



sketch.ino

diagram.json

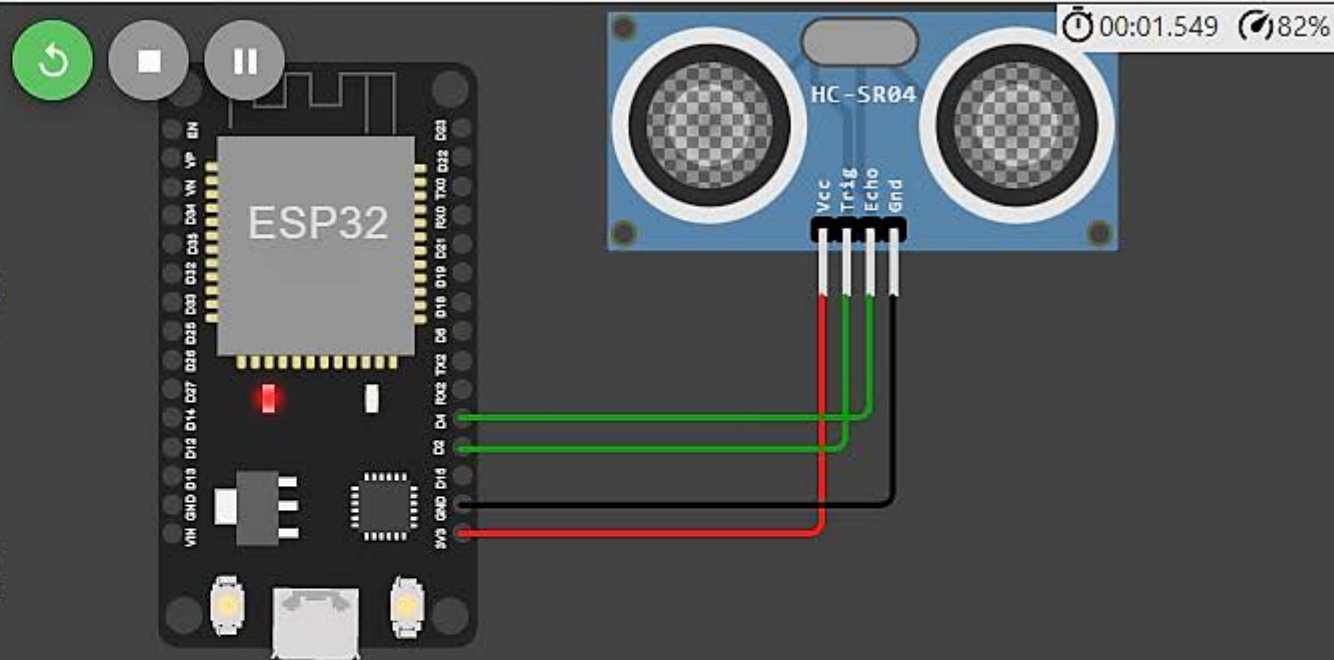
libraries.txt

Library Manager



Simulation

```
1 #include <WiFi.h>//library for wifi
2 #include <PubSubClient.h>//library for MQTT
3
4 int trigpin=2;
5 int echopin=4;
6 String data3;
7
8 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
9 #define ORG "z1frj0"//IBM ORGANITION ID
10 #define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
11 #define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
12 #define TOKEN "12345678" //Token
13
14 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
15 char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of even
16 char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT comma
17 char authMethod[] = "use-token-auth";// authentication method
18 char token[] = TOKEN;
19 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
20
21 WiFiClient wifiClient; // creating the instance for wificlient
22 PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefi
23
24
25 void setup() {
26   Serial.begin(9600);
27   Serial.println("Hello, ESP32!");
28   pinMode(2, OUTPUT);
29   pinMode(4, INPUT);
30   delay(10);
31 }
```



Hello, ESP32!

Connecting to ..

