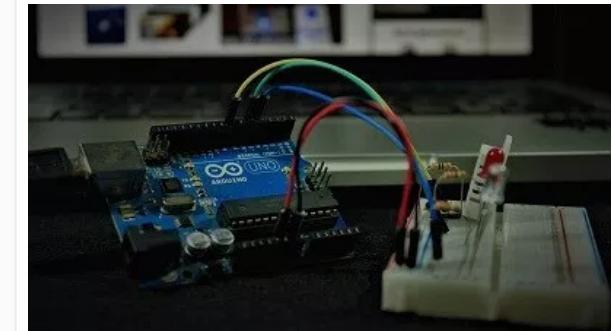


# Alexa (Echo) with ESP32 and ESP8266 – Voice Controlled Relay

In this project, you're going to learn how to control the ESP8266 or the ESP32 with voice commands using Alexa (Amazon Echo Dot). As an example, we'll control two 12V lamps connected to a relay module. We'll also add two 433 MHz RF wall panel switches to physically control the lamps.

**Note:** this tutorial is compatible with all Echo Dot generations and with the latest fauxmoESP library (3.1.0). It works with ESP32 and ESP8266.

## Watch the Project Video Demonstration



[Arduino Step-by-Step Projects »](#)  
Build 25 cool Arduino projects with our course even with no prior experience!



[ESP32-CAM Video Streaming Web Server \(works with Home Assistant\)](#)



We recommend the following tutorials as a reference:

- [Getting Started with ESP8266 Wi-Fi Transceiver](#)
- [Getting Started with ESP32 Dev Module](#)
- [Decode and Send 433 MHz RF Signals with Arduino](#)
- [Guide for Relay Module with Arduino](#)

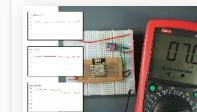
## Project Overview

This project works both with ESP8266 and ESP32. We provide instructions for both development boards. Before getting straight to the project, read this section to

[Build an ESP8266 Web Server – Code and Schematics](#)



[ESP32 Capacitive Touch Sensor Pins with Arduino IDE](#)



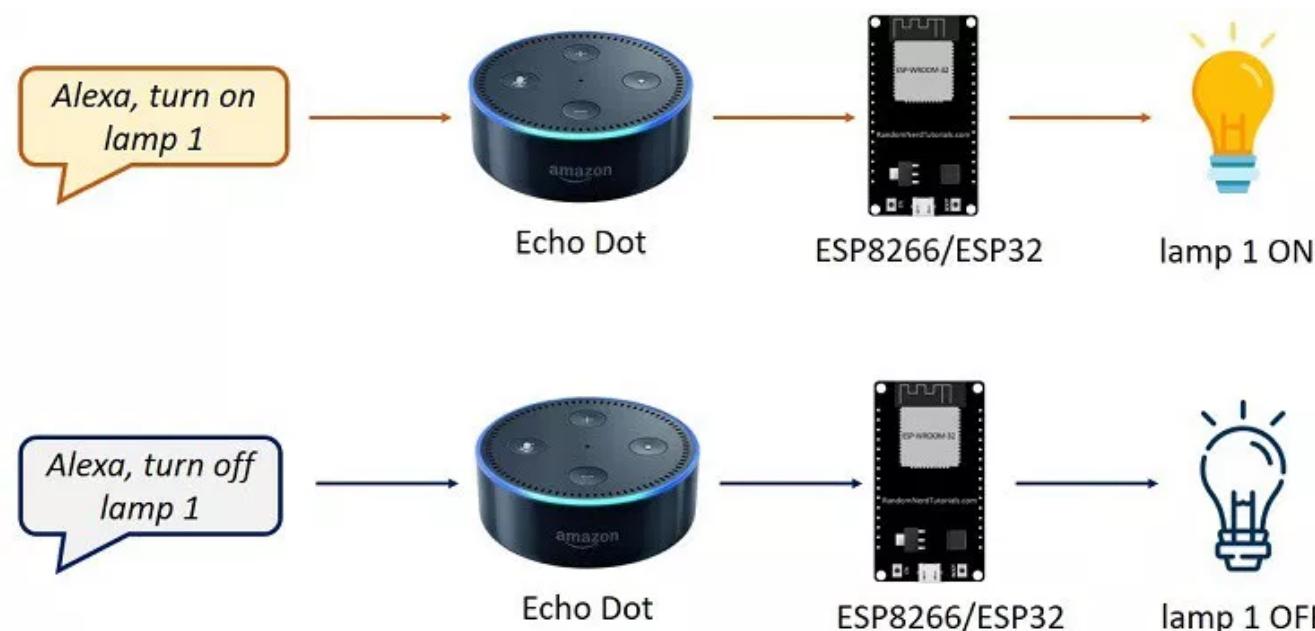
[Low Power Weather Station Datalogger using ESP8266 and BME280 with MicroPython](#)



