

ASSIGNMENT-4

DISTANCE DETECTION USING ULTRASONIC SENSOR

Date	22 October 2022
Team ID	PNT2022TMID11029
Name	Muthu Kumaran G
Student Roll Number	811519106088
Maximum Marks	2 Marks

Question1:

Write code and connections in work 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 <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4
5 void callback(char* subscibetopic, byte* payload, unsigned int payloadLength);
6
7 //-----credentials of IBM Accounts-----
8
9 #define ORG "4hn@jp" //IBM ORGANITION ID
10#define DEVICE_TYPE "ULTRASON" //Device type mentioned in ibm watson IOT Platform
11#define DEVICE_ID "DISTANCEDETECT" //Device ID mentioned in ibm watson IOT Platform
12#define TOKEN "wuo5s7PRzSegvk&Rx" //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
29
30 int LED = 4;
31 int trig = 5;
32 int echo = 18;
33 void setup()
34 {
35 Serial.begin(115200);
```



Edit with WPS Office

```
esp32-blink.ino ● diagram.json ● libraries.txt ● Library Manager ▾
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
```



Edit with WPS Office

```

55
56     /* creating the String in in form JSON to update the data to ibm cloud
57 */
58     String object;
59     if (dist <100)
60     {
61         digitalWrite(LED,HIGH);
62         Serial.println("object is near");
63         object = "Near";
64     }
65     else
66     {
67         digitalWrite(LED,LOW);
68         Serial.println("no object found");
69         object = "No";
70     }
71
72     String payload = "{\"distance\":";
73     payload += dist;
74     payload += "," "\"object\":\"";
75     payload += object;
76     payload += "}";
77
78     Serial.print("sending payload: ");
79     Serial.println(payload);
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98

```

```

esp32-blink.ino • diagram.json • libraries.txt • Library Manager ▾
98
99     if (client.publish(publisTopic, (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);
110         while (!client.connect(clientId, authMethod, token)) {
111             Serial.print(".");
112             delay(500);
113         }
114         initManagedDevice();
115         Serial.println();
116     }
117 }
118 void wificonnect() //function defination for wificonnect
119 {
120     Serial.println();
121     Serial.print("Connecting to ");
122
123     WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
124     while (WiFi.status() != WL_CONNECTED) {
125         delay(500);
126         Serial.print(".");
127     }
128     Serial.println("");
129     Serial.println("WiFi connected");
130     Serial.print("IP address: ");
131     Serial.println(WiFi.localIP());
132

```



Edit with WPS Office

```
esp32-blink.ino • diagram.json • libraries.txt • Library Manager ▾
123 WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
124 while (WiFi.status() != WL_CONNECTED) {
125     delay(500);
126     Serial.print(".");
127 }
128 Serial.println("");
129 Serial.println("WiFi connected");
130 Serial.println("IP address: ");
131 Serial.println(WiFi.localIP());
132 }
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 //        // do something with the data
159 //    }
160 }
```



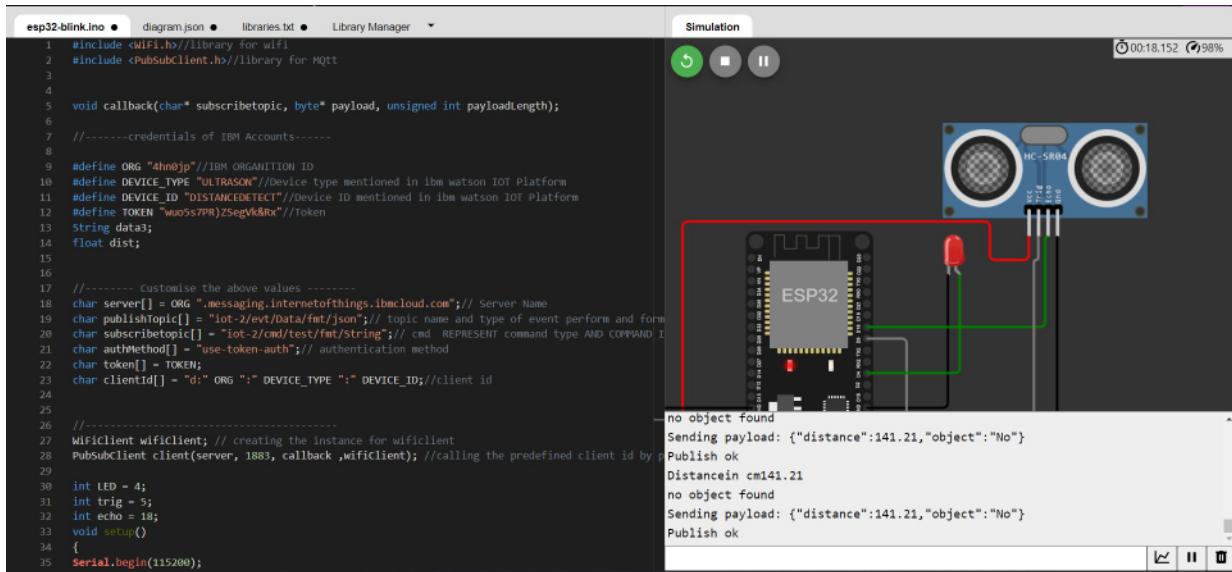
Edit with WPS Office

