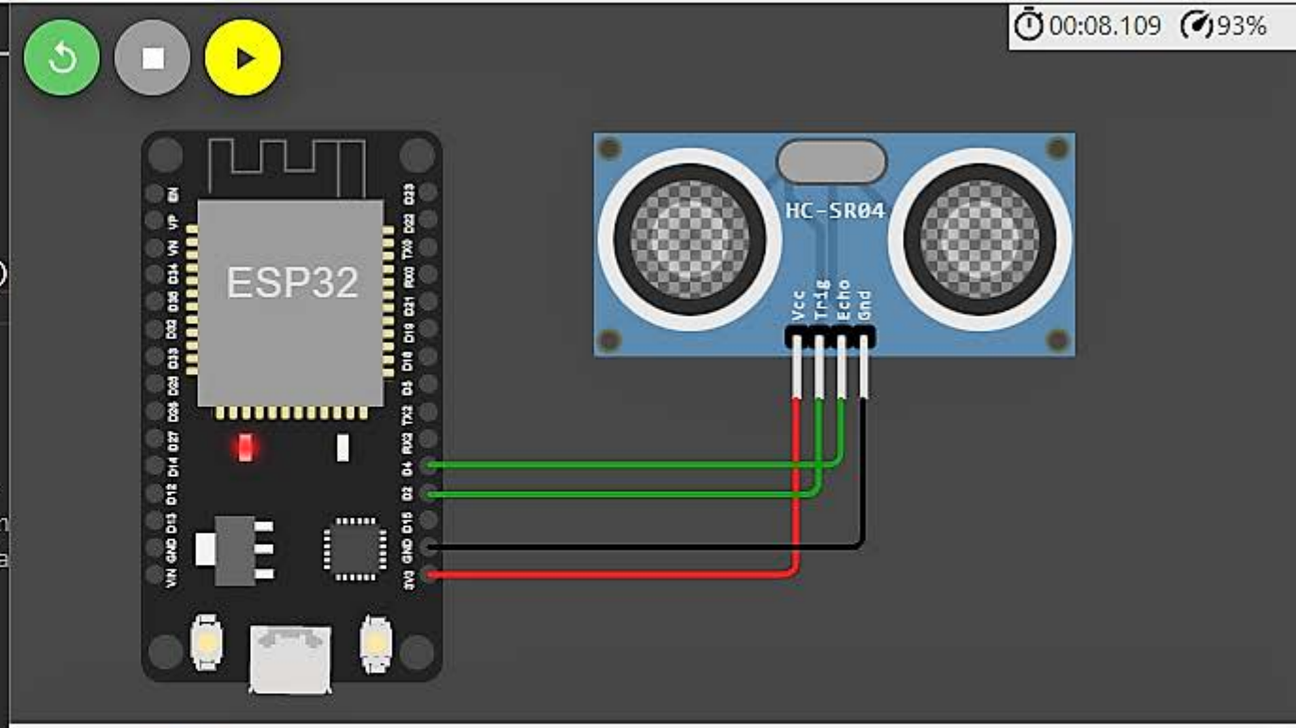
WOKWI SAVE SHARE Docs E

sketch.ino | diagram.json ● | libraries.txt | Library Manager

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

int trigpin=2;
int echopin=4;
String data3;

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
#define ORG "lfq5we"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of even
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT comma
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 predefi


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Hello, ESP32!");
  pinMode(2, OUTPUT);
  pinMode(4, INPUT);
```

Simulation




```
Hello, ESP32!