[Visit Maker Advisor – Tools and Gear for makers, hobbyists and DIYers »](#)

# Control Lamps using Alexa

By the end of this project you'll be able to control two lamps (lamp 1 and lamp 2) with voices commands using Alexa. The figure below shows a high-level overview on how the project works to control lamp 1 – it works similarly for lamp 2.

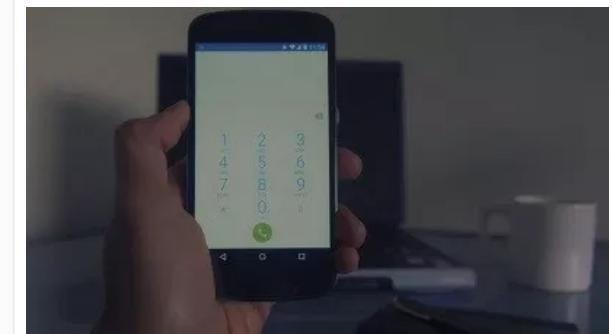


Alexa will respond to the following commands:

- “Alexa, turn on lamp 1”
- “Alexa, turn off lamp 1”
- “Alexa, turn on lamp 2”
- “Alexa, turn on lamp 2”
- “Alexa, turn on lamps” turns on both lamps



**Home Automation using ESP8266 eBook and video course »** Build IoT and home automation projects.

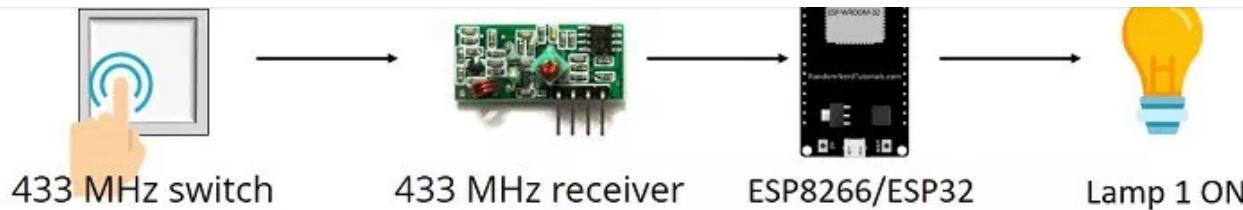


**Build a Home Automation System from Scratch »** With Raspberry Pi, ESP8266, Arduino, and Node-RED.

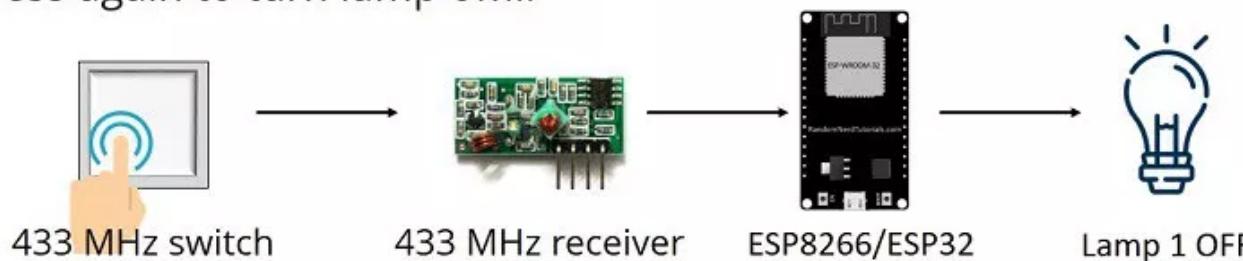
When you say something like "*Alexa, turn on lamp 1*", the ESP8266 or ESP32 will trigger a relay to turn on lamp 1. When you say something like "*Alexa, turn off lamp 1*", the ESP8266 or ESP32 will send a signal to the relay to turn off the lamp. This works similarly for lamp 2.

