

Assignment - 4 Wowki & IBM Cloud

Assignment Date	27 October 2022
Student Name	Sudharshan.S
Student Roll Number	310819106083
Maximum Marks	2 Marks

Question-1:

Write code and connections in wowki for the ultrasonic sensor. Whenever the distance is less than 100cms sent "alert" to IBM cloud and display in device recent events.

Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

WiFiClient wifiClient;

#define ORG "42cic2"
#define DEVICE_TYPE "sudharshanlaptop"
#define DEVICE_ID "12345"
#define TOKEN "*zSbeu!vc(!+Xn@520 "
#define speed 0.034
char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; char publishTopic[]
= "iot-2/evt/abcd_1/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();
const int trigpin=5;
const int echopin=18;
String command;
String data="";
String lat="14.167589";
String lon="80.248510";
String name="point2";
String icon="";
```

```

    long
duration; int
dist;
void setup()
{
    Serial.begin(115200);
    pinMode(trigpin, OUTPUT);
    pinMode(echopin, INPUT);
    wifiConnect();    mqttConnect();
}
void loop() {

publishData();
delay(500);
    if (!client.loop())
{
    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 mqttConnect() {    if
(!client.connected()) {
        Serial.print("Reconnecting MQTT client to "); Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print(".");    delay(1000);
    }
    initManagedDevice();
    Serial.println();
}
} void initManagedDevice() {
if (client.subscribe(topic)) {
    Serial.println(client.subscribe(topic));
    Serial.println("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){
dist=100-dist;
icon="fa-trash";