WiFi connected

IP address:

10.10.0.2

Reconnecting client to i3869j.messaging.internetofthings.ibmcloud.com



sketch.ino

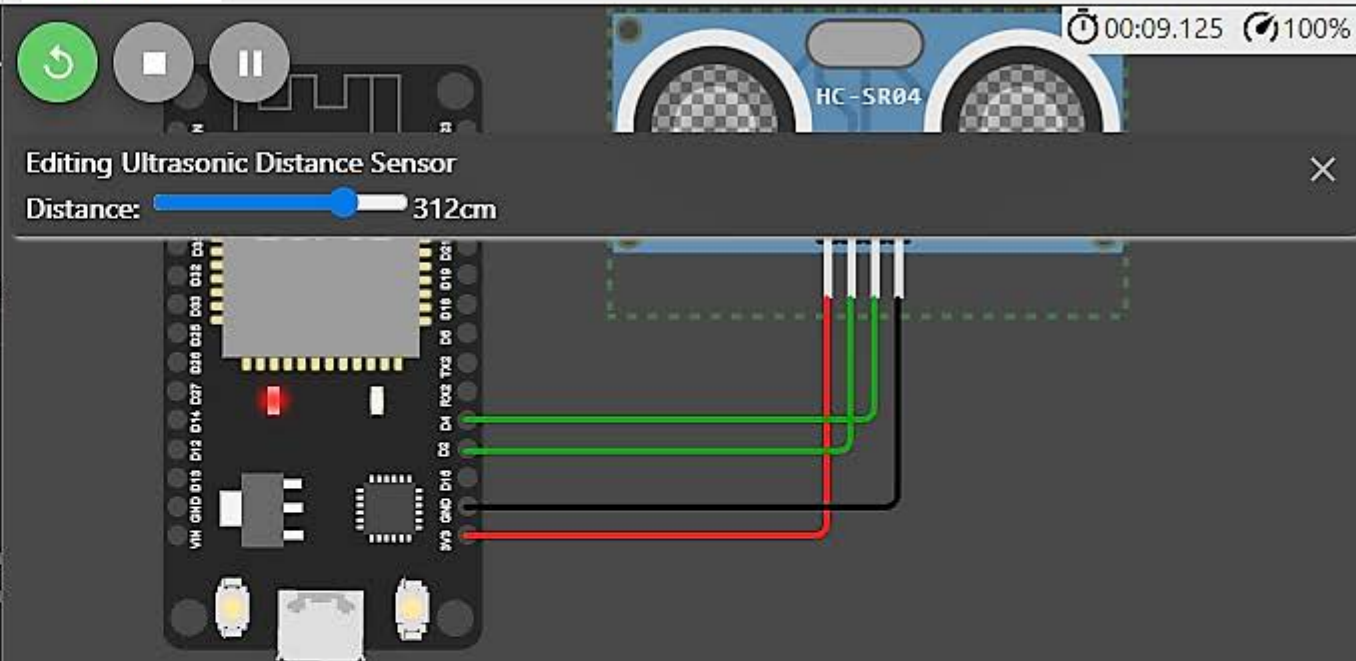
diagram.json

libraries.txt

Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4 int trigpin=2;
5 int echopin=4;
6 String data3;
7
8 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
9 #define ORG "z1frj0" //IBM ORGANITION ID
10 #define DEVICE_TYPE "abcd" //Device type mentioned in ibm watson IOT Platform
11 #define DEVICE_ID "1234" //Device ID mentioned in ibm watson IOT Platform
12 #define TOKEN "12345678" //Token
13
14 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
15 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of even
16 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT comma
17 char authMethod[] = "use-token-auth"; // authentication method
18 char token[] = TOKEN;
19 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
20
21 WiFiClient wificlient; // creating the instance for wificlient
22 PubSubClient client(server, 1883, callback ,wificlient); //calling the predefi
23
24
25 void setup() {
26   Serial.begin(9600);
27   Serial.println("Hello, ESP32!");
28   pinMode(2, OUTPUT);
29   pinMode(4, INPUT);
30   delay(10);
```

Simulation



