payload += object;
92 payload += "\";";
93
94
95 Serial.print("Sending payload: ");
96 Serial.println(payload);
97
98
99 if (client.publish(publishTopic, (char*) payload.c_str())) {
100     Serial.println("Publish ok");// if it sucessfully 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 void mqttconnect() {
107     if (!client.connected()) {
108         Serial.print("Reconnecting client to ");
109         Serial.println(server);
```

```
sketch.ino • diagram.json libraries.txt Library Manager
110 while (!client.connect(clientId, authMethod, token)) {
111     Serial.print(".");
112     delay(500);
113 }
114
115     initManagedDevice();
116     Serial.println();
117 }
118
119 void wificonnect() //function defination for wificonnect
120 {
121     Serial.println();
122     Serial.print("Connecting to ");
123
124     WiFi.begin("Wokwi-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");
```

```
sketch.ino • diagram.json libraries.txt Library Manager ▼
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=="lighton")
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   // }
```

```
sketch.ino • diagram.json libraries.txt Library Manager ▼
159
160   // }
161
162   // else
163   // {
164   //Serial.println(data3);
165   //digitalWrite(LED,LOW);
166
167   // }
168   data3="";
169
170
171   }
172
```

OUTPUT:

sketch.ino

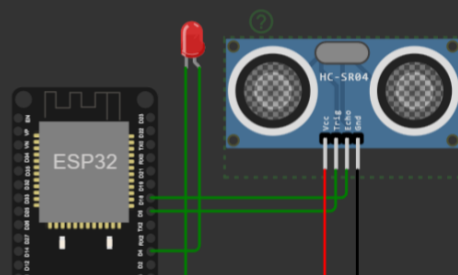
diagram.json

libraries.txt

Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include <Ultrasonic.h>
4
5 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
6
7 //-----credentials of IBM Accounts-----
8
9 #define ORG "4xmtBv" //IBM ORGANITION ID
10 #define DEVICE_TYPE "UltraS" //Device type mentioned in ibm watson IOT Platform
11 #define DEVICE_ID "ultra1814" //Device ID mentioned in ibm watson IOT Platform
12 #define TOKEN "WztSaoT0q-3oWhk8u0" //Token
13 String data3;
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/fmt/json"; // topic name and type of event
20 char subscribetopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command
21 char authMethod[] = "use-token-auth"; // authentication method
22 char token[] = TOKEN;
23 char clientId[] = "d:" 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 predefi
29
30 int LED=4;
```

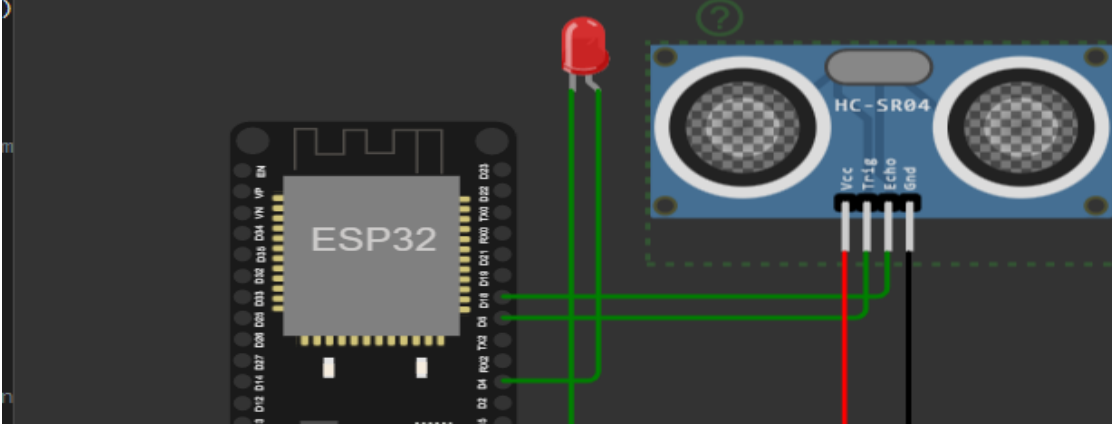
Simulation



object is near
Sending payload: {"distance":66.56,"object":"Near"}
Publish ok
Distance in centimeter:66.56
object is near
Sending payload: {"distance":66.56,"object":"Near"}
Publish ok

When the object is near or far from the ultrasonic sensor:

Simulation



object is near
Sending payload: {"distance":2.01,"object":"Near"}
Publish ok
Distance in centimeter:318.73
no object found
Sending payload: {"distance":318.73,"object":"No"}
Publish ok

Data send to IBM Cloud device:

IBM Watson IoT Platform

210419106105@smartintenz.com
ID: 4xmt8v

← Back

Device Drilldown - ultra1814

Connection Information

Recent Events

State

Device Information

Metadata

Diagnostics

Connection Logs

Device Actions

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

| Event | Value | Format | Last Received |
|-------|------------------------------------|--------|-------------------|
| Data | {"distance":66.56,"object":"Near"} | json | a few seconds ago |
| Data | {"distance":66.56,"object":"Near"} | json | a few seconds ago |
| Data | {"distance":218.89,"object":"No"} | json | a few seconds ago |
| Data | {"distance":123.05,"object":"No"} | json | a few seconds ago |
| Data | {"distance":230,"object":"No"} | json | a few seconds ago |

0 Simulations running

<https://wokwi.com/projects/347469744763306579>