

**Assignment -4**  
Python Programming

Assignment Date	03 November 2022
Student Name	A .Asmitha
Student Roll Number	960219106042
Maximum Marks	2 Marks

**Question:**

Write a code and make a connection in WOKWI for ultrasonic sensor. Whenever distance is less than 100, send "alert" to IBM cloud and display in device recent events.

**Program :**

```
#include <WiFi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG "atay8h"

#define DEVICE_TYPE "asmidevicetype"

#define DEVICE_ID "asmideviceid"

#define TOKEN "DXTJdUSEXiFPh9d(iH"

#define speed 0.034

#define led 14

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/Asmitha/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");

}

}

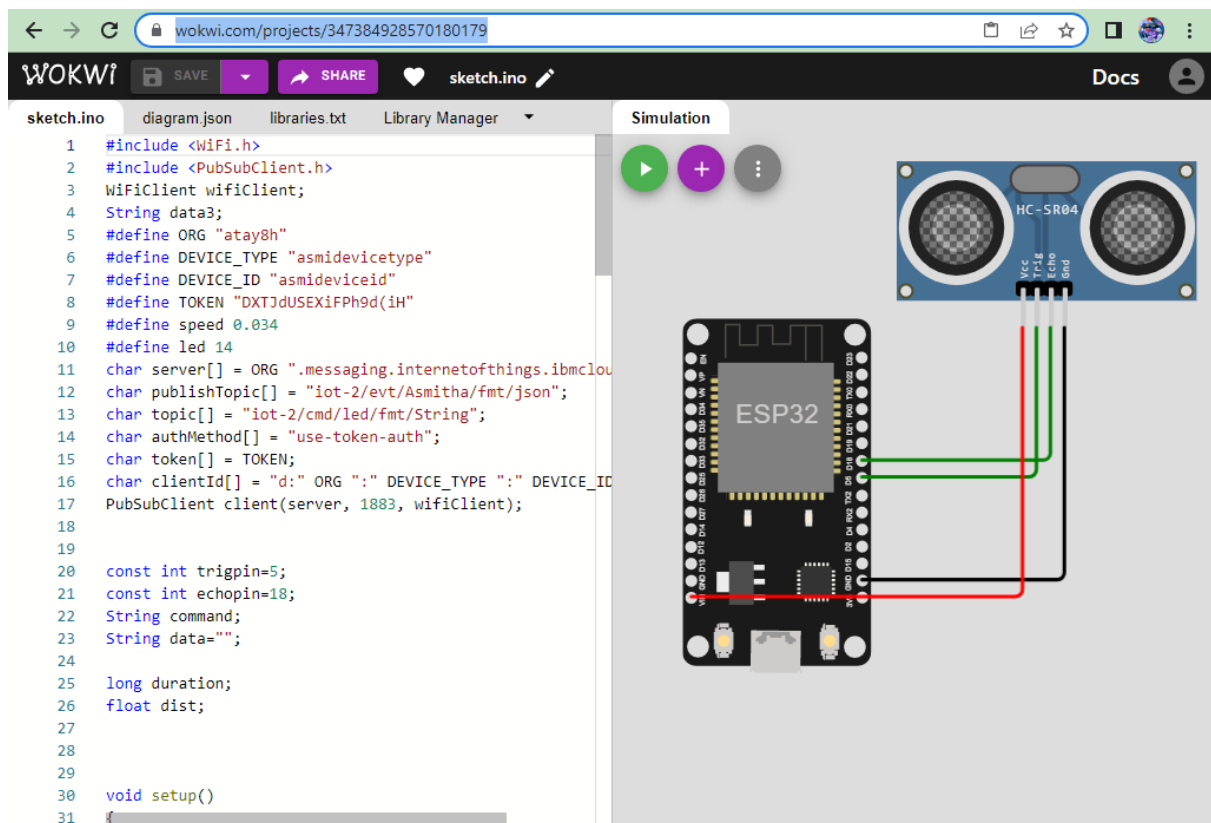
}

}

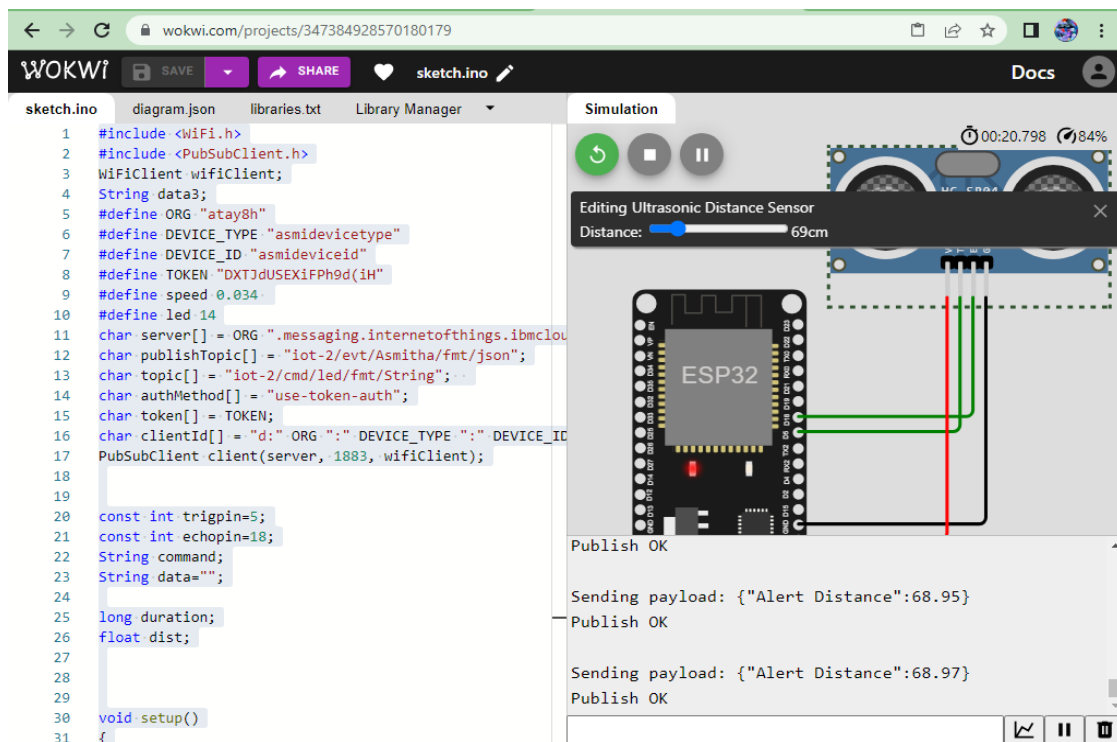
```

