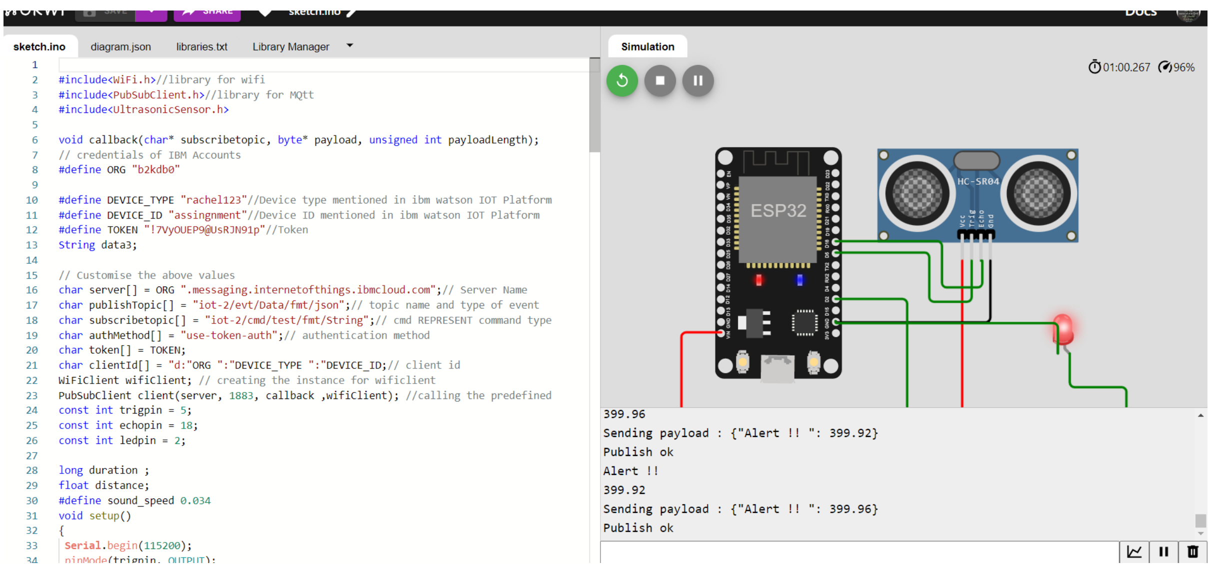
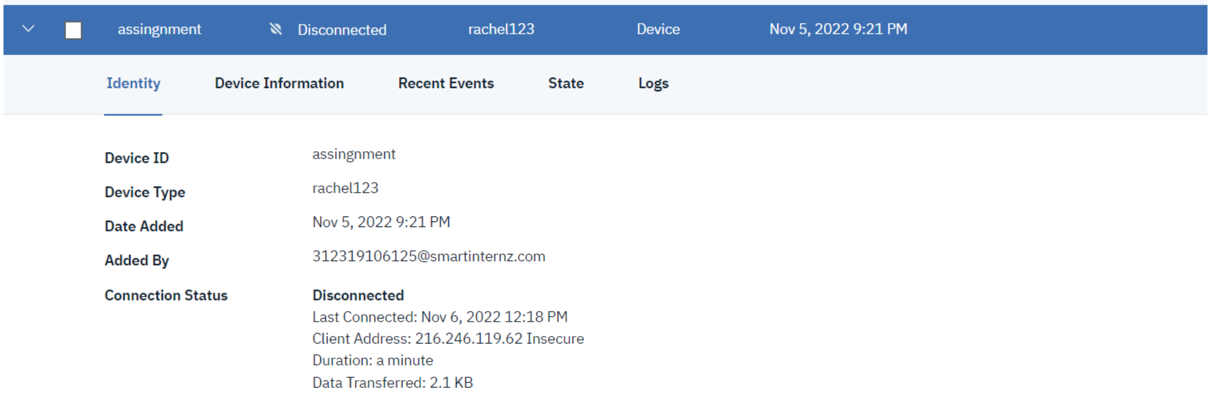


