

Assignment -4

Student Name	Haripriya U
Student Roll no	720319106012

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/348548872665039443>

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

#define trigpin    18
#define echopin    5

String data3;

#define ORG "kabw0z"//IBM ORGANITION ID
#define DEVICE_TYPE"smartfarming"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "sf1234sf"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "haripriya23"

#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="";
}
}

```

WOKWI

sketch.ino diagram.json libraries.txt Library Manager

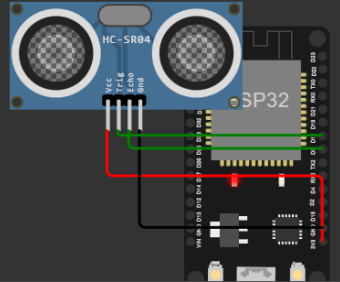
```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength){
4   #define ORG "kabw0z"
5   #define DEVICE_TYPE "smartfarming"
6   #define DEVICE_ID "sf1234sf"
7   #define TOKEN "haripriya23"
8   String data3;
9   char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
10  char publishTopic[] = "iot-2/evt/Data/fmt/json";
11  char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
12  char authMethod[] = "use-token-auth";
13  char token[] = TOKEN;
14  char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
15  WiFiClient wifiClient;
16  PubSubClient client(server, 1883, callback, wifiClient);
17  const int trigPin = 5;
18  const int echoPin = 18;
19  #define SOUND_SPEED 0.034
20  long duration;
21  float distance;
22  void setup() {
23    Serial.begin(115200);
24    pinMode(trigPin, OUTPUT);
25    pinMode(echoPin, INPUT);
26    wifiConnect();
27    mqttConnect();
28  }
29  void loop()
30  {
31    digitalWrite(trigPin, LOW);
32    delayMicroseconds(2);
33    digitalWrite(trigPin, HIGH);
34    delayMicroseconds(10);
35    digitalWrite(trigPin, LOW);
36    duration = pulseIn(echoPin, HIGH);
37    distance = (duration * SOUND_SPEED) / 2;
38    Serial.println("Distance: " + String(distance));
39    delay(1000);
40  }

```

Simulation

00:03.899 44%



Connecting to ...
 WiFi connected
 IP address:
 10.10.0.2
 Reconnecting client to kabw0z.messaging.internetofthings.ibmcloud.com

Output

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

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes the platform name and a user profile. The left sidebar contains icons for various dashboard sections. The main content area is titled 'Browse' and shows a table of recent events for a specific device. The table has four columns: Event, Value, Format, and Last Received. The events listed are all 'Data' events with a value of '{"Normal Distance":58.99}' or '{"Normal Distance":58.96}' in 'json' format, received 'a few seconds ago'. A pagination bar at the bottom indicates 'Items per page 50' and '1 of 1 page'.

IBM Watson IoT Platform

19ec014@acetcb.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 < 1 >