**OUTPUT:**

**WOKWI SIMULATION:**



When distance<100:



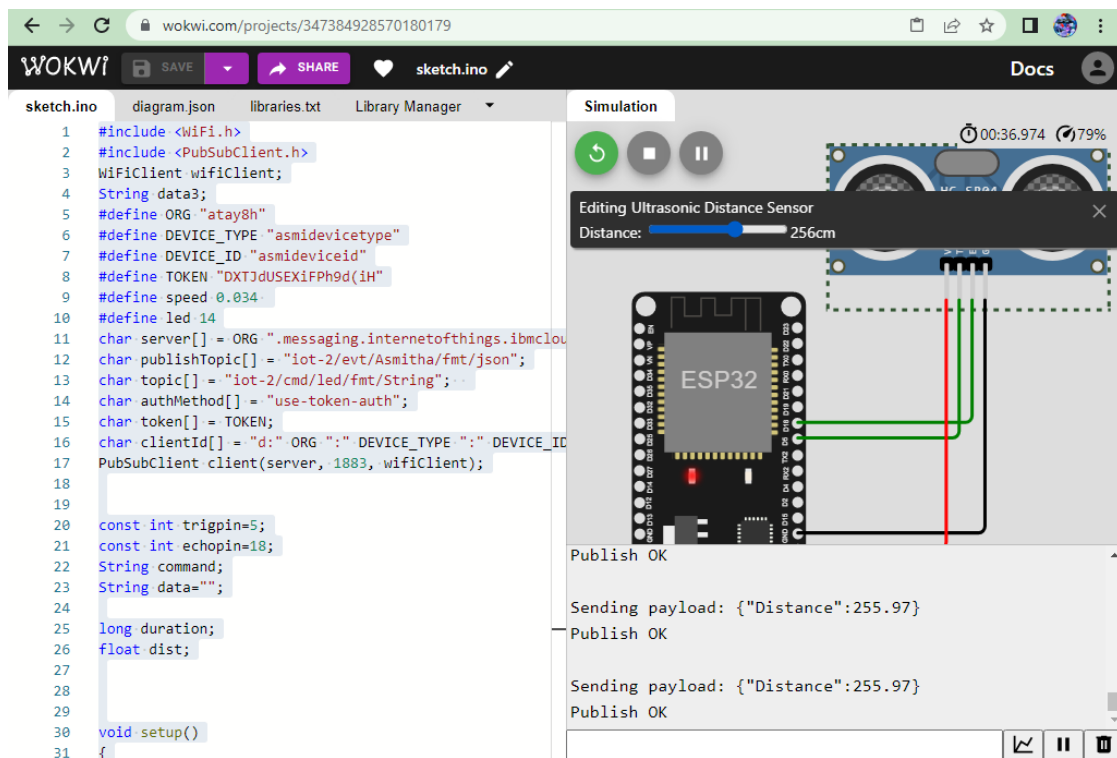
The screenshot shows the Wokwi web interface with a project titled "wokwi.com/projects/347384928570180179". The left pane displays the sketch code for an ESP32-based IoT device. The code includes libraries for WiFi and PubSubClient, and defines various constants and variables. The right pane shows a simulation of the hardware, including an ESP32 board and an Ultrasonic Distance Sensor. A dialog box titled "Editing Ultrasonic Distance Sensor" is open, showing a distance of 69cm. The simulation output shows the device sending payloads with alert distances.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "atay8h"
6 #define DEVICE_TYPE "asmidevicetype"
7 #define DEVICE_ID "asmideviceid"
8 #define TOKEN "DXTJdUSEXIFPh9d(iH"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.iot";
12 char publishTopic[] = "iot-2/evt/Asmitha/fmt/json";
13 char topic[] = "iot-2/cmd/led/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 const int trigpin=5;
21 const int echopin=18;
22 String command;
23 String data="";
24
25 long duration;
26 float dist;
27
28
29
30 void setup()
31 {
```

Simulation output:

```
Publish OK
Sending payload: {"Alert Distance":68.95}
Publish OK
Sending payload: {"Alert Distance":68.97}
Publish OK
```

When distance>100:



The screenshot shows the Wokwi web interface with the same project. The left pane displays the sketch code, which is identical to the previous one. The right pane shows the simulation of the hardware. A dialog box titled "Editing Ultrasonic Distance Sensor" is open, showing a distance of 256cm. The simulation output shows the device sending payloads with distances.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "atay8h"
6 #define DEVICE_TYPE "asmidevicetype"
7 #define DEVICE_ID "asmideviceid"
8 #define TOKEN "DXTJdUSEXIFPh9d(iH"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.iot";
12 char publishTopic[] = "iot-2/evt/Asmitha/fmt/json";
13 char topic[] = "iot-2/cmd/led/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 const int trigpin=5;
21 const int echopin=18;
22 String command;
23 String data="";
24
25 long duration;
26 float dist;
27
28
29
30 void setup()
31 {
```

Simulation output:

```
Publish OK
Sending payload: {"Distance":255.97}
Publish OK
Sending payload: {"Distance":255.97}
Publish OK
```

## IBM CLOUD OUTPUT:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes the platform name, a user profile, and a search bar. The main content area shows a list of devices, with one device selected and its details expanded. The 'Recent Events' tab is active, showing a stream of data events.

Device ID	Status	Device Type	Class ID
asmideviceid	Connected	asmidevicetype	Device

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	
Asmitha	{"Alert Distance":51.97}	json	a few seconds ago	
Asmitha	{"Alert Distance":51.97}	json	a few seconds ago	
Asmitha	{"Alert Distance":51.97}	json	a few seconds ago	
Asmitha	{"Alert Distance":51.99}	json	a few seconds ago	
Asmitha	{"Distance":100.98}	json	a few seconds ago	

## WOKWI LINK:

<https://wokwi.com/projects/347384928570180179>