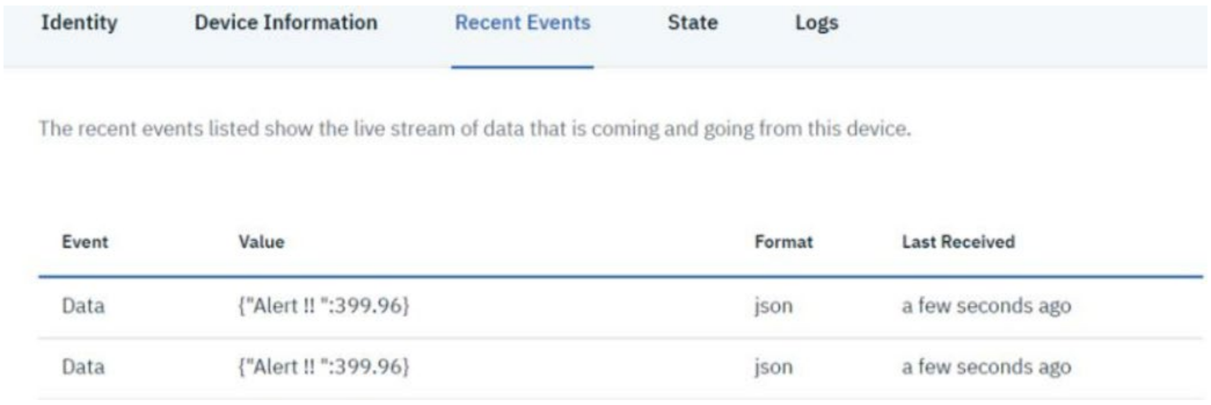
```
#include<WiFi.h>//library for wifi
#include<PubSubClient.h>//library for MQtt
#include<UltrasonicSensor.h>

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
// credentials of IBM Accounts
#define ORG "b2kdb0"

#define DEVICE_TYPE "rachel123"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "assingnment"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "!7VyOUEP9@UsRJN91p"//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
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command type
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
const int trigpin = 5;
const int echopin = 18;
const int ledpin = 2;

long duration ;
float distance;
#define sound_speed 0.034
void setup()
{
 Serial.begin(115200);
 pinMode(trigpin, OUTPUT);
```

```
399.96
Sending payload : {"Alert !! ": 399.92}
Publish ok
Alert !!
399.92
Sending payload : {"Alert !! ": 399.96}
Publish ok
```



assingnment	Disconnected	rachel123	Device	Nov 5, 2022 9:21 PM

Device ID	assingnment
Device Type	rachel123
Date Added	Nov 5, 2022 9:21 PM
Added By	312319106125@smartinternz.com
Connection Status	**Disconnected** Last Connected: Nov 6, 2022 12:18 PM Client Address: 216.246.119.62 Insecure Duration: a minute Data Transferred: 2.1 KB



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 !! ":399.96}	json	a few seconds ago
Data	{"Alert !! ":399.96}	json	a few seconds ago

CODE:

https://wokwi.com/projects/347503543895196243

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

void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
// credentials of IBM Accounts
#define ORG "b2kdb0"

#define DEVICE_TYPE "rachel123"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "assingnment"//Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "!7VyOUEP9@UsRJN91p"//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
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command
type
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
const int trigpin = 5;
const int echopin = 18;
const int ledpin = 2;

long duration ;
float distance;
#define sound_speed 0.034
void setup()
{
 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);
 }
 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
 }
 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
 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="";
}
```

LIBRARY:

```
# Wokwi Library List
# See https://docs.wokwi.com/guides/libraries

PubSubClient
UltrasonicSensor
```

DIAGRAM.JSON:

```
{
  "version": 1,
  "author": "Rachel A",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -15.34, "left": -
205.33, "attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -11.7, "left": -
69.17, "attrs": {} },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": 118.97,
      "left": 68.17,
      "attrs": { "color": "red" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 208.97,
      "left": 36.83,
      "attrs": { "value": "1000" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v182.56", "h-236.44", "v0" ] ],
    [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v51.9", "h-133.45" ] ],
    [ "esp:D18", "ultrasonic1:ECHO", "green", [ "h89.53", "v39.01", "h34", "v-
2" ] ],
    [ "esp:D5", "ultrasonic1:TRIG", "green", [ "h78.86", "v41.41", "h36" ] ],
    [ "esp:D2", "r1:1", "green", [ "h60.2", "v122.97", "h65.33", "v0", "h0",
"v-19.33" ] ],
    [ "led1:A", "r1:2", "green", [ "v28.4", "h47.66", "v27.33", "h-13.33" ] ],
    [ "led1:C", "esp:GND.1", "green", [ "v-25.6", "h-187" ] ]
  ]
}
```