

## Assignment 4

### Wokwi Assignment

Assignment Date	29 October 2022
Student Name	Ms. Chilakam Bindu Reddy
Student Roll Number	711119104018
Maximum Marks	2 Marks

Assignment Question:

Write code and connections in wokwi for ultrasonic sensor.

Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events.

Wokowi Link: <https://wokwi.com/projects/346504267904844371>

Program code:

```
#include <WiFi.h> //library for wifi
```

```
#include <PubSubClient.h> //library for MQTT
```

```
#define ECHO_GPIO 12
```

```
#define TRIGGER_GPIO 13
```

```
#define MAX_DISTANCE_CM 100 // Maximum of 5 meters
```

```
#include "Ultrasonic.h"
```

```
Ultrasonic ultrasonic(13, 12);
```

```
int distance;
```

```
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
```

```
//-----credentials of IBM Accounts-----
```

```

#define ORG "z35ag6"//IBM ORGANITION ID

#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform

#define DEVICE_ID "ESP"//Device ID mentioned in ibm watson IOT Platform

#define TOKEN "123456789" //Token

String data3;

float h, t;


//----- Customise the above values -----

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

char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in
which data to be send

char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command type AND
COMMAND IS TEST OF FORMAT STRING

char authMethod[] = "use-token-auth";// authentication method

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----

WiFiClient wifiClient; // creating the instance for wificlient

PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client id by passing
parameter like server id,portand wificredential


void setup()// configureing the ESP32
{
    Serial.begin(115200);

    delay(10);

    Serial.println();

    wificonnect();

    mqttconnect();

```

```
}
```

```
void loop()// Recursive Function
```

```
{
```

```
distance = ultrasonic.read(CM);
```

```
if (distance < 100) {
```

```
    Serial.print("Distance in CM: ");
```

```
    Serial.println(distance);
```

```
    PublishData(distance);
```

```
    delay(1000);
```

```
    if (!client.loop()) {
```

```
        mqttconnect();
```

```
    }
```

```
}
```

```
delay(1000);
```

```
}
```

```
/*.....retrieving to Cloud.....*/
```

```
void PublishData(float temp) {
```

```
    mqttconnect();//function call for connecting to ibm
```

```
    /*
```

```
        creating the String in in form JSon to update the data to ibm cloud
```

```
    */
```

```
    String payload = "{\"Alert Distance\":\"";
```

```
payload += temp;
```

```
payload += "}";
```

```
Serial.print("Sending payload: ");
```

```
Serial.println(payload);
```

```
if (client.publish(publishTopic, (char*) payload.c_str())) {
```

```
    Serial.println("Publish ok");// if it successfully upload data on the cloud then it will print publish ok
```

```
in Serial monitor or else it will print publish failed
```

```
    } else {
```

```
        Serial.println("Publish failed");
```

```
    }
```

```
}
```

```
void mqttconnect() {
```

```
    if (!client.connected()) {
```

```
        Serial.print("Reconnecting client to ");
```

```
        Serial.println(server);
```

```
        while (!client.connect(clientId, authMethod, token)) {
```

```
            Serial.print(".");
```

```
            delay(500);
```

```
        }
```

```
        initManagedDevice();
```

```
        Serial.println();
```

```
    }
```

```
}
```

```
void wificonnect() //function definition for wificonnect
```

```

{
    Serial.println();

    Serial.print("Connecting to ");

    WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void initManagedDevice() {
    if (client.subscribe(subscribetopic)) {
        Serial.println((subscribetopic));
        Serial.println("subscribe to cmd OK");
    } else {
        Serial.println("subscribe to cmd FAILED");
    }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
    Serial.print("callback invoked for topic: ");
    Serial.println(subscribetopic);
    for (int i = 0; i < payloadLength; i++) {