## Control Lamps using 433 MHz Wall Switches

In this project, we'll also add two 433 MHz wall switches to physically control the lamps. You'll have a switch for each lamp. The switch changes the lamp's state to the opposite of its current state. For example, if the lamp is off, press the wall switch to turn it on. To turn it off, you just need to press the switch again. Take a look at the figure below that illustrates how it works.



Press again to turn lamp off...



## Parts Required

Here's a complete list of the parts required for this project (click the links below to find the best price at [Maker Advisor](#)):



- ESP Board (you can use either ESP32 or ESP8266):
  - [ESP8266](#) – read [Best ESP8266 Wi-Fi Development Boards](#)
  - [ESP32](#) – we use the [ESP32 DOIT DEVKIT V1 Board – 36 GPIOs](#) (read [ESP32 development boards comparison](#))
- Alexa – Echo, Echo Show or Echo Dot (**read the next section for more details**)
- [433 MHz RF Wall Panel Switch](#)
- [433 MHz transmitter/receiver](#)
- [12V 2A power adaptor](#)
- [Step-down buck converter](#)
- [Relay module](#)
- [12V lamp](#)
- [12V lamp holder](#)

■ [jumper wires](#)

You can use the preceding links or go directly to [MakerAdvisor.com/tools](#) to find all the parts for your projects at the best price!

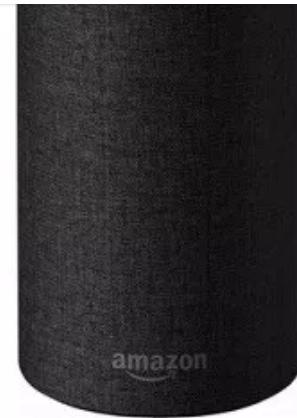


## How to Buy An Amazon Echo

You can use the links below to buy an Amazon Echo. There are several models available – all of them are compatible with this project.



**Echo Dot  
(2<sup>nd</sup> Generation)**



**Echo  
(2<sup>nd</sup> Generation)**



**Echo Show**

Buying an Amazon Echo through Amazon is not possible for all countries. We provide links for Amazon in UK, USA, and Germany. If the Amazon Echo does not ship to your country through Amazon, you can get one from eBay (available worldwide).

### **Echo Dot (2nd Generation)**

- [United States – Amazon.com](#)
- [Available Worldwide – eBay.com](#)
- [United Kingdom – Amazon.co.uk](#)
- [Germany – Amazon.de](#)

### **Echo (2nd Generation)**

- [United Kingdom - Amazon.co.uk](#)
- [Germany - Amazon.de](#)

## Echo Show

- [United States - Amazon.com](#)
- [Available Worldwide - eBay.com](#)
- [United Kingdom - Amazon.co.uk](#)
- [Germany - Amazon.de](#)