iot-2/cmd/command/fmt/String

subscribe to cmd OK

Distance: 208.77 cms

Distance: 208.77 cms

Distance: 208.77 cms

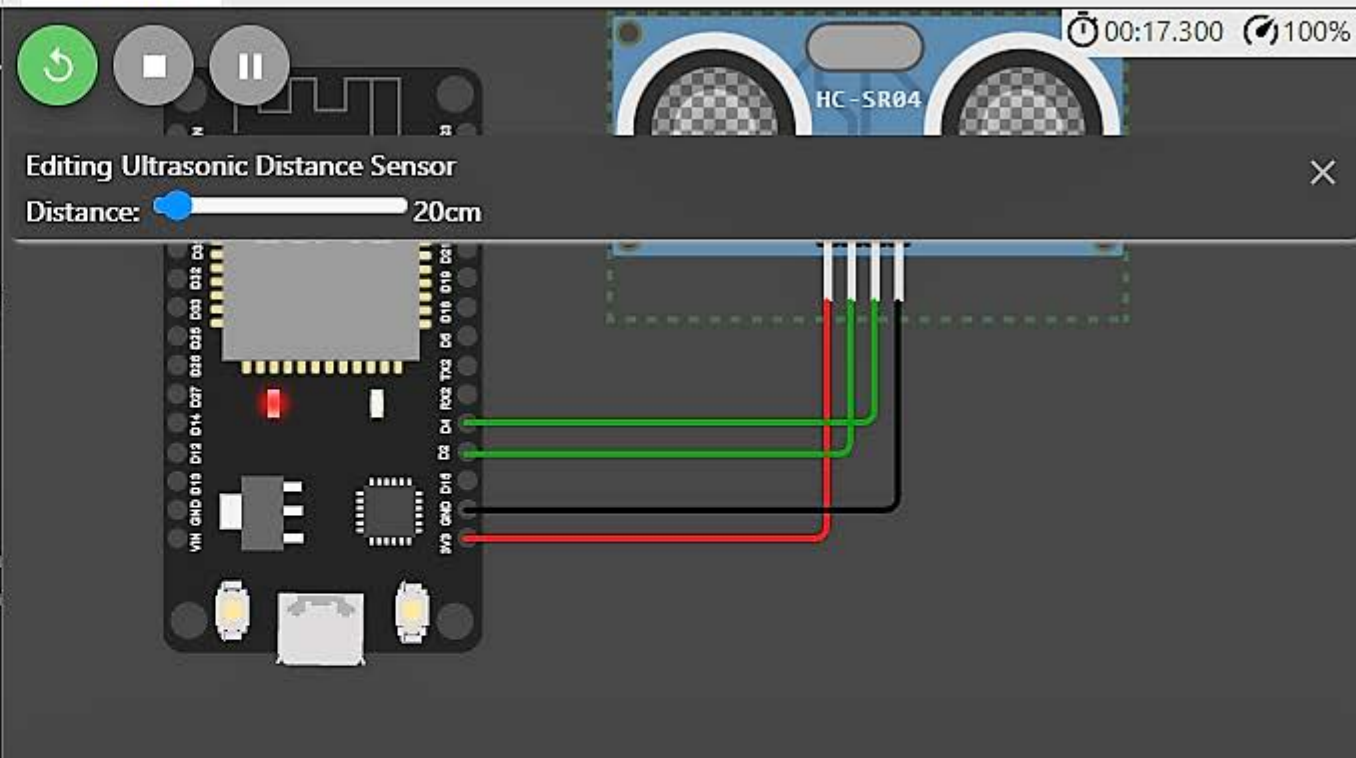
Distance: 314.70 cms



sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4 int trigpin=2;
5 int echopin=4;
6 String data3;
7
8 void callback(char* subscribtopic, byte* payload, unsigned int payloadLength)
9 #define ORG "z1frj0" //IBM ORGANITION ID
10 #define DEVICE_TYPE "abcd" //Device type mentioned in ibm watson IOT Platform
11 #define DEVICE_ID "1234" //Device ID mentioned in ibm watson IOT Platform
12 #define TOKEN "12345678" //Token
13
14 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
15 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of even
16 char subscribtopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT comma
17 char authMethod[] = "use-token-auth"; // authentication method
18 char token[] = TOKEN;
19 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
20
21 WiFiClient wificlient; // creating the instance for wificlient
22 PubSubClient client(server, 1883, callback ,wificlient); //calling the predefi
23
24
25 void setup() {
26   Serial.begin(9600);
27   Serial.println("Hello, ESP32!");
28   pinMode(2, OUTPUT);
29   pinMode(4, INPUT);
30   delay(10);
```

