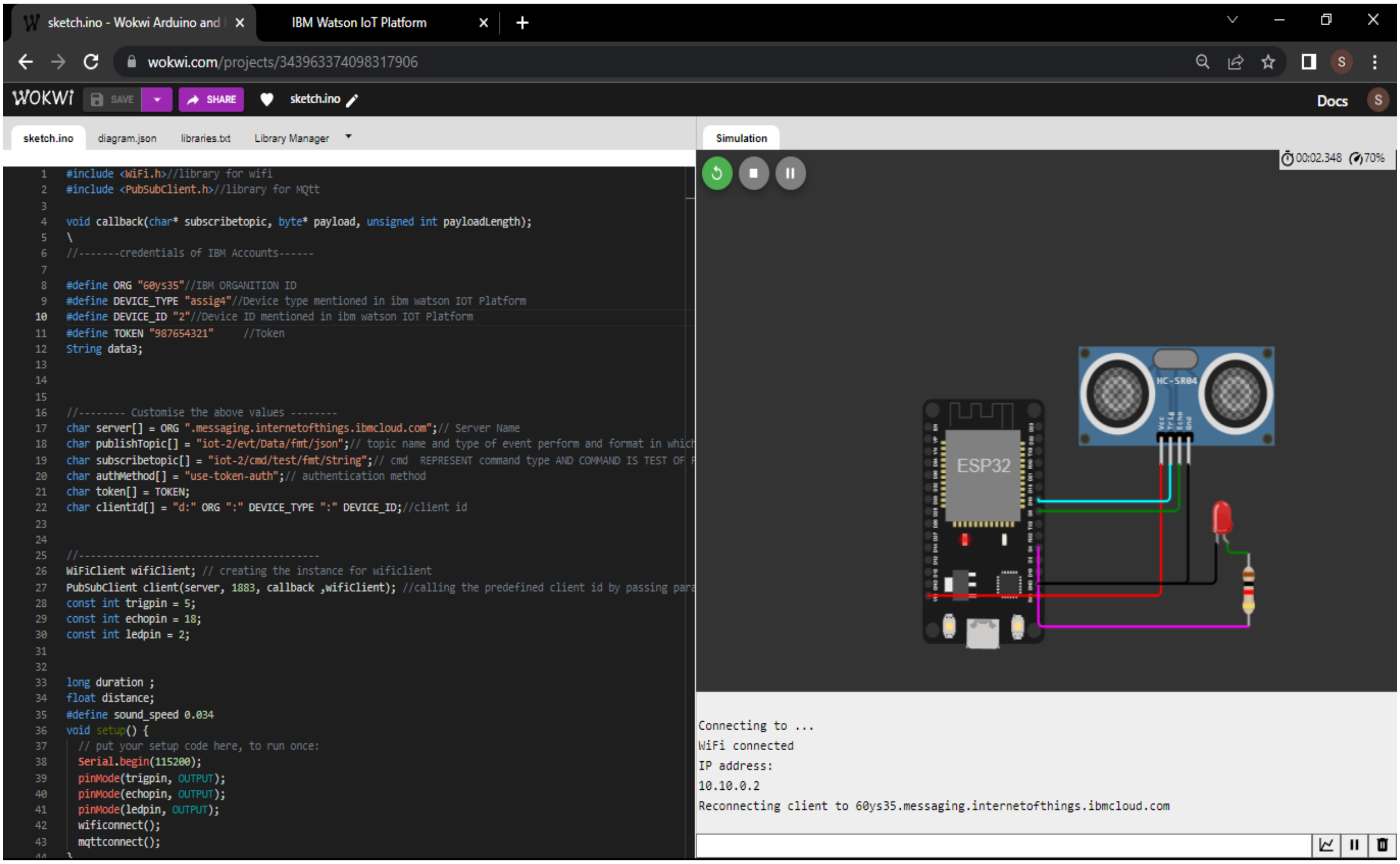
ASSIGNMENT – 04

i. Write code and connection in Wowki for ultrasonic sensor.

