

Assignment - 4

Wowki & IBM Cloud

Assignment Date	13 November 2022
Student Name	Akshaya R
Student Roll Number	211719104011
Maximum Marks	2 Marks

Question-1:

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

Code:

```
1 #include <WiFi.h>
#include <PubSubClient.h>
#include <Arduino.h>

WiFiClient wifiClient;

#define ORG "oa8490"
#define DEVICE_TYPE "TestDeviceType"
#define DEVICE_ID "12345"
#define TOKEN "-Al0raS44flfdjYBVS"
#define speed 0.034

1 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/abcd1/fms/json"; char topic[]
- "iot-2/cv d/home/but/Strinq" i char anrhMeshod[] - "use-token-
anth"; char token[] = TOKEN;

char ClientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID:
PubSubClient client(server, 1883, wifiClient);
publishData();
```

```

const int trigpin=5;
const int echopin=13;
String command;

String data=" ";
String lat="14. 167589";
String lon="80.248510";

String name="point2":
String icon="";

```

```

long duration;
int dist;

```

```

void zerap ()
( Serial.b-gi (115200) ;
  p- " :e(trigpin,
    OUTPUT)
  p- +(echopin, lPUT) ;
  wifiConnect():
  mqttConnect();
}

```

```

void loop() (

  publishData();
  - (500) ;

  if (!client.loop()) (
    mqXtZonectC;
  )
}
+ - - - - -

```

```

void wifiConnect() (
  Serial.print("Connecting to ");
  Serial.:: ("Wifi") ; WiFi.h=j ("Wokwi-GUEST", ,
  ;_while WiFi .status() != WL_CONNECTED) (

    r Se ial .1: (" .") ;

    Serial.¿== ("WiFi connected, lP address: ") ;
  Serial. : ( WiFi.localIP())
}

```

```

void mqttConnect() {
    if (! client.connected()) {
        Serial.pr n'("Reconnecting MQTT client to ") ;
        Serial.p >r l server); while (!client.connect(cli'entId,
        authMethod, taken) ( Serial.p =' (".")) ;
        delay(1000) ;
        } + - - - - -
        'initManagedDevice();
        Serial.pt_ >_ u() ;
    }
}

```

```

void initManagedDevice() {
    if ( client.subscribe(topic)) {
        Serial.pi: : .( client.subscri'be(topic));
        Ser lal . >i i* i("s u b s c r i b e   t o   c m d   O K" ) ;
    } : else {
        Serial.p ' > ' > ("subscribe to cmd FAILED") ;
    } } void
publishData() {
    ( I'm' ' !W °'••(trigpin,LOW)
    . a?' !'i' ••(trigpin,HIGH) ;'
    ... .. .... =*..1...
    q3' !:< i_=(trigpin,LQW) ;
    dur0tion=pu.'c ':(echopin,MIGM)
    ; dist=duration*speed/2;
}

```

```

if(dist<100){
    dist=100- dist; icon="fa-
    trash";
} else( distr:
    icon="fa-trasN-
    _O";
}