Simulation



```
Distance: 314.70 cms
Distance: 321.77 cms
Distance: 403.49 cms
Distance: 20.15 cms
Sending payload: {"Distance":20.15,"Message":"alert"}
Publish ok
Distance: 20.17 cms
```



sketch.ino

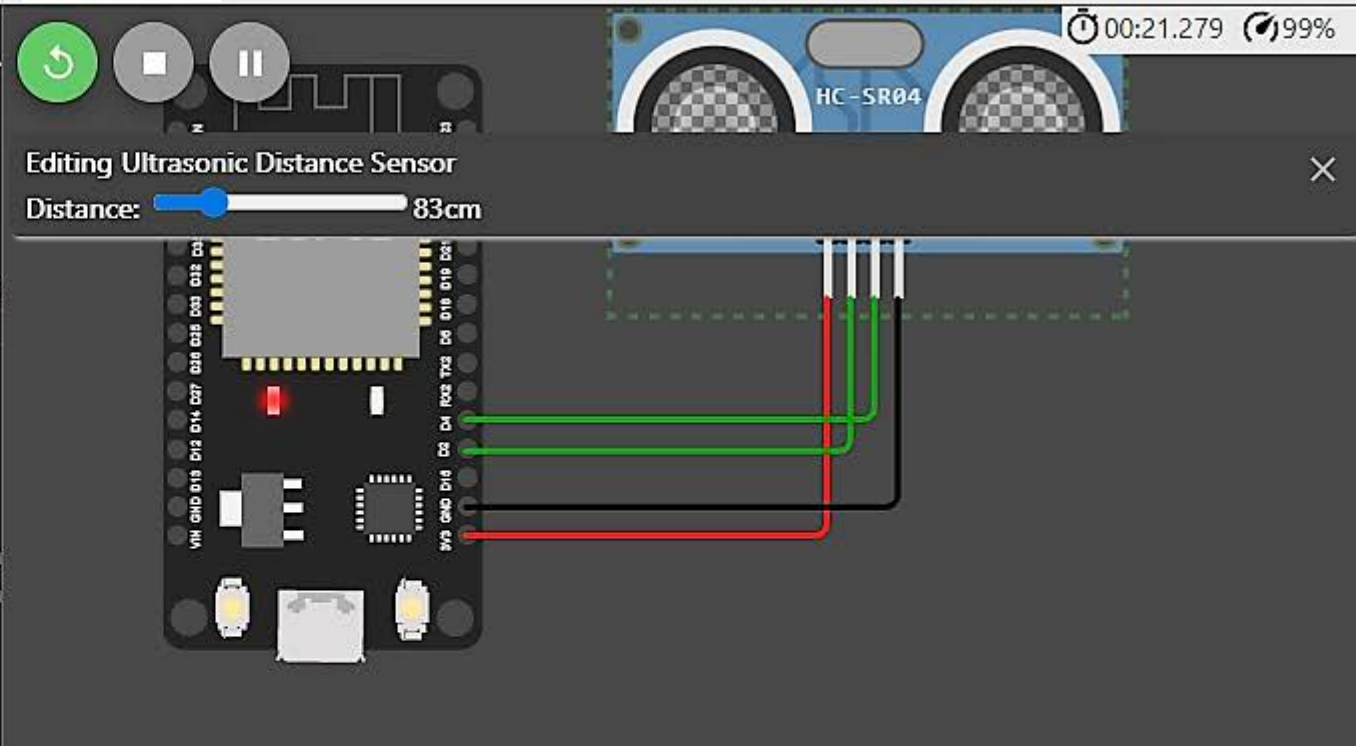
diagram.json

libraries.txt

Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4 int trigpin=2;
5 int echopin=4;
6 String data3;
7
8 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
9 #define ORG "z1frj0" //IBM ORGANITION ID
10 #define DEVICE_TYPE "abcd" //Device type mentioned in ibm watson IOT Platform
11 #define DEVICE_ID "1234" //Device ID mentioned in ibm watson IOT Platform
12 #define TOKEN "12345678" //Token
13
14 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
15 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of even
16 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT comma
17 char authMethod[] = "use-token-auth"; // authentication method
18 char token[] = TOKEN;
19 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
20
21 WiFiClient wifiClient; // creating the instance for wificlient
22 PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefi
23
24
25 void setup() {
26   Serial.begin(9600);
27   Serial.println("Hello, ESP32!");
28   pinMode(2, OUTPUT);
29   pinMode(4, INPUT);
30   delay(10);
```