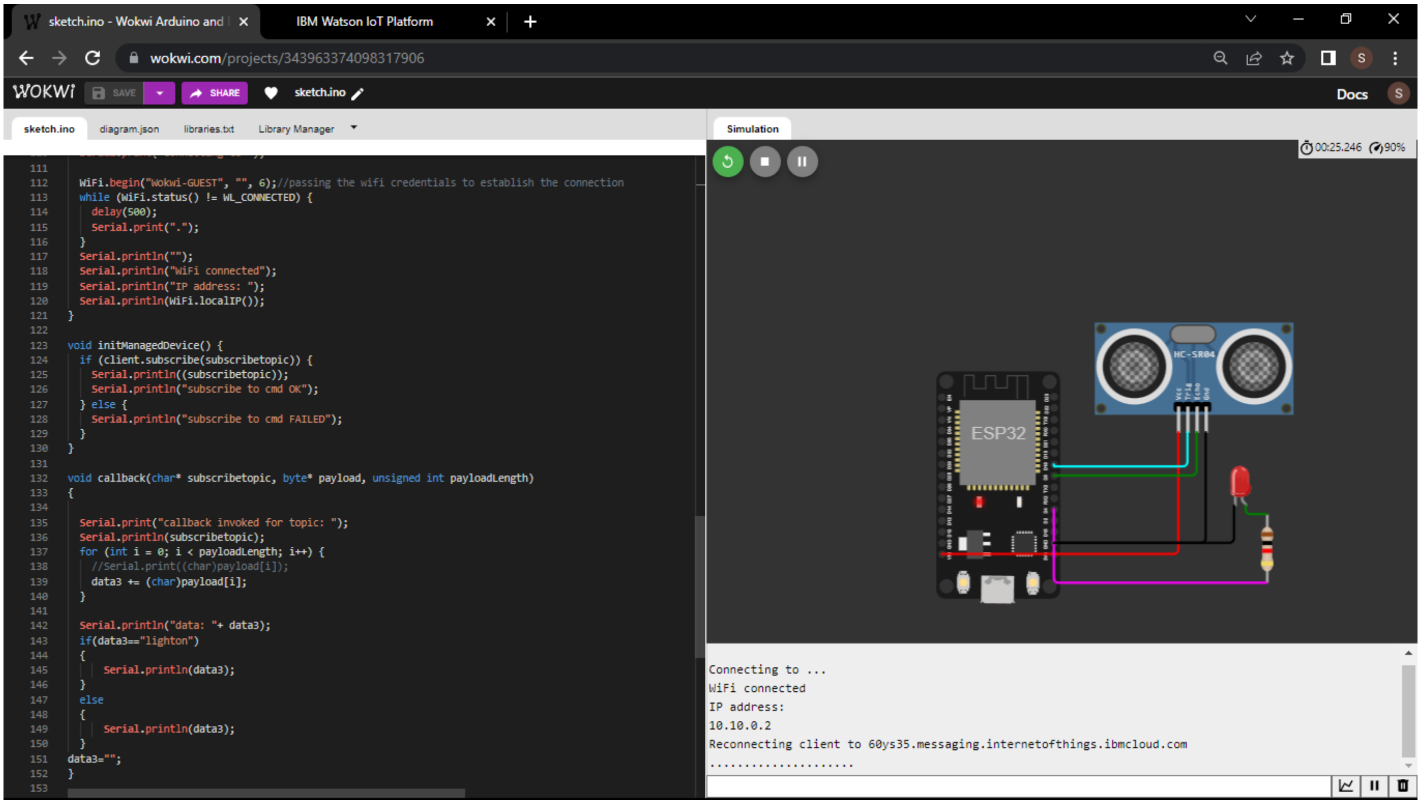
ii. Whenever distance is less than 100 cm send **“Alert”** to IBM cloud and display in device recent events.

Step 1 . Completed to build Circuit and run program.



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

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
\
//-------credentials of IBM Accounts------

#define ORG "60ys35"//IBM ORGANITION ID
#define DEVICE_TYPE "assig4"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "2"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "987654321"     //Token
String data3;



//-------- 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
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF F
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 para
const int trigpin = 5;
const int echopin = 18;
const int ledpin = 2;


long duration ;
float distance;
#define sound_speed 0.034
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, OUTPUT);
  pinMode(ledpin, OUTPUT);
  wificonnect();
  mqttconnect();
}
```

