

## 4

## SENSOR

Date	20 October 2022
Clg name	Sona College of Technology
Name	Krish S
Student Roll Number	1919106041
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 <stdio.h> //for printf
2 #include <stdlib.h> //for library for exit
3
4
5 void callback(char* subscriptionTopic, byte* payload, unsigned int payloadLength);
6
7 //----- Credentials of IoT Accounts -----
8
9 #define URL "http://iot.digicloud.com" //IoT Account URL ID
10 #define DEVICE_TYPE "ESP8266" //Device Type mentioned in the Watson IoT Platform
11 #define DEVICE_ID "DST2008FRTCT" //Device ID mentioned in the Watson IoT Platform
12 #define TOKEN "aawc7PqZkg6kksr" //Token
13 String dataIn;
14 float dist;
15
16
17 //----- Function to store values -----
18 char server[] = URL; //messaging.internetofthings.ibmcloud.com// Server Name
19 char publishTopic[] = "iot-2/rest/api/v2/pmc" // topic name and type of event perform and format to which data to be send
20 char subscribeTopic[] = "iot-2/mqttact/get/string" // iot devices command type and connect to test or control action
21 char authHeader[] = "iot-token-auth" // authentication method
22 char token[] = TOKEN;
23 char clientId[] = "dt:" URL "/" DEVICE_TYPE "/" DEVICE_ID;//client id
24
25
26 //
27
28 #if !defined(WIFI_CLIENT) // creating the instance for wifi client
29 #include <WiFiClient.h> //calling the predefine client id by passing parameter like server id,portand differentiation
30
31 int ssp = 0;
32 int trig = 5;
33 int echo = 10;
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 += ",";
89     payload += "\"object\":";
90     payload += object;
91     payload += "\"}";
92
93     Serial.print("Sending payload: ");
94     Serial.println(payload);
95
96
97
98

```

```

emp32@blink:~$ cat chapter_20.ino
100
101 if (client.publish(topic, (char*) payload.c_str())) {
102     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
103 } else {
104     Serial.println("publish failed");
105 }
106
107 }
108
109 void mqttConnect() {
110     if (!client.connected()) {
111         Serial.println("reconnecting client to ");
112         Serial.println(server);
113         while (!client.connect(topic, username, password)) {
114             Serial.print(".");
115             delay(100);
116         }
117
118         lastMsgRecvTime();
119         Serial.println();
120     }
121 }
122
123 void mqttConnect() //function definition for mqttConnect
124 {
125     Serial.println();
126     Serial.print("connecting to ");
127
128     WiFi.begin("ssid", "password"); //passing the wifi credentials to establish the connection
129     while (WiFi.status() != WL_CONNECTED) {
130         delay(500);
131         Serial.print(".");
132     }
133     Serial.println("");
134     Serial.println("WiFi connected");
135     Serial.println("IP address: ");
136     Serial.println(WiFi.localIP());
137 }