Simulation



Distance: 20.17 cms

Sending payload: {"Distance":20.17,"Message":"alert"}

Publish ok

Distance: 20.12 cms

Sending payload: {"Distance":20.12,"Message":"alert"}

Publish ok

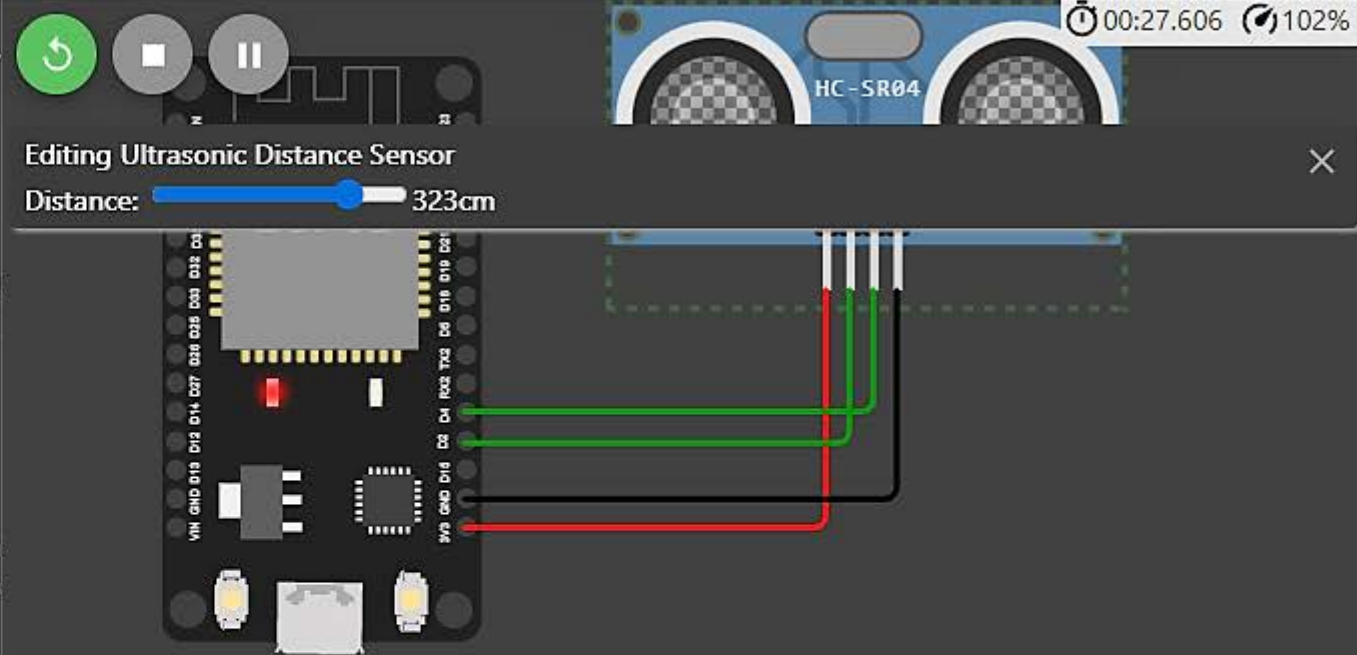
Distance: 83.73 cms



sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4 int trigpin=2;
5 int echopin=4;
6 String data3;
7
8 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
9 #define ORG "z1frj0" //IBM ORGANITION ID
10 #define DEVICE_TYPE "abcd" //Device type mentioned in ibm watson IOT Platform
11 #define DEVICE_ID "1234" //Device ID mentioned in ibm watson IOT Platform
12 #define TOKEN "12345678" //Token
13
14 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
15 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of even
16 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT comma
17 char authMethod[] = "use-token-auth"; // authentication method
18 char token[] = TOKEN;
19 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
20
21 WiFiClient wificlient; // creating the instance for wificlient
22 PubSubClient client(server, 1883, callback ,wificlient); //calling the predefi
23
24
25 void setup() {
26   Serial.begin(9600);
27   Serial.println("Hello, ESP32!");
28   pinMode(2, OUTPUT);
29   pinMode(4, INPUT);
30   delay(10);
```