```
Connecting to ...
WiFi connected
IP address:
10.10.0.2
Reconnecting client to 60ys35.messaging.internetofthings.ibmcloud.com
```

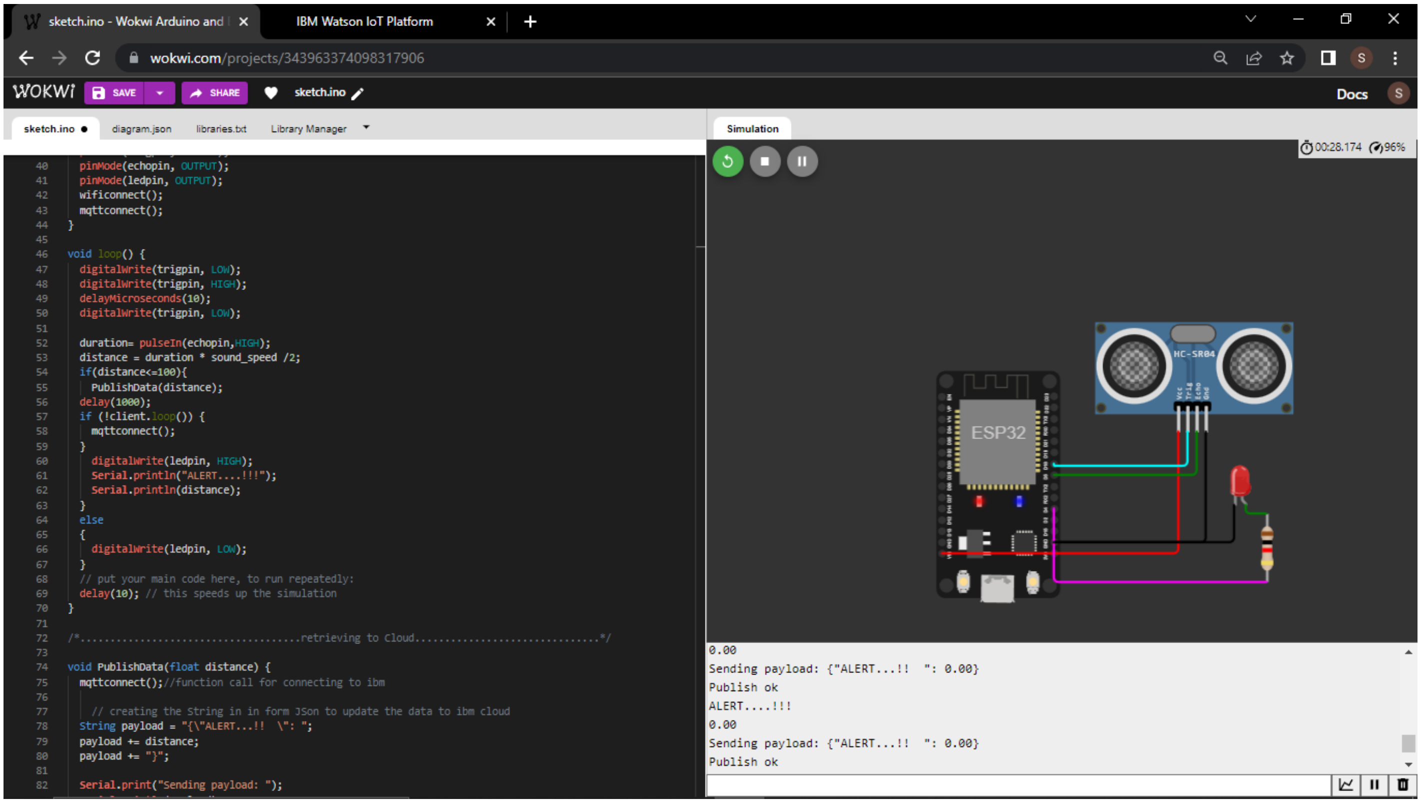


```
  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());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