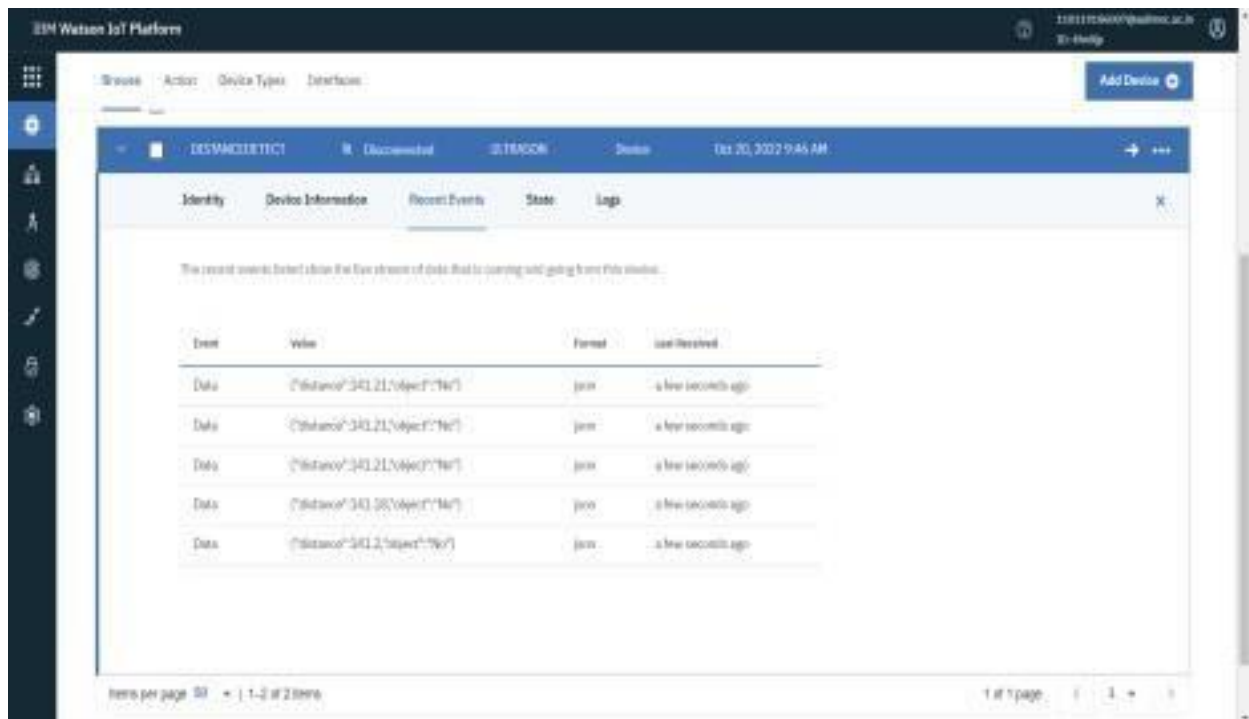
```

```
123
124   WiFi.begin("Wokwi-GUEST", ""); //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 }
```

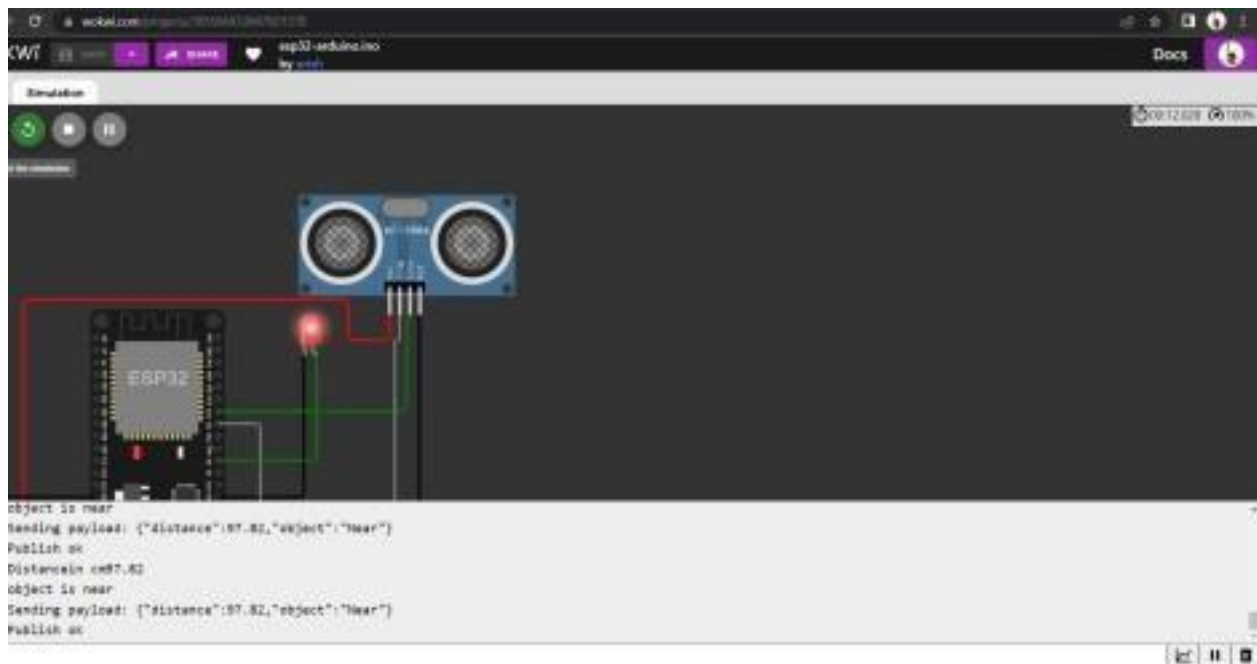
```

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

```



when object is near to the ultrasonic sensor



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