```
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     //  
167     data3="";  
168  
169  
170  
171 }
```



Edit with WPS Office

OUTPUT:



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

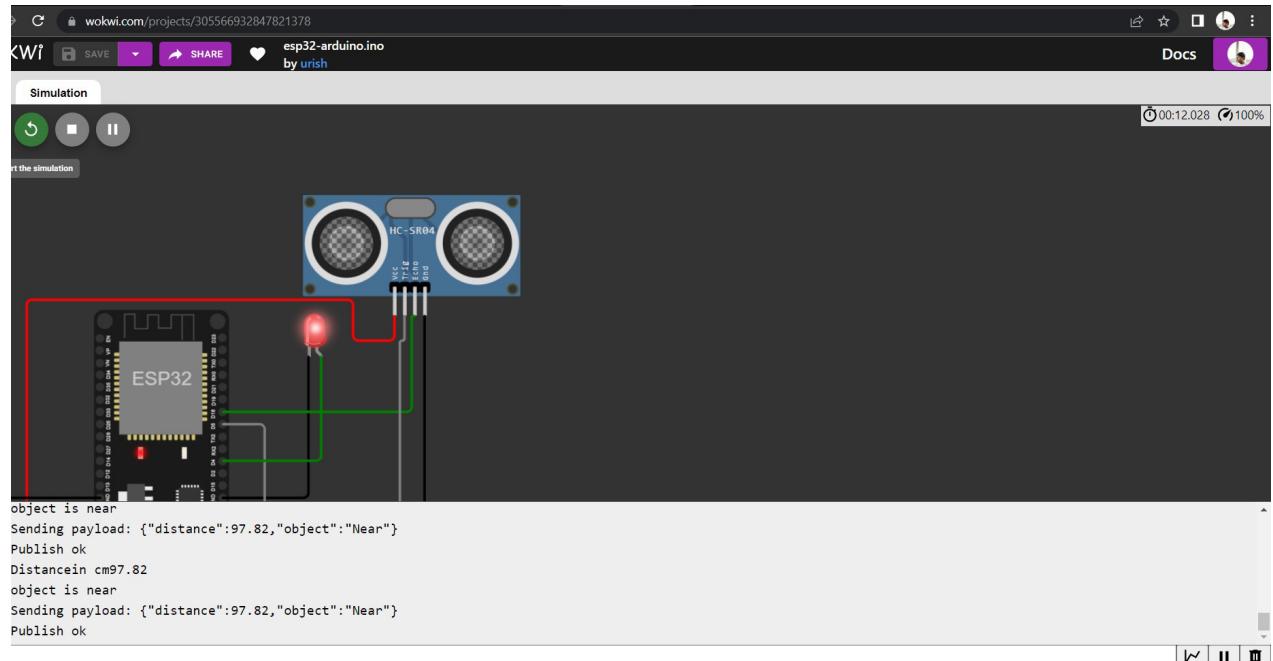
The screenshot shows the IBM Watson IoT Platform Device Events page. It displays a list of recent events for a device named DISTANCEDETECT. The events are listed in a table with columns: Event, Value, Format, and Last Received. All events show a distance of 141.21 and an object status of "No".

Event	Value	Format	Last Received
Data	{"distance":141.21,"object":"No"}	json	a few seconds ago
Data	{"distance":141.21,"object":"No"}	json	a few seconds ago
Data	{"distance":141.21,"object":"No"}	json	a few seconds ago
Data	{"distance":141.18,"object":"No"}	json	a few seconds ago
Data	{"distance":141.2,"object":"No"}	json	a few seconds ago



Edit with WPS Office

when object is near to the ultrasonic sensor



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

Event	Value	Format	Last Received
Data	{'distance':79.66,'object':'Near'}	json	a few seconds ago
Data	{'distance':79.64,'object':'Near'}	json	a few seconds ago
Data	{'distance':79.66,'object':'Near'}	json	a few seconds ago
Data	{'distance':79.64,'object':'Near'}	json	a few seconds ago
Data	{'distance':79.66,'object':'Near'}	json	a few seconds ago

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



Edit with WPS Office