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="";
}
```

```
Connecting to ...
WiFi connected
IP address:
10.10.0.2
Reconnecting client to 60ys35.messaging.internetofthings.ibmcloud.com
.....................
```

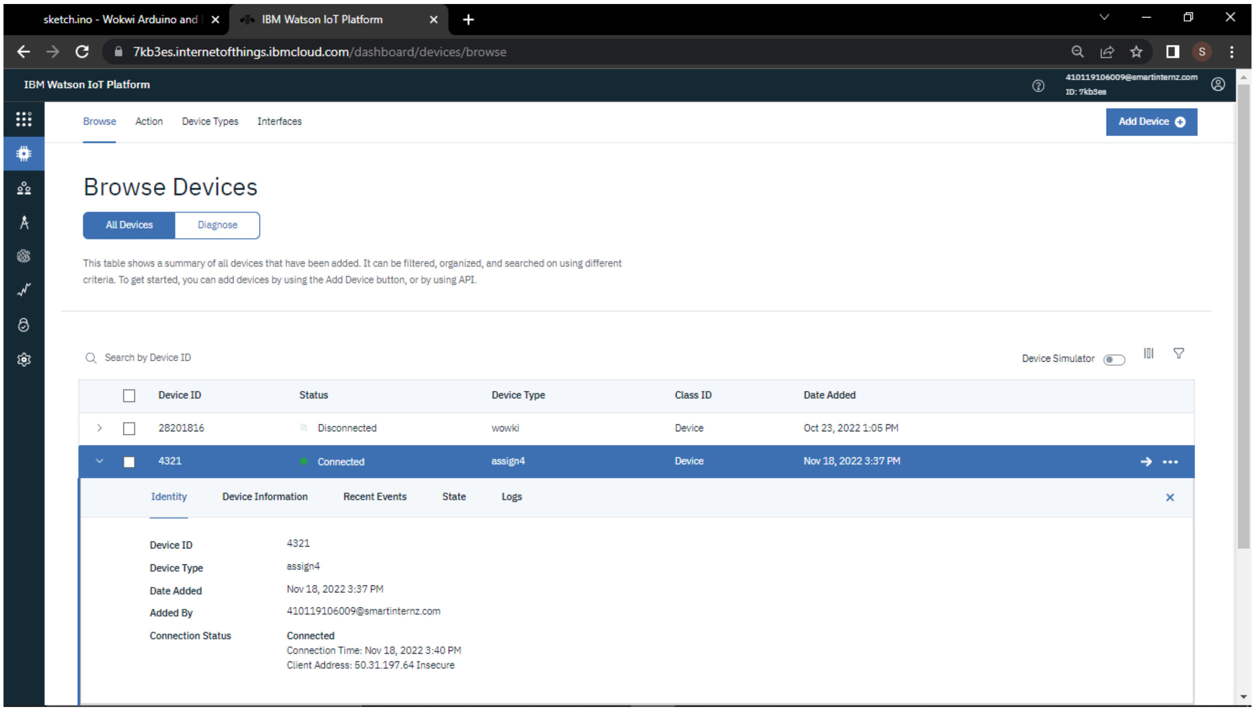
Step 2. Output in WOWKI

(When Distance is below 100 cm send alert to user)



```
  pinMode(echopin, OUTPUT);
  pinMode(ledpin, OUTPUT);
  wificonnect();
  mqttconnect();
}

void loop() {
  digitalWrite(trigpin, LOW);
  digitalWrite(trigpin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin, LOW);

  duration= pulseIn(echopin,HIGH);
  distance = duration * sound_speed /2;
  if(distance<=100){
    PublishData(distance);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
    digitalWrite(ledpin, HIGH);
    Serial.println("ALERT....!!!");
    Serial.println(distance);
  }
  else
  {
    digitalWrite(ledpin, LOW);
  }
  // put your main code here, to run repeatedly:
  delay(10); // this speeds up the simulation
}

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

void PublishData(float distance) {
  mqttconnect();//function call for connecting to ibm

    // creating the String in in form JSon to update the data to ibm cloud
  String payload = "{\"ALERT...!!  \": ";
  payload += distance;
  payload += "}";

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

```
0.00
Sending payload: {"ALERT...!!  ": 0.00}
Publish ok
ALERT....!!!
0.00
Sending payload: {"ALERT...!!  ": 0.00}
Publish ok
```


sketch.ino - Wokwi Arduino and
IBM Watson IoT Platform
7kb3es.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
410119106009@smartinternz.com
ID: 7kb3es
Browse
Action
Device Types
Interfaces
Add Device

