

Assignment -4

Submitted by

A. Sri Veni

Roll no: 961819106050

ECE

Ponjesly College of Engineering

Batch no: B12-6A2E

Question-1:

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

Code :

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "hycgw4"
#define DEVICE_TYPE "Distance"
#define DEVICE_ID "Ultrasonic"
#define TOKEN "WD6Mb(-d2F+X0xWqnB"
#define speed 0.034 #define led 14 char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; char publishTopic[] = "iot2/evt/event2/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);

const int trigpin=5; const int
echopin=18;
String command; String data="";
long duration;
float dist;

void setup()
{
  Serial.begin(115200); pinMode(led, OUTPUT);
  pinMode(trigpin, OUTPUT); pinMode(echopin,
  INPUT); wifiConnect(); mqttConnect();
} void loop() { bool isNearby = dist <
100; digitalWrite(led, isNearby);
publishData();
delay(500);
if (!client.loop()) {
mqttConnect();
}
}
void wifiConnect() {
```

```

    Serial.print("Connecting to "); Serial.print("Wifi");
    WiFi.begin("WokwiGUEST", "", 6); while (WiFi.status() !=
    WL_CONNECTED) {    delay(500);
        Serial.print(".");
    }
    Serial.print("WiFi connected, IP address: ");
    Serial.println(WiFi.localIP());
} 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 = "{\"Alert!! Alert!! Distance\":\"";    payload += dist;
        payload += "\"}";

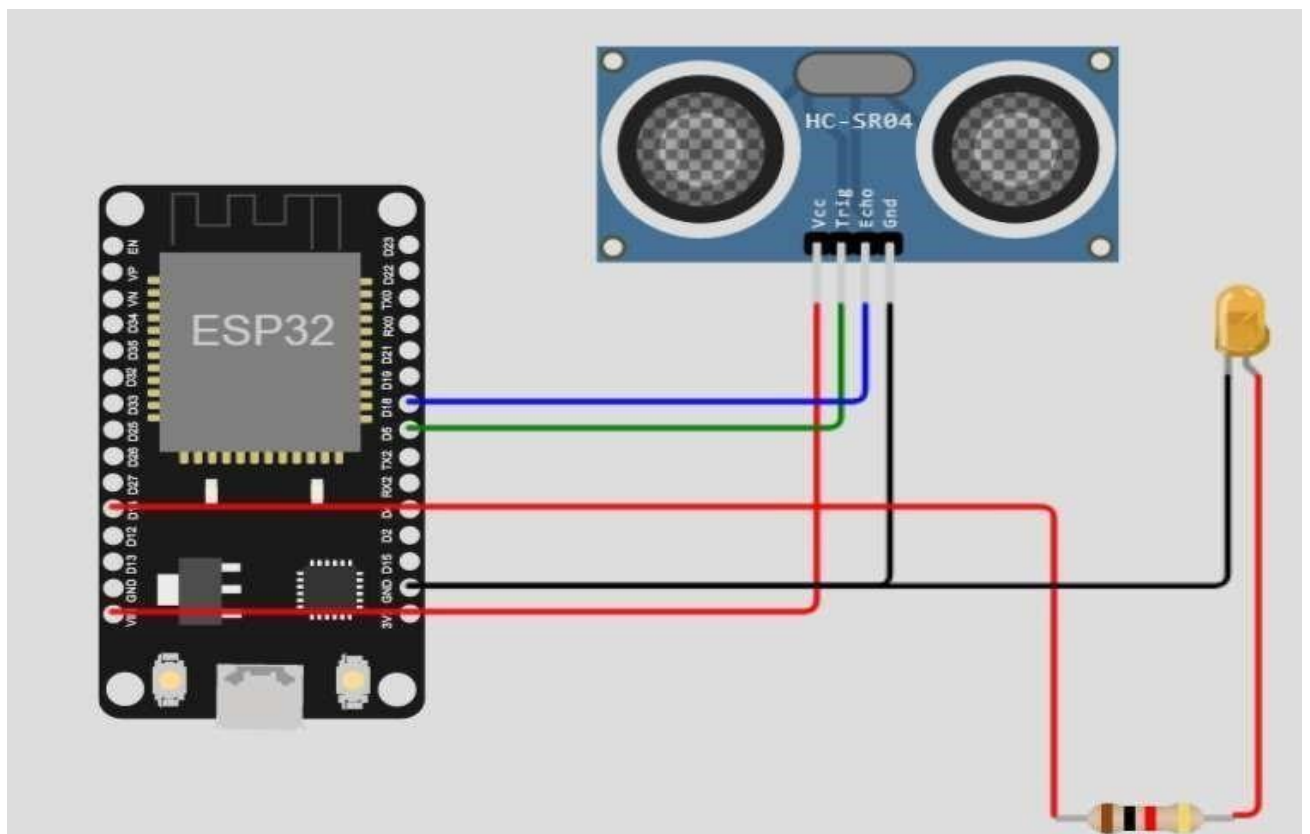
        Serial.print("\n");
        Serial.print("Sending payload: ");    Serial.println(payload);
        if (client.publish(publishTopic, (char*) payload.c_str())) {
            Serial.println("Publish OK");
        }
    }
    if(dist>100){
        String payload = "{\"Distance\":\"";    payload += dist;
        payload += "\"}";