Connecting to .....
WiFi connected
IP address:
10.10.0.2
Reconnecting client to i3869j.messaging.internetofthings.ibmcloud.com
```

WOKWI SAVE SHARE Docs E

sketch.ino | diagram.json ● | libraries.txt | Library Manager

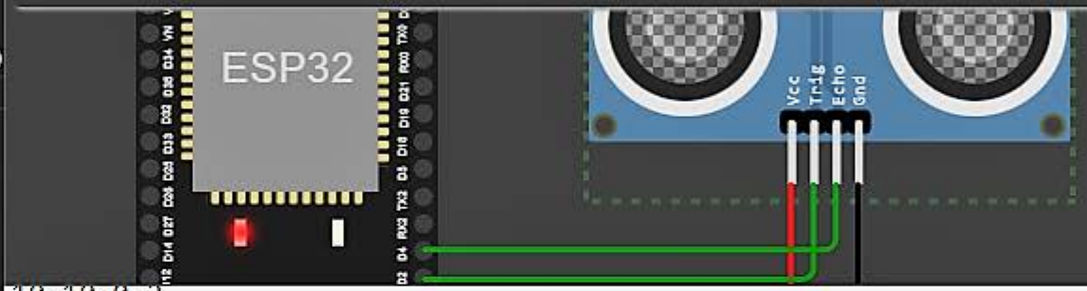
```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt

int trigpin=2;
int echopin=4;
String data3;

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
#define ORG "lfq5we"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of even
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT comma
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 predefi


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Hello, ESP32!");
  pinMode(2, OUTPUT);
  pinMode(4, INPUT);
```

Simulation

00:08.109 93%

Editing Ultrasonic Distance Sensor

Distance: 351cm




```
10.10.0.2
Reconnecting client to i3869j.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK

Distance: 354.06 cms
Sending payload: {"Distance":354.06,"Message":"alert"}
Publish ok
Distance: 354.13 cms
Sending payload: {"Distance":354.13,"Message":"alert"}
Publish ok
Distance: 354.10 cms
```

WOKWI SAVE SHARE Docs E

sketch.ino | diagram.json ● | libraries.txt | Library Manager

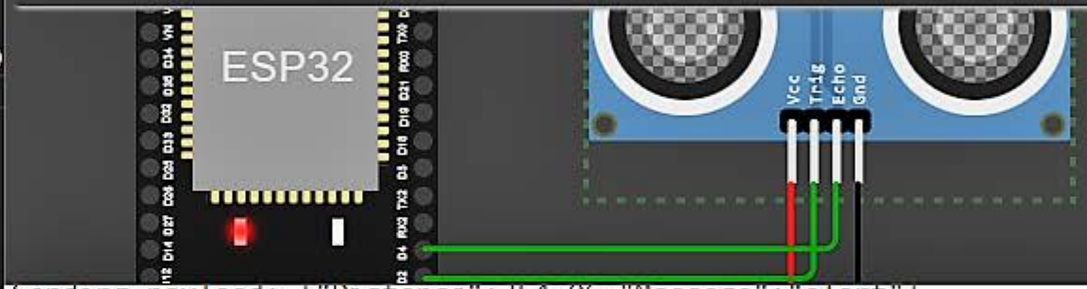
```ino
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt

int trigpin=2;
int echopin=4;
String data3;

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
#define ORG "lfq5we"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of even
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT comma
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 predefi


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Hello, ESP32!");
  pinMode(2, OUTPUT);
  pinMode(4, INPUT);
```

Simulation

00:12.338 103%

Editing Ultrasonic Distance Sensor

Distance: 217cm




```
Sending payload: {"Distance":354.06,"Message":"alert"}
Publish ok
Distance: 354.13 cms
Sending payload: {"Distance":354.13,"Message":"alert"}
Publish ok
Distance: 354.10 cms
Sending payload: {"Distance":354.10,"Message":"alert"}
Publish ok
Distance: 218.90 cms
Sending payload: {"Distance":218.90,"Message":"alert"}
Publish failed
Distance: 218.89 cms
```

WOKWI SAVE SHARE Docs E

sketch.ino | diagram.json ● | libraries.txt | Library Manager

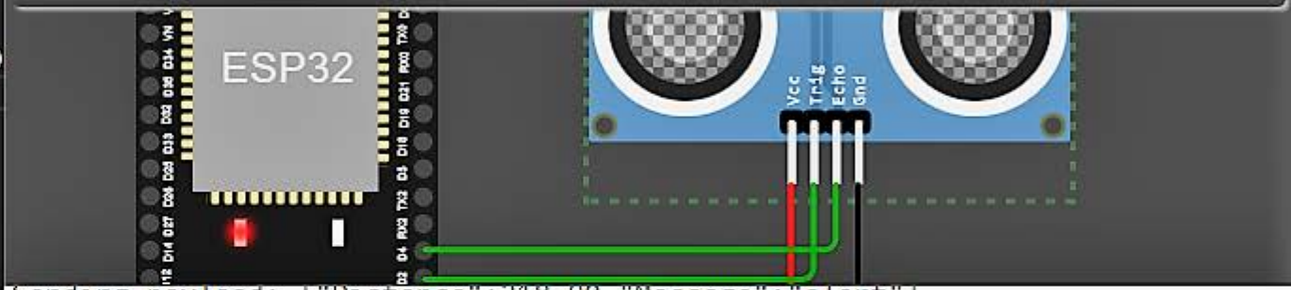
```ino
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt