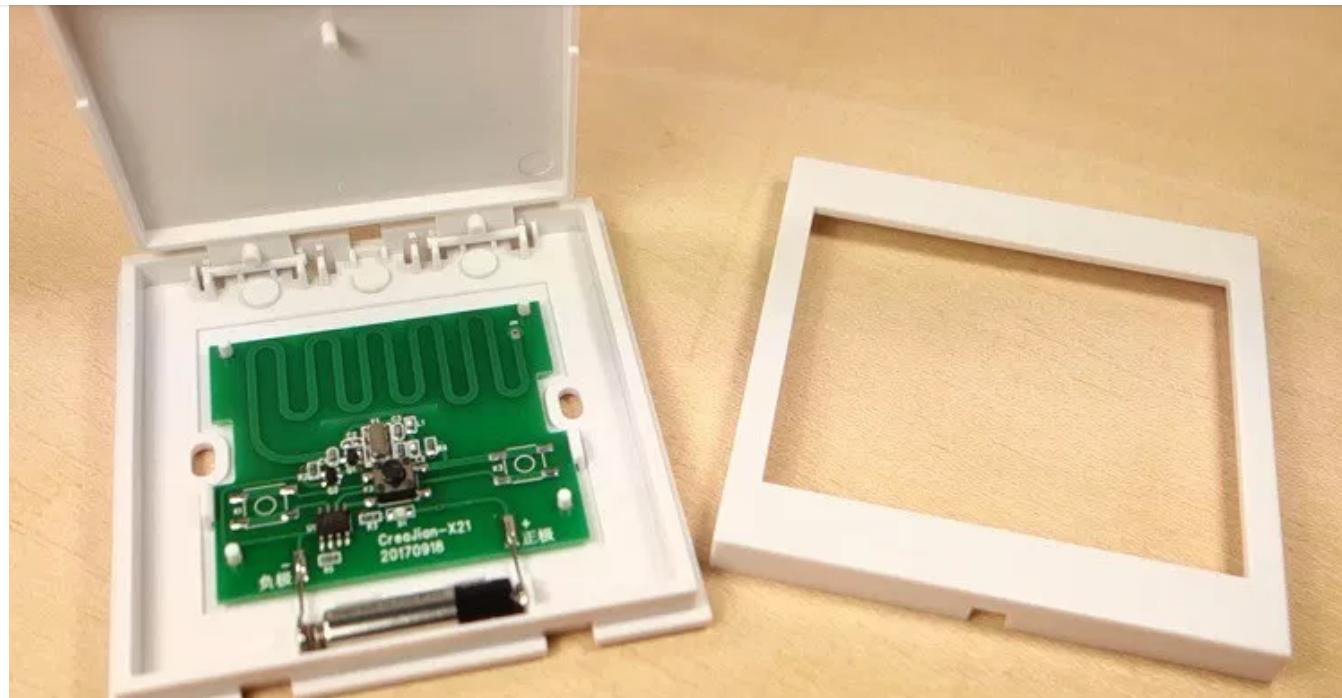
# 433 MHz RF Wall Panel Switch

The 433 MHz RF wall panel switch is a great way to remotely control devices. It can be easily attached to a wall with adhesive tape, without the need to make holes on the walls. Additionally, it is wireless, so you don't need to worry about wiring and then hiding cables.

In this project we're using two wall panel switches. Instead, you can use a panel switch with two buttons – there are also another version with three switches.



This wall panel switch has a push button in its circuit, as shown in the figure below, that when pressed emits a 433 MHz signal. You can use that signal to control whatever you want. This wall panel switch uses a 27A 12V type battery (not included in the package). So, you may want to buy one, when you get your wall panel switch.