Simulation



Publish ok

Distance: 83.73 cms

Sending payload: {"Distance":83.73,"Message":"alert"}

Publish ok

Distance: 225.97 cms

Distance: 225.93 cms

Distance: 325.82 cms

sketch.ino

diagram.json

libraries.txt

Library Manager

```

1  #include <WiFi.h>//library for wifi
2  #include <PubSubClient.h>//library for MQTT
3
4  int trigpin=2;
5  int echopin=4;
6  String data3;
7
8  void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
9  #define ORG "z1frj0"//IBM ORGANITION ID
10 #define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
11 #define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
12 #define TOKEN "12345678" //Token
13
14 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
15 char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of even
16 char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT comma
17 char authMethod[] = "use-token-auth";// authentication method
18 char token[] = TOKEN;
19 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
20
21 WiFiClient wificlient; // creating the instance for wificlient
22 PubSubClient client(server, 1883, callback ,wificlient); //calling the predefi
23
24
25 void setup() {
26   Serial.begin(9600);
27   Serial.println("Hello, ESP32!");
28   pinMode(2, OUTPUT);
29   pinMode(4, INPUT);
30   delay(10);

```

Simulation

00:33.450

100%

HC-SR04

Editing Ultrasonic Distance Sensor

Distance: 16cm

Distance: 225.93 cms

Distance: 325.82 cms

Distance: 325.82 cms

Distance: 16.10 cms

Sending payload: {"Distance":16.10,"Message":"alert"}

Publish ok

Distance: 16.12 cms

[Browse](#) [Action](#) [Device Types](#) [Interfaces](#)[Add Device](#) 

Device ID

Status

Device Type

Class ID

Date Added



1234

Connected

abcd

Device

24 Oct 2022 20:22

[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	{"Distance":16.1,"Message":"alert"}	json	a few seconds ago
Data	{"Distance":83.73,"Message":"alert"}	json	a few seconds ago
Data	{"Distance":20.12,"Message":"alert"}	json	a few seconds ago
Data	{"Distance":20.17,"Message":"alert"}	json	a few seconds ago
Data	{"Distance":20.15,"Message":"alert"}	json	a few seconds ago

Link for wokwi project:

<https://wokwi.com/projects/346399595888116308>