```

```

DynamicJ sonDocument doc (1024) i
String payload; doc{"Name"}= I
' name; doc{"Latitude"}= lat;
doc{"Longitude"}= lon;
doc["Icon"]=icon;
doc{"FillPercent"}= diet;
serializeJ son(doc, payload);
delay(3000) ;
Serial.print("\n") ,

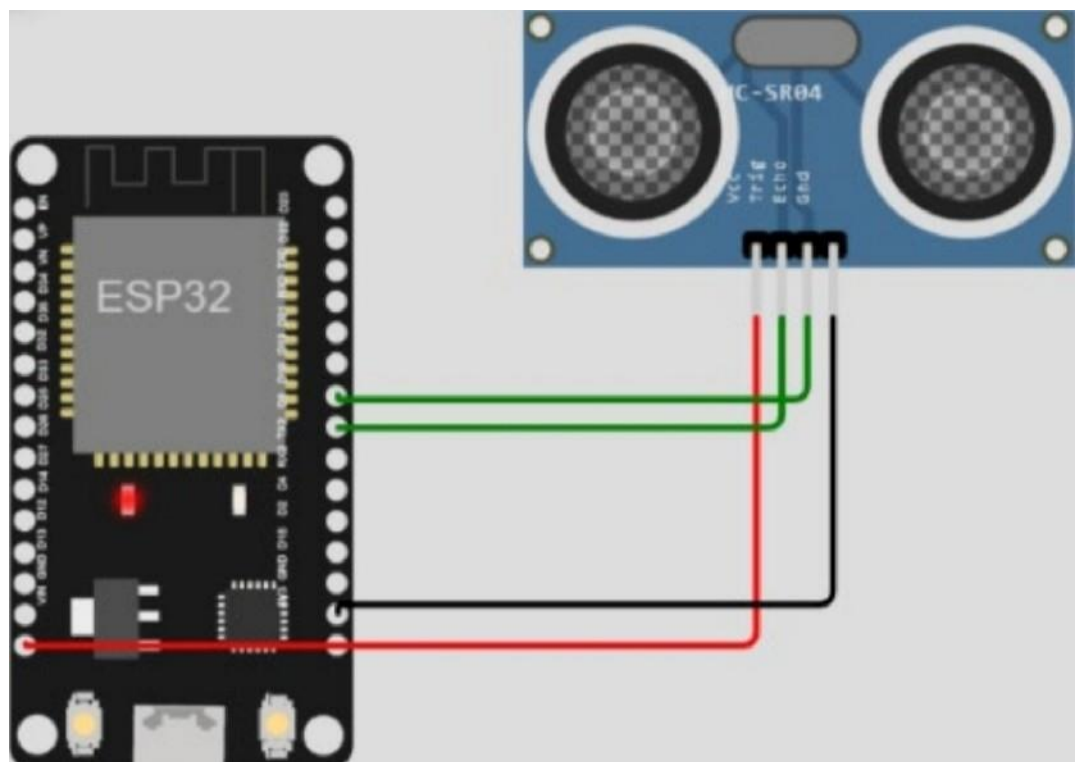
```

```

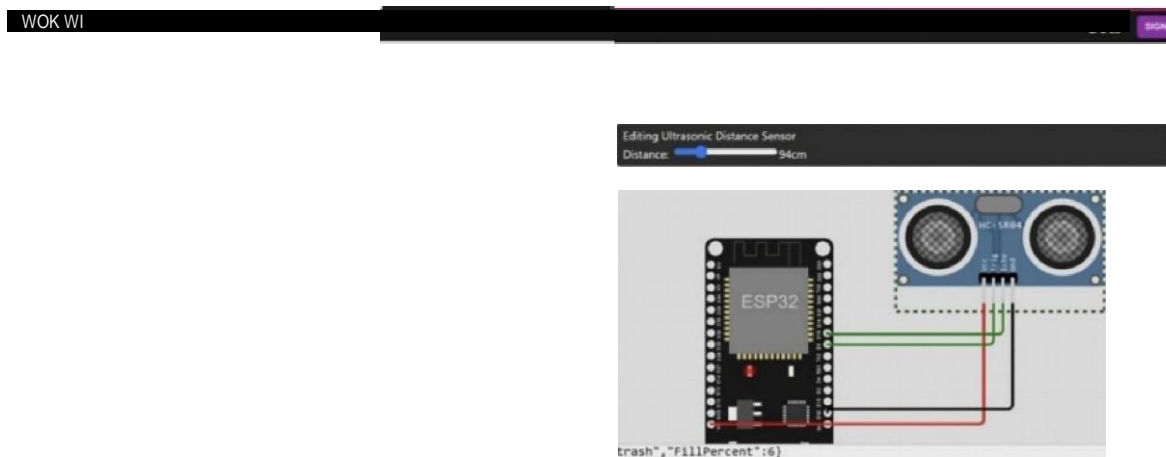
Serial.p  .' ("lending payload: ") ; I
1 Serial.p _.' .( payload):      _      I
' if (client.publish(publishTopic, (ohar') payload.c_str()))
  ( Serial.p  .' .("Publish OK") ;
) else 1
'   Serial.)  .' .("Publish FAILED") ; I

```

Connecdons:



Output:



Output :(IBM Cloud)

The screenshot displays the IBM Watson IoT Platform web interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area shows a table of devices with columns for Device ID, Status, Device Type, Class ID, Date Added, and Descriptive Location. One device with ID '12345' is listed as 'Disconnected'. Below the table, the 'Recent Events' tab is selected, showing a list of events. The events are displayed in a table with columns for Event, Value, Format, and Last Received. The events are JSON payloads containing distance measurements. At the bottom right, there is a '1 Simulation running' status and an 'Activate Windows' watermark.

Event	Value	Format	Last Received
event_1	{\"Alert Distance\":8}	json	a few seconds ago
event_1	{\"Alert Distance\":81}	json	a few seconds ago
event_1	{\"Alert Distance\":56}	json	a few seconds ago
event_1	{\"Alert Distance\":98}	json	a few seconds ago
event_1	{\"Alert Distance\":72}	json	a few seconds ago