```

```

//Serial.print((char)payload[i]);

data3 += (char)payload[i];
}

Serial.println("data: " + data3);

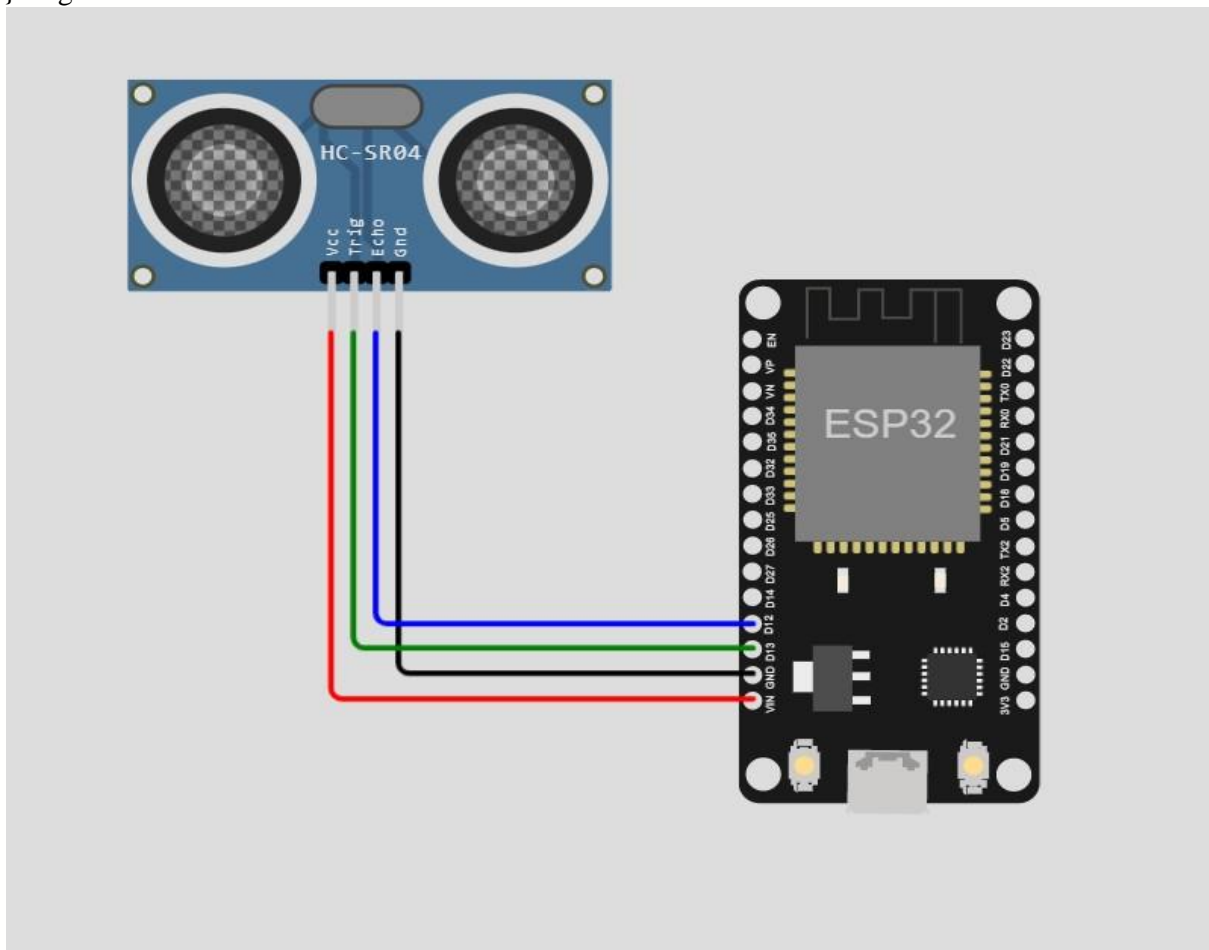
if (data3 == "lighton")
{
    Serial.println(data3);
}

else

{
    Serial.println(data3);
}

data3 = "";
}
Diagram:

```



When the distance is less than 100cm the alert is sent to the IBM cloud

Service Details - IBM Cloud

IBM Watson IoT Platform

z35ag6.internetofthings.ibmcloud.com/dashboard/devices/drilldown/ESP32-ESP?returnTo=/devices/browse

chr711119104018@it.ac.in  
ID: z35ag6

IBM Watson IoT Platform

Back

Device Drilldown - ESP

Connection Information

Recent Events

State

Device Information

Metadata

Diagnostics

Connection Logs

Device Actions

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"Alert Distance":91}	json	a few seconds ago
Data	{"Alert Distance":91}	json	a few seconds ago
Data	{"Alert Distance":91}	json	a few seconds ago
Data	{"Alert Distance":91}	json	a few seconds ago
Data	{"Alert Distance":91}	json	a few seconds ago

State

This table shows a list of data points that are reported by this device.

0 Simulations running

Simulation

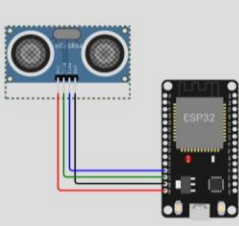
Code

00:59.317

71%

Editing Ultrasonic Distance Sensor

Distance: 87cm



Sending payload: {"Alert Distance":91.00}

Publish ok

Distance in CM: 91

Sending payload: {"Alert Distance":91.00}

When the distance is more than 100cm the alert is not sent to the IBM cloud

The screenshot shows the IBM Watson IoT Platform interface. The page title is "Device Drilldown - ESP". The left sidebar contains a navigation menu with options: Recent Events, State, Device Information, Metadata, Diagnostics, Connection Logs, and Device Actions. The main content area displays "Connection Information" with a "Client Address: 145.40.94.93 Insecure". Below this, the "Recent Events" section shows a table with headers: Event, Value, Format, and Last Received. The table is currently empty, and a message "Waiting for device events..." is displayed. A status box at the bottom right indicates "0 Simulations running".

The screenshot shows the WOKWI mobile app interface. The top bar displays the time as 8:43 PM and the battery level as 98%. The app is in "Simulation" mode, showing a timer at 01:42.500. The main content area displays a simulation of an "Ultrasonic Distance Sensor" connected to an "ESP32" microcontroller. The sensor's distance is shown as 276cm. The bottom section shows the following log output:

```
Sending payload: {"Alert
Distance:":91.00}
Publish ok
Distance in CM: 91
Sending payload: {"Alert
Distance:":91.00}
```