

Name:KR.Vashundhara

Reg N0:910619104098

Assignment:4

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. Upload document with wokwi share link and image of ibm cloud

Code:

```
#include "WiFi.h"

#include "WiFiClient.h"

#define DEVICE_TYPE "ESP32_controller"

#define DEVICE_ID "Assignment_4"

#define TOKEN "&S?LQdLRzh3n2gkyDt"

char server[] = "ORG".meassaging.internertofthings.ibmcloud.com";

char pubTopic1[] = "iot-2/evt/status1/fmt/json";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

long now;

WiFiClient wificlient;

pubsubclient client(server, 1883, NULL, wificlient);

#define echoPin 2 // attach pin D2 Arduino to pin Echo of HC-SR04

#define trigPin 3 //attach pin D3 Arduino to pin Trig of HC-SR04

// defines variables

long duration; // variable for the duration of sound wave travel
```

```
int distance; // variable for the distance measurement

void setup() {

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Serial Monitor
  Serial.println("with Arduino UNO R3");
}

void loop() {

  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)

  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);
}
```

```
if(distance<=100)
{
    string payload="{\"d\":{\"Name\":\"\" DEVICE_ID\"\"";
    payload += "\",\"Distance\":";
    payload += distance;
    payload += "}}";
    Serial.print("sending payload: ")
    Serial.println(payload);
    if (client.publish(pubTopic1, (char*) payload.c_str()))
    {
        Serial.print("***ALERT***");
    }

}

}
```

WOKWI

SAVE SHARE

Docs SIGN IN

sketch.ino diagram.json wifi.h Library Manager

```

1 // ----- //
2 // Arduino Ultrasonic Sensor HC-SR04
3 // Re-written by Arbi Abdul Jabbaar
4 // Using Arduino IDE 1.8.7
5 // Using HC-SR04 Module
6 // Tested on 17 September 2019
7 // ----- //
8 #include "WiFi.h"
9 #include "WiFiClient.h"
10 #define DEVICE_TYPE "ESP32_controller"
11 #define DEVICE_ID "Assignment_4"
12 #define TOKEN "&S?LQdLRzh3n2gkyDt"
13 char server[] = "org.messaging.internetofthings.ibmcloud.com";
14 char pubTopic1[] = "iot-2/evt/status1/fmt/json";
15 char authMethod[] = "use-token-auth";
16 char token[] = TOKEN;
17 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
18 long now;
19 WiFiClient wificlient;
20 PubSubClient client(server, 1883, NULL, wificlient);
21
22 #define echoPin 2 // attach pin D2 Arduino to pin Echo of HC-SR04
23 #define trigPin 3 // attach pin D3 Arduino to pin Trig of HC-SR04
24
25 // defines variables
26 long duration; // variable for the duration of sound wave travel
27 int distance; // variable for the distance measurement
28

```

Simulation

28°C Mostly cloudy

20:09 27-10-2022

WOKWI

SAVE SHARE

Docs SIGN IN

sketch.ino diagram.json wifi.h Library Manager

```

41 digitalWrite(trigPin, HIGH);
42 delayMicroseconds(10);
43 digitalWrite(trigPin, LOW);
44 // Reads the echoPin, returns the sound wave travel time in microseconds
45 duration = pulseIn(echoPin, HIGH);
46 // Calculating the distance
47 distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and
48 // Displays the distance on the Serial Monitor
49 Serial.print("Distance: ");
50 Serial.print(distance);
51 Serial.println(" cm");
52 delay(500);
53 if (distance <= 100)
54 {
55   string payload = "{\"d\":\"{" Name\":\"" DEVICE_ID "\"";
56   payload += ", \"Distance\":\"";
57   payload += distance;
58   payload += "\"}";
59   Serial.print("sending payload: ")
60   Serial.println(payload);
61   if (client.publish(pubTopic1, (char*) payload.c_str()))
62   {
63     Serial.print("***ALERT***");
64   }
65 }
66 }
67
68

```

Simulation

28°C Mostly cloudy

20:11 27-10-2022

OUTPUT:

Distance: 305 cm

Distance: 40 cm

ALERTDistance: 40 cm