

Assignment -4 Wowki & IBM Cloud

| | |
|----------------------------|-----------------|
| Assignment Date | 04 October 2022 |
| Student Name | Gopikrishnan K |
| Student Roll Number | 310819205033 |
| 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;

String data3;

#define ORG "sf54ab"

#define DEVICE_TYPE "MyDeviceType"

#define DEVICE_ID "12345"

#define TOKEN "ZIA1zf@zL9s)Pumv11"

#define speed 0.034 #define led 14 char server[] = ORG

".messaging.internetofthings.ibmcloud.com"; char publishTopic[]

= "iot-2/evt/shreedharen/fmt/json";
```

```

char topic[] = "iot-2/cmd/led/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("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(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 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");

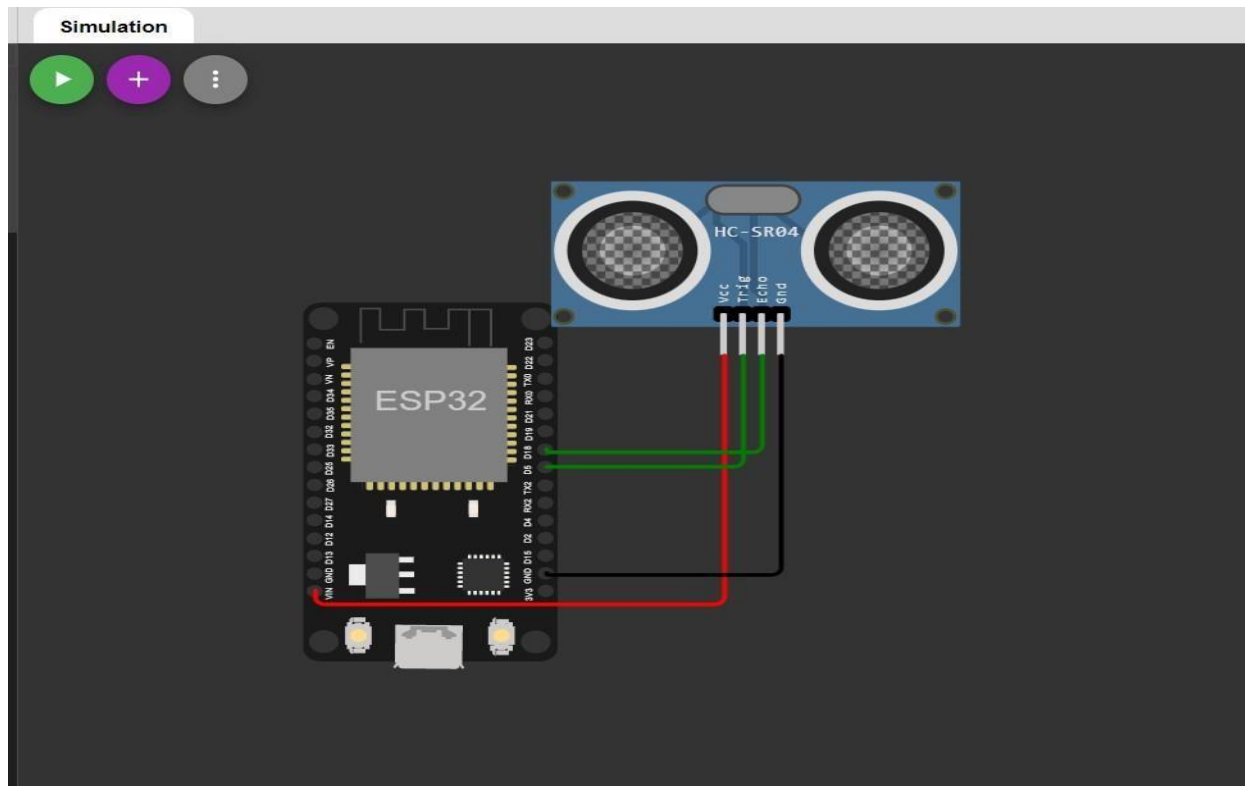
}

}

}

```

Connections:



Output:

```

1  #include <WiFi.h>
2  #include <PubSubClient.h>
3  #include <ArduinoJson.h>
4
5  WiFiClient wifiClient;
6
7  #define ORG "sf54ab"
8  #define DEVICE_TYPE "MyDevicetype"
9  #define DEVICE_ID "12345"
10 #define TOKEN "Z1A1zf@2L9s)Pumv11"
11 #define speed 0.034
12
13 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
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

```

Simulation
00:34.720 100%

Editing Ultrasonic Distance Sensor

✕

Distance: 76cm

subscribe to cmd OK
✕

Sending payload:

```
{
  "Name": "point2",
  "Latitude": "14.167589",
  "Longitude": "80.248510",
  "Icon": "fa-trash",
  "FillPercent": 25
}
```

Publish OK
✕

Output:(IBM Cloud)

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various IoT-related functions. The main content area shows a device with ID '12345' in a 'Disconnected' state. Below this, a 'Recent Events' tab is selected, displaying a table of events. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. The events are all of type 'event_test' and contain JSON data about alert distances. A status message at the bottom right indicates '1 Simulation running'. The browser's address bar shows the URL 'sf54ab.internetofthings.ibmcloud.com/dashboard/devices/browse'. The bottom of the image shows a Windows taskbar with various application icons and a system tray displaying the date and time.

| Event | Value | Format | Last Received |
|------------|-----------------------|--------|-------------------|
| event_test | {"Alert distance":26} | json | a few seconds ago |
| event_test | {"Alert distance":80} | json | a few seconds ago |
| event_test | {"Alert distance":20} | json | a few seconds ago |
| event_test | {"Alert distance":52} | json | a few seconds ago |
| event_test | {"Alert distance":13} | json | a few seconds ago |

Link : <https://wokwi.com/projects/346587874175484499>