

ASSIGNMENT 4

Ultrasonic sensor simulation in Wokwi

Team ID : PNT2022TMID25209

IBM ID : IBM-Project- 54539-1662208275

Question :

Write a code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100cms send an “Alert” to IBM cloud and display in the device recent events.

Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "vg9s67"
#define DEVICE_TYPE "sethuass0004"
#define DEVICE_ID "ass04"
#define TOKEN "1234567890"
#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");
}
}
}

```

diagram.json:

```

{
  "version": 1,
  "author": "Sethupathi",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -167.12, "left": 63.69, "attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -54.66, "left": -160.76, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v101.24", "h-228.44" ] ],
    [ "ultrasonic1:TRIG", "esp:D5", "green", [ "v33.9", "h-138.33" ] ],
    [ "ultrasonic1:ECHO", "esp:D18", "green", [ "v25.24", "h-145.56" ] ],
    [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v88.57", "h-152.78" ] ]
  ]
}

```

Simulation & Output:

The screenshot shows the Wokwi IDE interface. On the left, the Arduino IDE editor displays the code for 'asse04.ino'. The code includes libraries for WiFi and PubSubClient, defines device credentials, and sets up a MQTT client that publishes data to a topic. The main loop publishes data every 500ms. On the right, the 'Simulation' window shows a 3D model of an ESP32 board connected to a blue HX-1800 module. The code is as follows:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4 WiFiClient wifiClient;
5 #define ORG "vg9s67"
6 #define DEVICE_TYPE "sethuass0004"
7 #define DEVICE_ID "ass04"
8 #define TOKEN "1234567890"
9 #define speed 0.034
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/abcd/1/fmt/json";
12 char topic[] = "iot-2/cnd/home/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
16 PubSubClient client(server, 1883, wifiClient);
17 void publishData();
18 const int trigpin=5;
19 const int echopin=18;
20 String command;
21 String data="";
22 String lat="14.167589";
23 String lon="80.248510";
24 String name="point2";
25 String icon="";
26 long duration;
27 int dist;
28 void setup()
29 {
30   Serial.begin(115200);
31   pinMode(trigpin, OUTPUT);
32   pinMode(echopin, INPUT);
33   wifiConnect();
34   mqttConnect();
35 }
36 void loop() {
37   publishData();
38   delay(500);
39   if (!client.loop()) {
```

IBM Watson IOT connection:

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area displays a table of devices. The first device, 'ass04', is in a 'Connected' state. Below the table, a detailed view of the device is shown, including its identity, device information, recent events, state, and logs. The device information section shows the following details:

Device ID	Status	Device Type	Class ID	Date Added
ass04	Connected	sethuass0004	Device	Nov 13, 2022 4:41 PM

Below the table, the device details are listed:

- Device ID:** ass04
- Device Type:** sethuass0004
- Date Added:** Nov 13, 2022 4:41 PM
- Added By:** nspgipsy@gmail.com
- Connection Status:** Connected
Connection Time: Nov 13, 2022 4:55 PM
Client Address: 50.31.197.64 Insecure

At the bottom of the dashboard, it shows 'Items per page 50' and '1-1 of 1 item'. On the right side, it indicates '0 Simulations running'.

Cloud output:

IBM Watson IoT Platform

nsppgpsy@gmail.com
ID: vg9s67

?

Browse

Action

Device Types

Interfaces

Add Device

Device ID

Status

Device Type

Class ID

Date Added

ass04

Connected

sethuass0004

Device

Nov 13, 2022 4:41 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
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago
abcd_1	{"Name":"point2","Latitude":"14.167589","Longi...	json	a few seconds ago

0 Simulations running