# Decode the Wall Panel Switch 433 MHz RF Signals

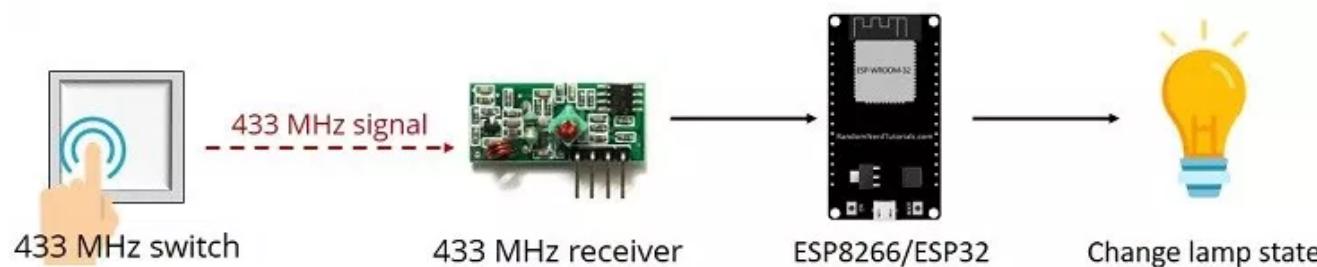
When you press the 433 MHz wall panel switch, it sends a 433 MHz signal. You need to decode that signal using a 433 MHz receiver. To learn how to decode the 433 MHz signal read the following post: [Decode and Send 433 MHz RF Signals with Arduino](#) – read the “Decoder Sketch” part. The sketch works with Arduino, ESP32, and ESP8266.

Decimal: 6819768 (24Bit) Binary: 01101000000011110111000 Tri-State: not app.  
Raw data: 7946,252,781,781,268,783,269,252,784,781,271,252,784,252,784,252,784  
  
Decimal: 9463928 (24Bit) Binary: 100100000110100001111000 Tri-State: not app.  
Raw data: 8064,796,273,258,793,258,793,795,274,257,796,256,796,256,796,256,796

In my case:

- switch 1: **6819768**
  - switch 2: **9463928**

You should get different values. You'll then use these signals in your ESP8266 or ESP32 sketch. When you press the switch, it sends a 433 MHz signal. This signal is detected by the receiver that is connected to the ESP. This way, the ESP knows the switch was pressed and it inverts the lamp's current state.



To control your ESP8266 or ESP32 with Amazon Echo, you need to install the FauxmoESP library. This library emulates a Belkin Wemo device, allowing you to control your ESP32 or ESP8266 using this protocol. This way, the Echo or Echo Dot instantly recognizes the device, after uploading the code, without any extra skills or third party services. You can read more about [FauxmoESP here](#).

## Installing the FauxmoESP Library

1. [Click here to download the FauxmoESP library](#). You should have a .zip folder in your Downloads
2. Unzip the .zip folder and you should get **xoseperez-fauxmoesp-50cbcf3087fd** folder
3. Rename your folder from **xoseperez-fauxmoesp-50cbcf3087fd** to **xoseperez\_fauxmoesp**
4. Move the xoseperez\_fauxmoesp folder to your Arduino IDE installation **libraries** folder
5. Finally, re-open your Arduino IDE

## Alexa – Echo Dot with ESP8266

Follow these next instructions if you're using an ESP8266.

### Installing the ESP8266 Board in Arduino IDE

In order to upload code to your ESP8266 using Arduino IDE, you should install an add-on for the Arduino IDE that allows you to program the ESP8266 using the

- [How to Install the ESP8266 Board in Arduino IDE.](#)