Browse Devices

All Devices | Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

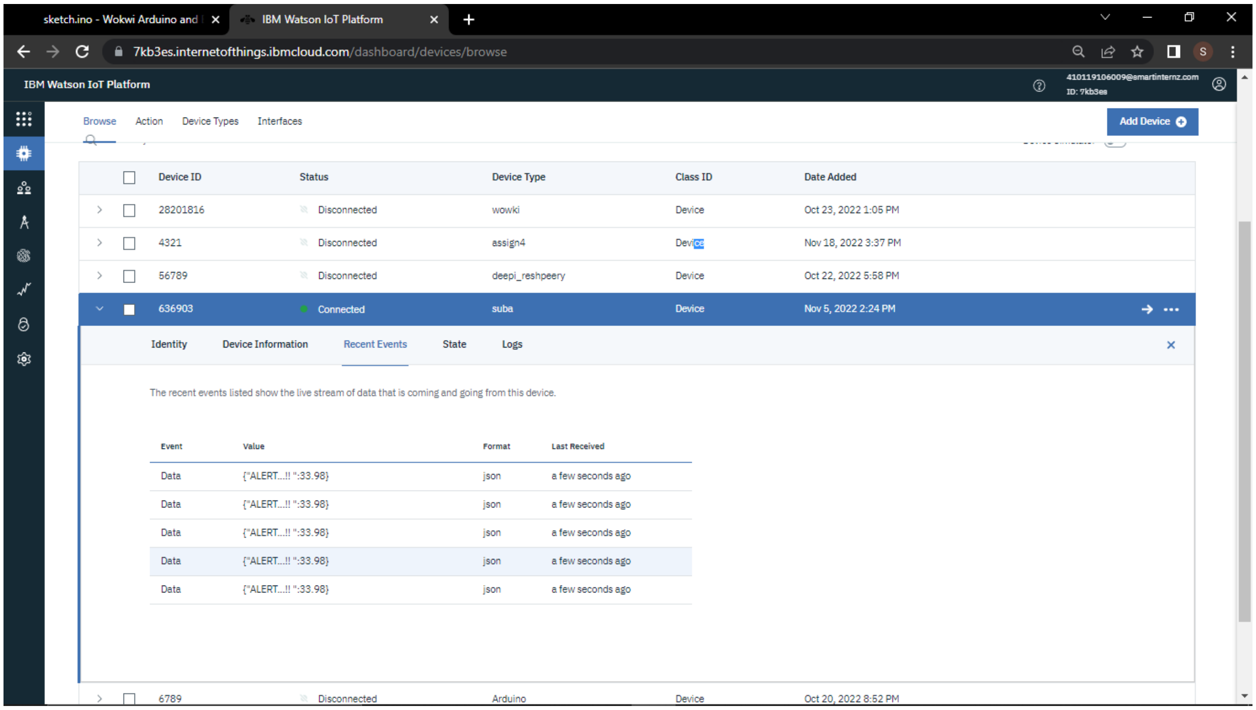
Device Simulator

□	Device ID	Status	Device Type	Class ID	Date Added
□	28201816	Disconnected	wowki	Device	Oct 23, 2022 1:05 PM
□	4321	● Connected	assign4	Device	Nov 18, 2022 3:37 PM

Identity | Device Information | Recent Events | State | Logs

Device ID	4321
Device Type	assign4
Date Added	Nov 18, 2022 3:37 PM
Added By	410119106009@smartinternz.com
Connection Status	Connected Connection Time: Nov 18, 2022 3:40 PM Client Address: 50.31.197.64 Insecure

Step 3.Output in IBM CLOUD (Watson Platform)


sketch.ino - Wokwi Arduino and
IBM Watson IoT Platform
7kb3es.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
410119106009@smartinternz.com
ID: 7kb3es
Browse
Action
Device Types
Interfaces
Add Device

□	Device ID	Status	Device Type	Class ID	Date Added
□	28201816	Disconnected	wowki	Device	Oct 23, 2022 1:05 PM
□	4321	Disconnected	assign4	Device	Nov 18, 2022 3:37 PM
□	56789	Disconnected	deepi_reshpeery	Device	Oct 22, 2022 5:58 PM
□	636903	● Connected	suba	Device	Nov 5, 2022 2:24 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
Data	{"ALERT...!! ":33.98}	json	a few seconds ago
Data	{"ALERT...!! ":33.98}	json	a few seconds ago
Data	{"ALERT...!! ":33.98}	json	a few seconds ago
Data	{"ALERT...!! ":33.98}	json	a few seconds ago
Data	{"ALERT...!! ":33.98}	json	a few seconds ago

□	Device ID	Status	Device Type	Class ID	Date Added
□	6789	Disconnected	Arduino	Device	Oct 20, 2022 8:52 PM

PROGRAM:

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

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
\
//-------credentials of IBM Accounts------

#define ORG "60ys35"//IBM ORGANITION ID
#define DEVICE_TYPE "assign4"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID " 4321"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "0987654321"     //Token
String data3;
//-------- 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/test/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
const int trigpin = 5;
const int echopin = 18;
const int ledpin = 2;


long duration ;
float distance;
#define sound_speed 0.034
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, OUTPUT);
  pinMode(ledpin, OUTPUT);
  wificonnect();
  mqttconnect();
}

void loop() {
  digitalWrite(trigpin, LOW);
  digitalWrite(trigpin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin, LOW);
  duration= pulseIn(echopin,HIGH);
  distance = duration * sound_speed /2;
  if(distance<=100){
    PublishData(distance);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
    digitalWrite(ledpin, HIGH);
    Serial.println("ALERT....!!!");
    Serial.println(distance);
  }
  else
  {
    digitalWrite(ledpin, LOW);
  }
  // put your main code here, to run repeatedly:
  delay(10); // this speeds up the simulation
}

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

void PublishData(float distance) {
  mqttconnect();//function call for connecting to ibm

    // creating the String in in form JSon to update the data to ibm cloud
  String payload = "{\"ALERT...!!  \": ";
  payload += distance;
  payload += "}";

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


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully 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 defination 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="";
}
```