

4

SENSOR

Date	20 October 2022
Team ID	PNT2022TMID32971
Name	Srika.s
Student Roll Number	820419106056
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.

CODE :

[illegible]

```
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

```

```

esp32-01rk-rc • setup.py • browser • 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 so also it will print publish failed
102 } else {
103     Serial.println("publish failed");
104 }
105 }
106
107 void setup() {
108     if (client.connected()) {
109         Serial.print("Reconnecting client to ");
110         Serial.println(server);
111         while (!client.connect(topic, authmethod, token)) {
112             Serial.print("-");
113             delay(500);
114         }
115         IoTManagerDevice();
116         Serial.println();
117     }
118 }
119
120 void setup_wifi() //function definition for setup_wifi
121 {
122     Serial.println();
123     Serial.println("connecting to ");
124
125     WiFi.begin("wifid-01rk-rc", ""); //passing the wifi credentials to establish the connection
126     while (WiFi.status() != WL_CONNECTED) {
127         delay(500);
128         Serial.print(".");
129     }
130     Serial.println("");
131     Serial.println("WiFi connected");
132     Serial.println("IP address: ");
133     Serial.println(WiFi.localIP());

```

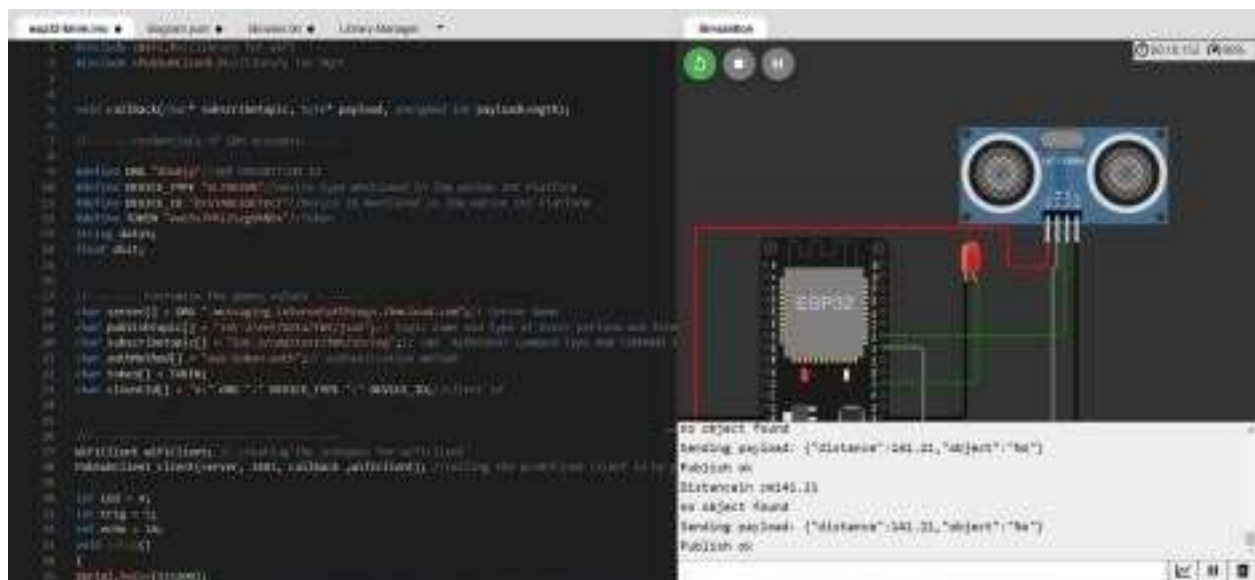
```
123
124   WiFi.begin("Wokwi-GUEST", "", 0); //passing the wifi credentials to establish the connection
125   while (WiFi.status() != WL_CONNECTED) {
126       delay(500);
127       Serial.print(".");
128   }
129   Serial.println("");
130   Serial.println("WiFi connected");
131   Serial.println("IP 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     // }
159 }
```

```

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

```

OUTPUT:



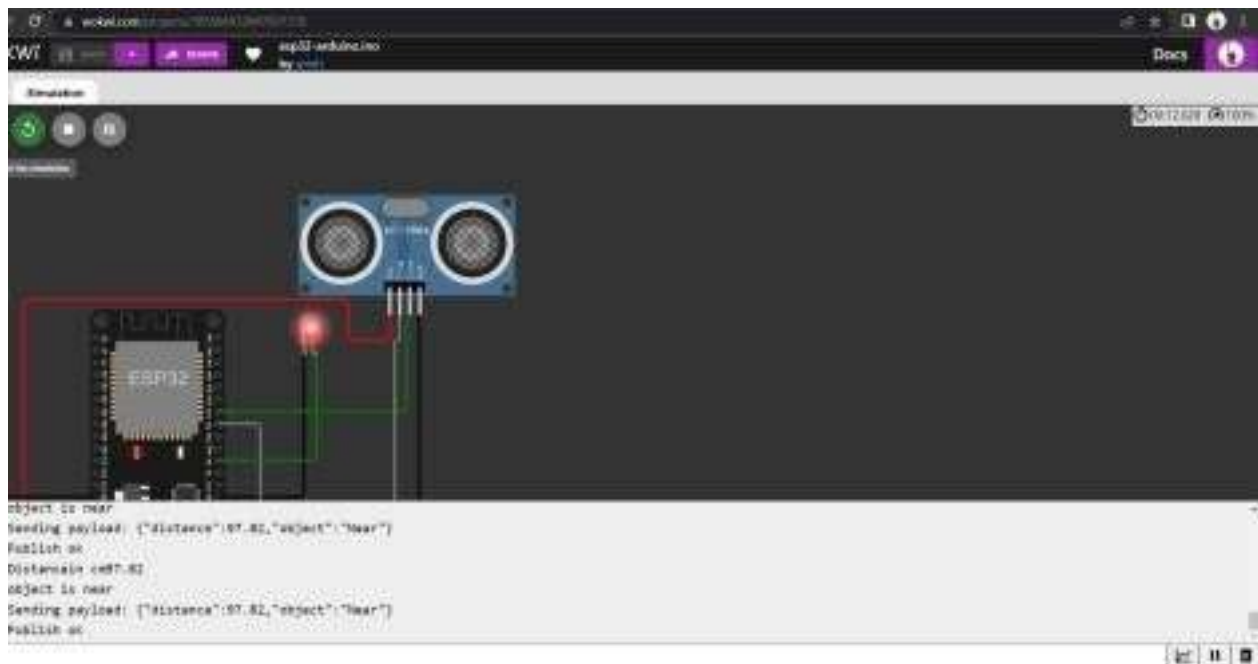
Data send to the IBM cloud device when the object is far

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Devices', 'Alerts', 'Device Types', and 'Integrations'. A sidebar on the left contains various icons for navigation. The main content area displays a table of recent events for a device named 'ultrasonic'.

Event	Value	Format	Last Received
Data	["distance":341.21,"object":"near"]	json	a few seconds ago
Data	["distance":341.21,"object":"near"]	json	a few seconds ago
Data	["distance":341.21,"object":"near"]	json	a few seconds ago
Data	["distance":341.21,"object":"near"]	json	a few seconds ago
Data	["distance":341.21,"object":"near"]	json	a few seconds ago

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

When object is near to the ultrasonic sensor



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