int trigpin=2;
int echopin=4;
String data3;

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
#define ORG "lfq5we"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of even
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT comma
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 predefi


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Hello, ESP32!");
  pinMode(2, OUTPUT);
  pinMode(4, INPUT);
```

Simulation

00:19.098 101%

Editing Ultrasonic Distance Sensor

Distance: 62cm




```
Sending payload: {"Distance":218.90,"Message":"alert"}
Publish failed
Distance: 218.89 cms
Reconnecting client to i3869j.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK

Sending payload: {"Distance":218.89,"Message":"alert"}
Publish ok
Distance: 94.79 cms
Distance: 62.49 cms
Distance: 62.49 cms
```

WOKWI SAVE SHARE Docs E

sketch.ino | diagram.json ● | libraries.txt | Library Manager

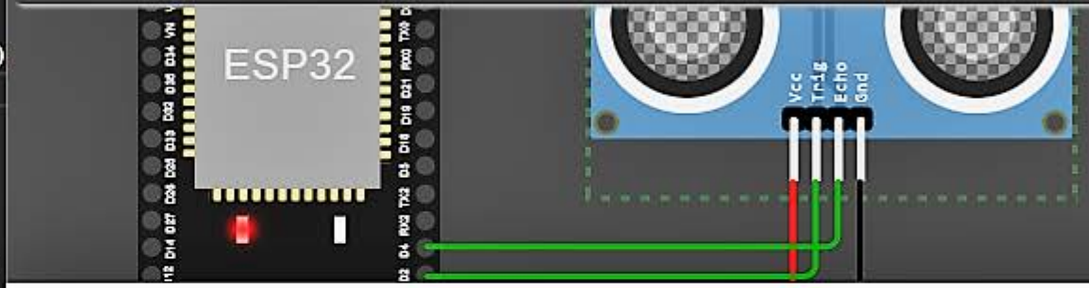
```ino
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt

int trigpin=2;
int echopin=4;
String data3;

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
#define ORG "lfq5we"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of even
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT comma
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 predefi


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Hello, ESP32!");
  pinMode(2, OUTPUT);
  pinMode(4, INPUT);
```

Simulation

00:27.173 99%

Editing Ultrasonic Distance Sensor

Distance: 266cm




```
Sending payload: {"Distance":218.89,"Message":"alert"}
Publish ok
Distance: 94.79 cms
Distance: 62.49 cms
Distance: 62.49 cms
Distance: 62.49 cms
Distance: 62.49 cms
Distance: 268.29 cms
Sending payload: {"Distance":268.29,"Message":"alert"}
Publish ok
Distance: 268.74 cms
```

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":268.74,"Message":"alert"}	json	a few seconds ago
Data	{"Distance":268.29,"Message":"alert"}	json	a few seconds ago
Data	{"Distance":218.89,"Message":"alert"}	json	a few seconds ago
Data	{"Distance":218.9,"Message":"alert"}	json	a few seconds ago
Data	{"Distance":354.1,"Message":"alert"}	json	a few seconds ago

https://wokwi.com/projects/346399595888116308

Link for wokwi project.