

ASSIGNMENT - 4

| | |
|---------------------|-------------------|
| Date | 19 September 2022 |
| Student Name | DHANUSHYA.J |
| Student Roll Number | 917719D116 |
| Maximum Marks | 2 Marks |

QUESTION:

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

SOLUTION:

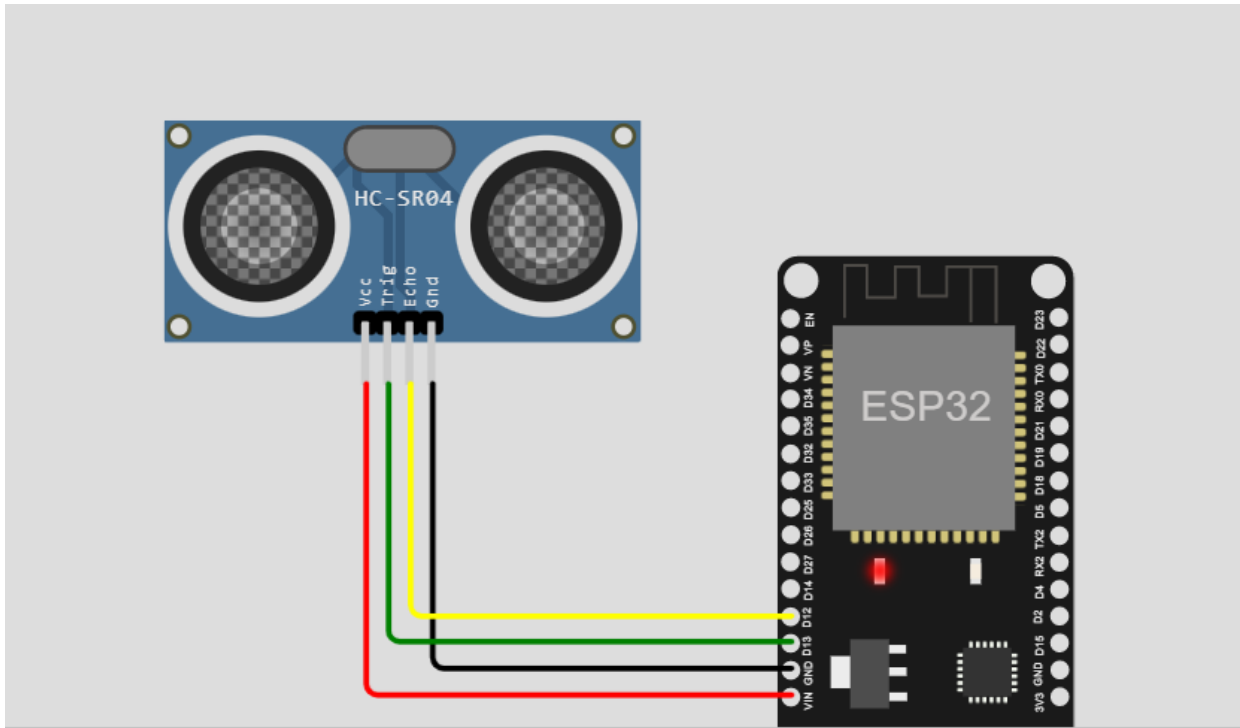
Link: <https://wokwi.com/projects/346784825273746003>

The screenshot displays the Wokwi web interface for a simulation project. On the left, the code editor shows the following code:

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #define TrigPIN 15
4 #define EchoPIN 4
5 #define MINDIST 100
6
7
8
9 void callback(char* topic, byte* payload, unsigned int payloadLength);
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "4qb92" //IBM ORGANIZATION ID
14 #define DEVICE_TYPE "rasperry" //Device type mentioned in ibm watson IOT Platform
15 #define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT Platform
16 #define TOKEN "123456789" //Token
17 String data;
18 float h, t;
19
20
21 //----- Customise the above values -----
22 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
23 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform a
24 char subscribeTopic[] = "iot-2/cmd/command/fmt/string"; // cmd REPRESENT command type AND
25 char authMethod[] = "use-token-auth"; // authentication method
26 char token[] = TOKEN;
27 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
28
29
30 //-----
31 WiFiClient wifiClient; // creating the instance for wifiClient
32 PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client
33
34
35 void setup() // configuring the ESP32
```

The simulation window on the right shows an HC-SR04 ultrasonic sensor connected to an ESP32 microcontroller. The sensor's VCC pin is connected to the ESP32's 5V pin, GND to GND, Trig to D15, and Echo to D4. The console at the bottom shows the following output:

```
Sending payload: {"MESSAGE":"ALERT"}
Publish ok
Sending payload: {"MESSAGE":"ALERT"}
Publish ok
Sending payload: {"MESSAGE":"ALERT"}
Publish ok
Reconnecting client to 4qb92.messaging.internetofthings.ibmcloud.com
```



OUTPUT:

IBM Watson IoT Platform

4qbk92.internetofthings.ibmcloud.com/dashboard/devices/browse

shakinscha@student.tcu.edu ID: 4qbk92

Search by Device ID

Device Simulator

| Device ID | Status | Device Type | Class ID | Date Added | Descriptive Location |
|-----------|-----------|-------------|----------|-------------------|----------------------|
| 12345 | Connected | rasperrypi | Device | 21 Oct 2022 22:13 | |

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 |
|-------|----------------------|--------|-------------------|
| Data | {"MESSAGE": "ALERT"} | json | a few seconds ago |
| Data | {"MESSAGE": "ALERT"} | json | a few seconds ago |
| Data | {"MESSAGE": "ALERT"} | json | a few seconds ago |
| Data | {"MESSAGE": "ALERT"} | json | a few seconds ago |
| Data | {"MESSAGE": "ALERT"} | json | a few seconds ago |

0 Simulations running