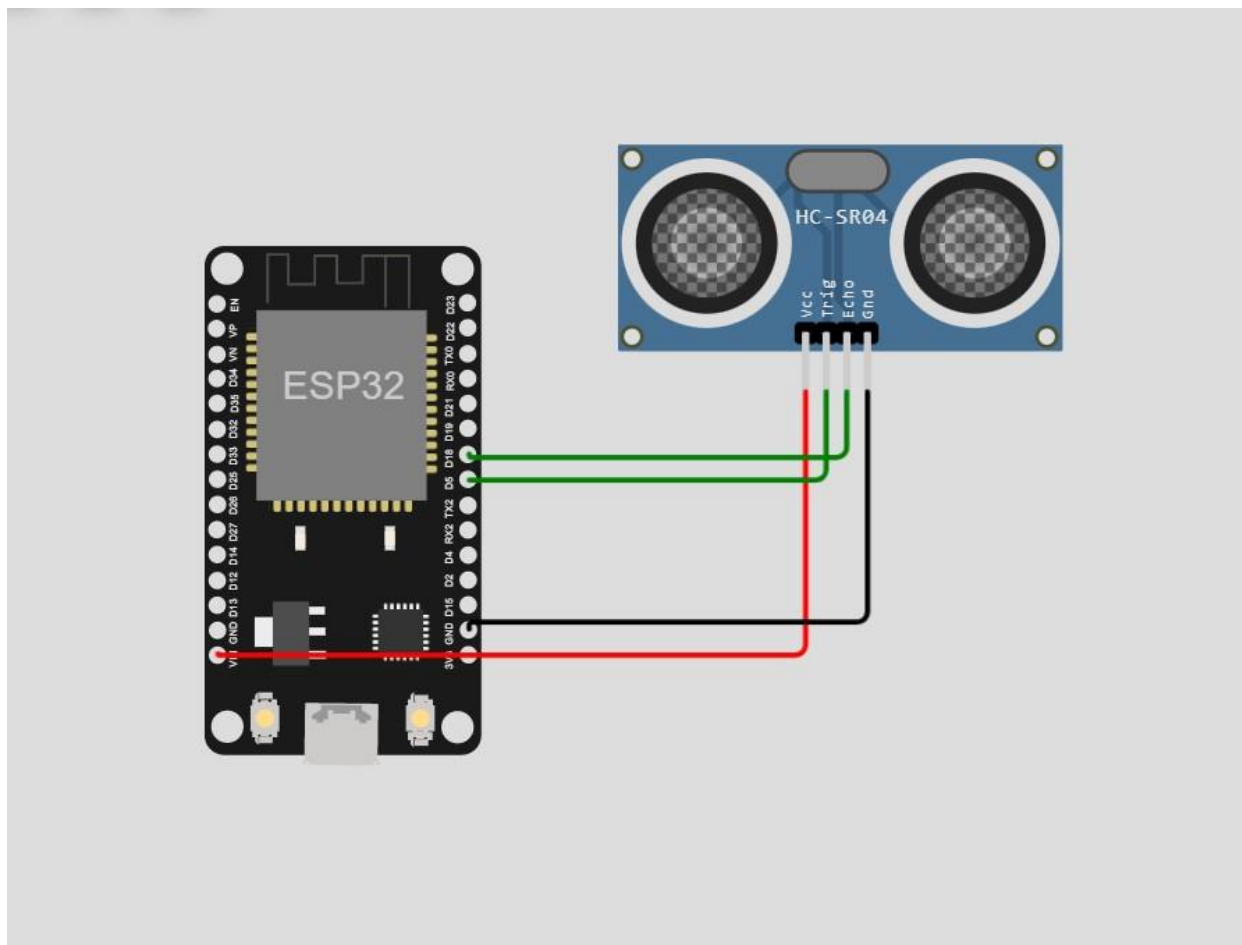
```

```

}else{    dist=0;
icon="fa-trash-o";
    }
    DynamicJsonDocument doc(1024);
String payload;    doc["Name"]=name;
doc["Latitude"]=lat;
doc["Longitude"]=lon;
doc["Icon"]=icon;
doc["FillPercent"]=dist;
serializeJson(doc, payload);
delay(3000);
    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:



Output:

The screenshot shows the Wokwi IoT simulation interface. On the left, the 'sketch.ino' file contains the following code:

```
14 char publishTopic[] = "iot-2/evt/abcd_1/fmt/json";
15 char topic[] = "iot-2/cmd/home/fmt/string";
16 char authMethod[] = "use-token-auth";
17 char token[] = TOKEN;
18 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
19 PubSubClient client(server, 1883, wifiClient);
20 void publishData();
21
22 const int trigpin=5;
23 const int echopin=18;
24 String command;
25 String data="";
26 String lat="14.167589";
27 String lon="80.248510";
28 String name="point2";
29 String icon="";
30
31 long duration;
32 int dist;
33
34 void setup()
35 {
36   Serial.begin(115200);
37   pinMode(trigpin, OUTPUT);
38   pinMode(echopin, INPUT);
39   wifiConnect();
40   mqttConnect();
41 }
42
43 void loop() {
44   publishData();
45   delay(500);
46 }
47
```

On the right, a simulation of an ESP32 microcontroller is shown connected to an HC-SR04 ultrasonic sensor. The sensor's VCC pin is connected to the ESP32's 5V pin, GND to GND, and the Trig pin to pin 5. The Echo pin is connected to pin 18. The simulation status bar at the top right shows a timer at 00:14:180 and 89% completion.

Below the circuit, the console output shows:

```
trash,"FillPercent":47}
Publish OK

Sending payload:
{"Name":"point2","Latitude":"14.167589","Longitude":"80.248510","Icon":"fa-trash","FillPercent":47}
Publish OK
```

Output:(IBM Cloud)

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area displays a table of recent events for a device with ID 42cic2.

Event	Value	Format	Last Received
event_1	{"alert distance":85}	json	a few seconds ago
event_1	{"alert distance":15}	json	a few seconds ago
event_1	{"alert distance":12}	json	a few seconds ago
event_1	{"alert distance":33}	json	a few seconds ago
event_1	{"alert distance":33}	json	a few seconds ago

At the bottom of the dashboard, a status bar indicates 'Simulation running'.

Link :<https://wokwi.com/projects/346857404558738004>