

ASSIGNMENT 4

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

code:

```
#include <WiFi.h>
#include <PubSubClient.h>
#define ECHO_PIN 2
#define TRIG_PIN 4
#define LED 5
//IBM credentials
#define ORG "ld4vg3"
#define DEVICE_TYPE "TestDevice"
#define DEVICE_ID "102030"
#define TOKEN "4+*C1BQ7bY7+b3E2vf"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/loTSensor/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883,wifiClient);
void setup(){
  Serial.begin(115200);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}
float readDistanceCM() {
```

```

digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
int duration = pulseIn(ECHO_PIN, HIGH);
return duration * 0.034 / 2;
}
void loop(){
float distance = readDistanceCM();
bool isNearby = distance < 100;
digitalWrite(LED, isNearby);
Serial.print("Distance: ");
Serial.println(distance);
delay(100);
if (isNearby == 1){
PublishData(distance);
}
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
void PublishData(float distance) {
mqttconnect();
String payload = "{\"Alert\":\"\"";
payload += distance;
payload += " is less than 100cms\"";
payload += "\"}";
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish ok");
}
else {
Serial.println("Publish failed");
}
}
void mqttconnect() {
if (!client.connected()) {

```

```

Serial.print("Reconnecting client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
  Serial.print(".");
  delay(500);
} initManagedDevice();
Serial.println();
}
}

void wificonnect(){
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}
}

```

Output:

The screenshot displays the Wokwi IoT simulator interface. On the left, the Arduino IDE environment shows the following code for an ESP32:

```
11 //Customise the above values
12 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
13 char publishTopic[] = "iot-2/type/1014/id/123/evt/IoTSensor/fmt/json";
14 char subscribetopic[] = "iot-2/cmd/test/fmt/String";
15 char authMethod[] = "use-token-auth";
16 char token[] = TOKEN;
17 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
18 //---
19 WiFiClient wificlient;
20 PubSubClient client(server, 1883,wificlient);
21 void setup()
22 {
23   Serial.begin(115200);
24   pinMode(TRIG_PIN, OUTPUT);
25   pinMode(ECHO_PIN, INPUT);
26   pinMode(LED, OUTPUT);
27   delay(10);
28   Serial.println();
29   wificlient.connect();
30   mqtt.connect();
31 }
32 float readDistanceCM() {
33   digitalWrite(TRIG_PIN, LOW);
34   delayMicroseconds(2);
35   digitalWrite(TRIG_PIN, HIGH);
36   delayMicroseconds(10);
37   digitalWrite(TRIG_PIN, LOW);
38   int duration = pulseIn(ECHO_PIN, HIGH);
39   return duration * 0.034 / 2;
40 }
41 void loop()
42 {
43   float distance = readDistanceCM();
44   bool isNearby = distance < 100;
45   digitalWrite(LED, isNearby);
46   PubSubClient client(server, 1883, wificlient);
```

The simulation window on the right shows the physical connection between the ESP32 and the HC-SR04 sensor. Below the simulation, the output log displays the following distance readings:

```
Distance: 399.92
Distance: 399.92
Distance: 399.92
Distance: 399.96
Distance: 399.92
Distance: 399.92
Distance: 399.92
Distance: 399.94
```

IBM IOT WATSON OUTPUT:

The screenshot shows the IBM Watson IoT Platform dashboard. The device 'TestDevice_1' is listed as 'Connected' with a last update time of '11 Nov 2022 14:43'. The 'Recent Events' tab is active, showing a live stream of data from the device. The events are as follows:

Event	Value	Format	Last Received
event_1	{"distance":58,"cmd":"alert"}	json	a few seconds ago
event_1	{"distance":37,"cmd":"alert"}	json	a few seconds ago
event_1	{"distance":94,"cmd":"alert"}	json	a few seconds ago
event_1	{"distance":66,"cmd":"alert"}	json	a few seconds ago
event_1	{"distance":96,"cmd":"alert"}	json	a few seconds ago

At the bottom of the dashboard, it indicates '2 Simulations running'.