

Assignment -4

Student Name	Harikrishnan S
Student Roll Number	720319106010

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. Upload document with wokwi share link and images of IBM cloud

Link: <https://wokwi.com/projects/new/esp32>

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;

#define trigpin    18
#define echopin    5

String data3;

#define ORG "nn9be5"//IBM ORGANITION ID
#define DEVICE_TYPE"harikrishnan"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "Hari123"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "harikrishnan123@"

#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();

String command;
String data="";
long duration;
float dist;

void setup()
```

```

{
    Serial.begin(115200);
    wifiConnect();
    pinMode(trigpin, OUTPUT);
    pinMode(echopin, INPUT);

    mqttConnect();
}

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

void loop()
{

    int pulseWidth = 0;
    digitalWrite(trigpin, HIGH);
    delayMicroseconds(100);
    digitalWrite(trigpin, LOW);
    pulseWidth = pulseIn(echopin, HIGH);
    Serial.print("AlertDistance: ");
    Serial.println(pulseWidth/58);

    publishData();
    if (!client.loop()) {
        mqttConnect();
    }
}

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

```

```

    }
}

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

void publishData()
{
    digitalWrite(trigpin,LOW);
    digitalWrite(trigpin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin,LOW);
    duration=pulseIn(echopin,HIGH);
    dist=duration*speed/2;
    if(dist<100){
        String payload = "{\"Normal Distance\":\"";
        payload += dist;
        payload += "\"}";

        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);
        if (client.publish(publishTopic, (char*) payload.c_str())) {
            Serial.println("Publish OK");
        }

    }

    if(dist>101 && dist<111){
        String payload = "{\"Alert distance\":\"";
        payload += dist;
        payload += "\"}";

        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);
        if(client.publish(publishTopic, (char*) payload.c_str())) {
            Serial.println("Warning crosses 110cm -- it automatically of the loop");
        }
    }
    else {
        Serial.println("Publish FAILED");
    }
}

```

```

    }

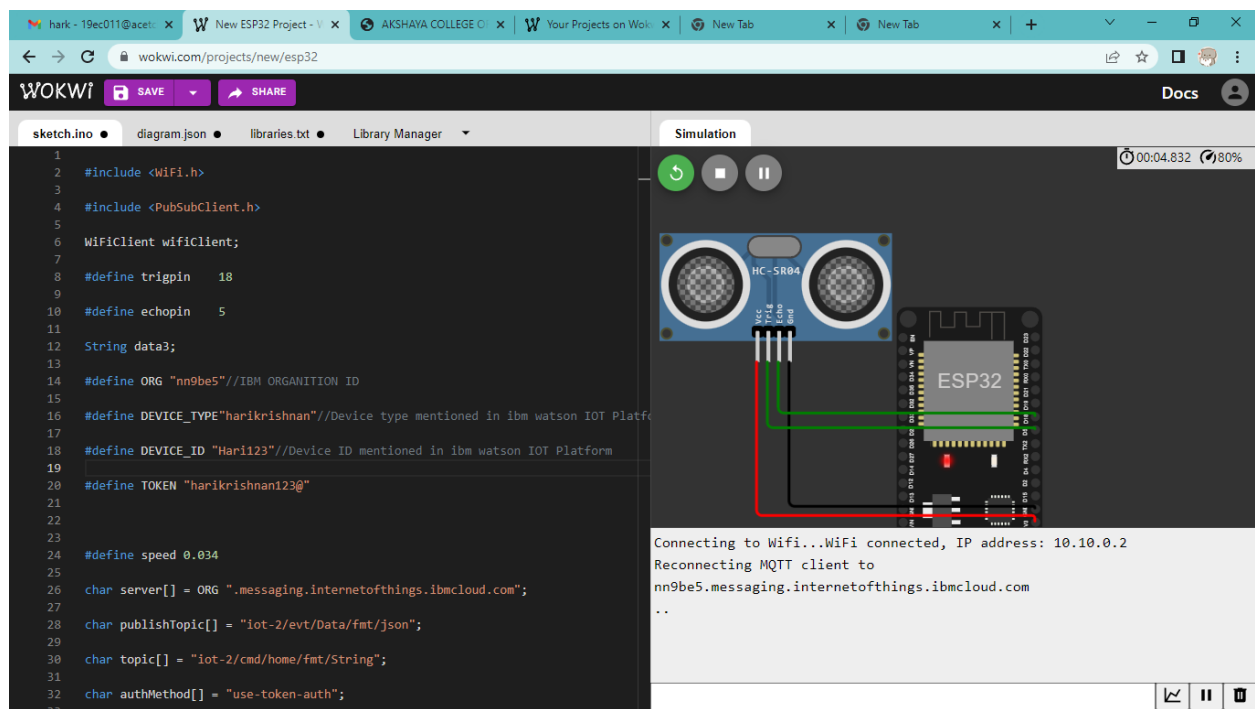
}

}

void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength){
  Serial.print("callback invoked for topic:");
  Serial.println(subscribeTopic);
  for(int i=0; i<payloadLength; i++){
    dist += (char)payload[i];
  }
  Serial.println("data:" + data3);
  if(data3=="lighton"){
    Serial.println(data3);

  }
  data3="";
}

```



Output

Link: <https://v9y7gx.internetofthings.ibmcloud.com/dashboard/devices/browse>

The screenshot displays the IBM Watson IoT Platform dashboard for a specific device. The interface includes a top navigation bar with the platform name and user information, a left sidebar with navigation icons, and a main content area. The main content area shows a table of recent events for the device, with columns for Event, Value, Format, and Last Received. The table lists five data points, all with the value {"Normal Distance":58.99} or {"Normal Distance":58.96} and a format of json. The last received time for all events is "a few seconds ago".

IBM Watson IoT Platform

19ec014@acetbce.edu.in
ID: v9y7gx

Browse Action Device Types Interfaces

Add Device

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"Normal Distance":58.99}	json	a few seconds ago
Data	{"Normal Distance":58.99}	json	a few seconds ago
Data	{"Normal Distance":58.99}	json	a few seconds ago
Data	{"Normal Distance":58.96}	json	a few seconds ago
Data	{"Normal Distance":58.96}	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

1 of 1 page