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

```

```
}  
}  
  
}
```

Connections:



WOKWI AND IBM CLOUD CONNECTED:

Browse

Action

Device Types

Interfaces

Add Device

Browse Devices

All Devices

Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Q

Search by Device ID

Device Simulator

	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
>	12	Disconnected	abod	Device	Oct 12, 2022 6:39 PM	
>	Ultrasonic	Connected	Distance	Device	Oct 25, 2022 7:04 PM	→ ...

Items per page 50

Ultrasonic

of 2 items

1 of 1 page

<

1

>

1 Simulation running

OUTPUT:

- Distance = 95 cm Status = Alert Message

sketch.ino

diagram.json

libraries.txt

Library Manager

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "hycgw4"
6 #define DEVICE_TYPE "Distance"
7 #define DEVICE_ID "Ultrasonic"
8 #define TOKEN "ND6Mb(-d2F+X0xWqNB"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/event2/fmt/json";
13 char topic[] = "iot-2/cmd/home/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wificlient);
18
19
20
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
32 {
33   Serial.begin(115200);
34   pinMode(led, OUTPUT);

```

Simulation

00:25.433

99%

Editing Ultrasonic Distance Sensor

Distance: 95cm

Connecting to Wifi.....WiFi connected, IP address: 10.10.0.2

Reconnecting MQTT client to hycgw4.messaging.internetofthings.ibmcloud.com

IBM subscribe to cmd OK

Sending payload: {"Alert!! Alert!! Distance":95.03}

Publish OK

Items per page 50 | 1-2 of 2 items

Event	Value	Format	Last Received
event2	{"Alert!! Alert!! Distance":94.98}	json	a few seconds ago
event2	{"Alert!! Alert!! Distance":94.98}	json	a few seconds ago
event2	{"Alert!! Alert!! Distance":94.98}	json	a few seconds ago
event2	{"Alert!! Alert!! Distance":94.98}	json	a few seconds ago
event2	{"Alert!! Alert!! Distance":94.98}	json	a few seconds ago

1 Simulation running

Wokwi data publishing to ibm cloud

2. Distance = 162 cm Status = Normal

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "hycgwa"
6 #define DEVICE_TYPE "Distance"
7 #define DEVICE_ID "Ultrasonic"
8 #define TOKEN "wD6Mb(-d2f+X0xkqNB"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/event2/fmt/json";
13 char topic[] = "iot-2/cmd/home/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wificlient);
18
19
20
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
32 {
33   Serial.begin(115200);
  
```

Simulation

00:54.34 96%

Editing Ultrasonic Distance Sensor

Distance: 162cm

Publish OK

Sending payload: {"Distance":162.25}

Publish OK

Sending payload: {"Distance":161.94}

Publish OK

Browse Action Device Types Interfaces
 Add Device

Ultrasonic Connected Distance Device Oct 25, 2022 7:04 PM
 → ...

Identity Device Information **Recent Events** State Logs
 ×

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

Event	Value	Format	Last Received
event2	{"Distance":161.94}	json	a few seconds ago
event2	{"Distance":161.94}	json	a few seconds ago
event2	{"Distance":161.94}	json	a few seconds ago
event2	{"Distance":161.94}	json	a few seconds ago
event2	{"Distance":161.94}	json	a few seconds ago

Items per page 50 | 1-2 of 2 items

1 Simulation running

3. Distance = 27 cm Status = Alert Message

sketch.ino diagram.json libraries.txt Library Manager

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "hycgw4"
6 #define DEVICE_TYPE "Distance"
7 #define DEVICE_ID "Ultrasonic"
8 #define TOKEN "wD6Mb(-d2F+X0x0qN8"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/event2/fmt/json";
13 char topic[] = "iot-2/cmd/home/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wificlient);
18
19
20
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
32 {
33   Serial.begin(115200);
34 }

```

Simulation

01:03.901 99%

Editing Ultrasonic Distance Sensor

Distance: 27cm

Publish OK

Sending payload: {"Alert!! Alert!! Distance":26.98}

Publish OK

Sending payload: {"Alert!! Alert!! Distance":26.98}

Publish OK

