

ASSIGNMENT 4

Ultrasonic sensor simulation in Wokwi

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

Code:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#define ECHO_PIN 2
#define TRIG_PIN 4
#define LED 5

//-----credentials of IBM Accounts-----

#define ORG "lvqpbw" //IBM ORGANIZATION ID
#define DEVICE_TYPE "geetha2610" //Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "26102001" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "&4pqjz27LOxdseI**7D" //Token

//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command
type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth"; // authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id

//-----
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, wifiClient); //calling the predefined client id
by passing parameter like server id, port and wificredential
void setup() // configuring the ESP32
{
```

```

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()// Recursive Function
{
    float distance = readDistanceCM();
    bool isNearby = distance < 100;
    digitalWrite(LED, isNearby);
    Serial.print("Measured distance: ");
    Serial.println(distance);
    delay(100);
    if (isNearby == 1){
        PublishData(distance);
    }
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

/*.....retrieving to
Cloud. .... */

void PublishData(float distance) {
    mqttconnect();//function call for connecting to ibm

```

```

/*
    creating the String in in form JSon to update the data to ibm cloud
*/
String payload = "{\"Alert\":\"\"";
payload += distance;
payload += " is less than 100cms\"";
payload += "}";

```

Diagram.json

```

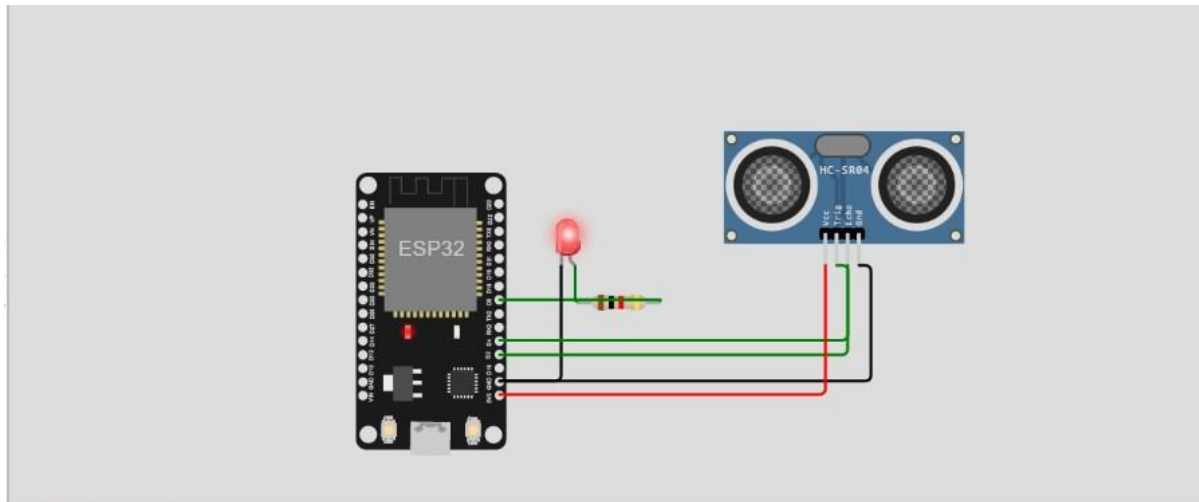
{
  "version": 1,
  "author": "abinaya",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -41.73, "left": -108.18, "attrs": {} },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": -16.04,
      "left": 21.83,
      "attrs": { "color": "red" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 41.63,
      "left": 48.17,
      "attrs": { "value": "1000" }
    },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -69.2, "left": 151.85, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [ ] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [ ] ],
    [ "led1:A", "r1:1", "green", [ "v0" ] ],
    [ "r1:2", "esp:D5", "green", [ "v0" ] ],
    [ "led1:C", "esp:GND.1", "black", [ "v0" ] ],
    [ "esp:D4", "ultrasonic1:TRIG", "green", [ "h246.49", "v-79.83" ] ],
    [ "esp:D2", "ultrasonic1:ECHO", "green", [ "h0" ] ],
    [ "esp:GND.1", "ultrasonic1:GND", "black", [ "h262.72", "v-104.77" ] ],
    [ "ultrasonic1:VCC", "esp:3V3", "red", [ "v0" ] ]
  ]
}

```

}

Wokwi link:

[sketch.ino copy - Wokwi Arduino and ESP32 Simulator](#)



```
Publish ok
Measured distance: 71.94
Sending payload: {"Alert":"71.94 is less than 100cms"}
Publish ok
Measured distance: 71.94
Sending payload: {"Alert":"71.94 is less than 100cms"}
Publish ok
```

Event	Value	Format	Last Received
Data	{"Alert":"71.94 is less than 100cms"}	json	a few seconds ago
Data	{"Alert":"71.98 is less than 100cms"}	json	a few seconds ago
Data	{"Alert":"72.03 is less than 100cms"}	json	a few seconds ago
Data	{"Alert":"71.94 is less than 100cms"}	json	a few seconds ago
Data	{"Alert":"71.94 is less than 100cms"}	json	a few seconds ago