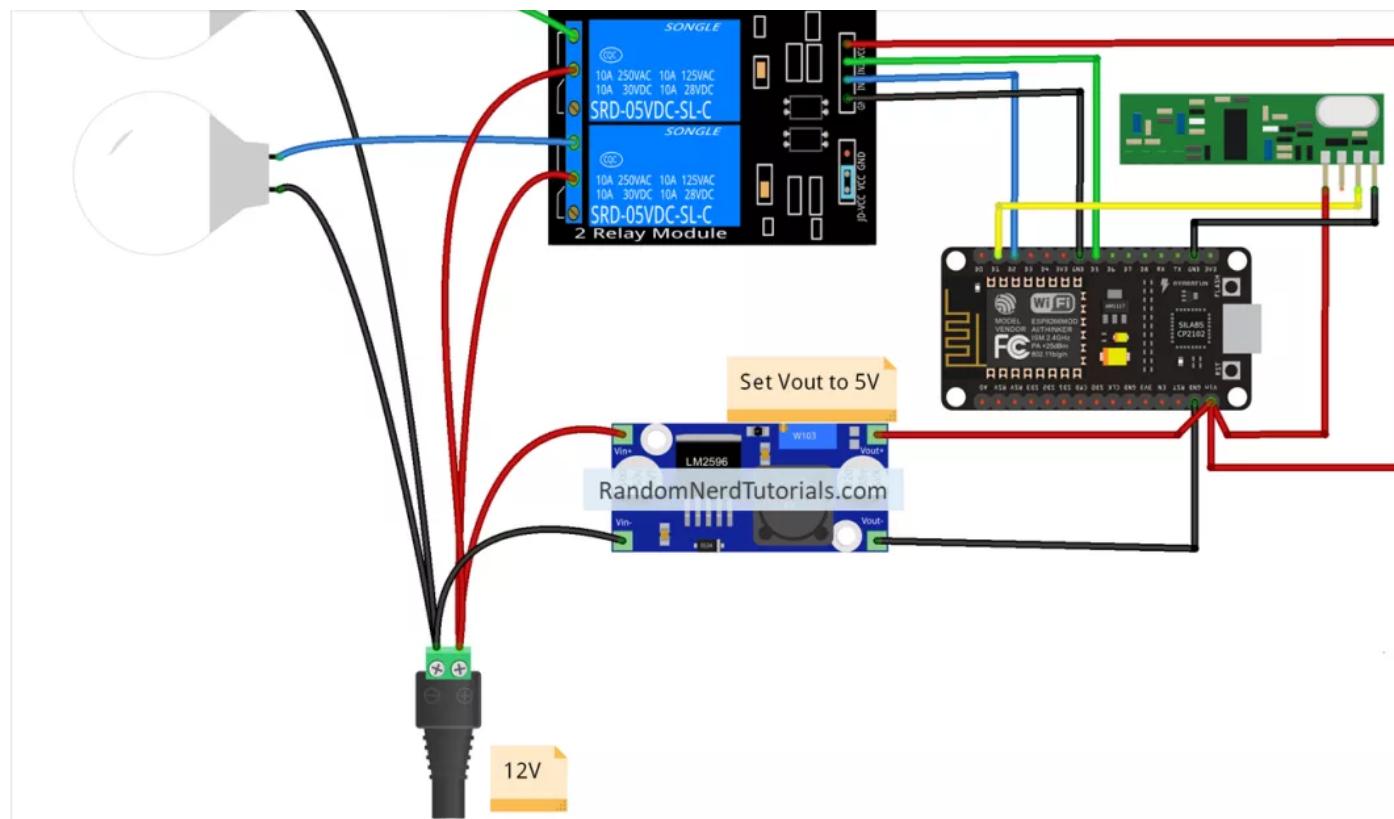
## Installing the ESPAsyncTCP Library

You also need to install the ESPAsyncTCP Library library. Follow the next instructions to install it:

1. [Click here to download the ESPAsyncTCP library.](#) You should have a .zip folder in your Downloads
2. Unzip the .zip folder and you should get **ESPAsyncTCP-master** folder
3. Rename your folder from **ESPAsyncTCP-master** to **ESPAsyncTCP**
4. Move the ESPAsyncTCP folder to your Arduino IDE installation **libraries** folder
5. Finally, re-open your Arduino IDE

## Schematic

If you're using an ESP8266 board, assemble your circuit by following the next schematic diagram – you can click the image to zoom.



If you're having trouble following the circuit diagram, you can use the following table as a reference:

ESP8266	Connect to
GPIO 5	433 MHz receiver data pin
GPIO 4	Relay IN1 pin

**IMPORTANT NOTE:** before applying power, make sure you set your step-down buck converter output voltage to 5V! Otherwise, you may damage your ESP.



## Alexa – Echo Dot with ESP32

Follow these next instructions if you're using an ESP32.

### Installing the ESP32 Board in Arduino IDE

Arduino IDE and its programming language. If you haven't installed the ESP32 add-on for the Arduino IDE, follow the next tutorial:

- [Windows instructions](#) – Installing the ESP32 Board in Arduino IDE
- [Mac and Linux instructions](#) – Installing the ESP32 Board in Arduino IDE

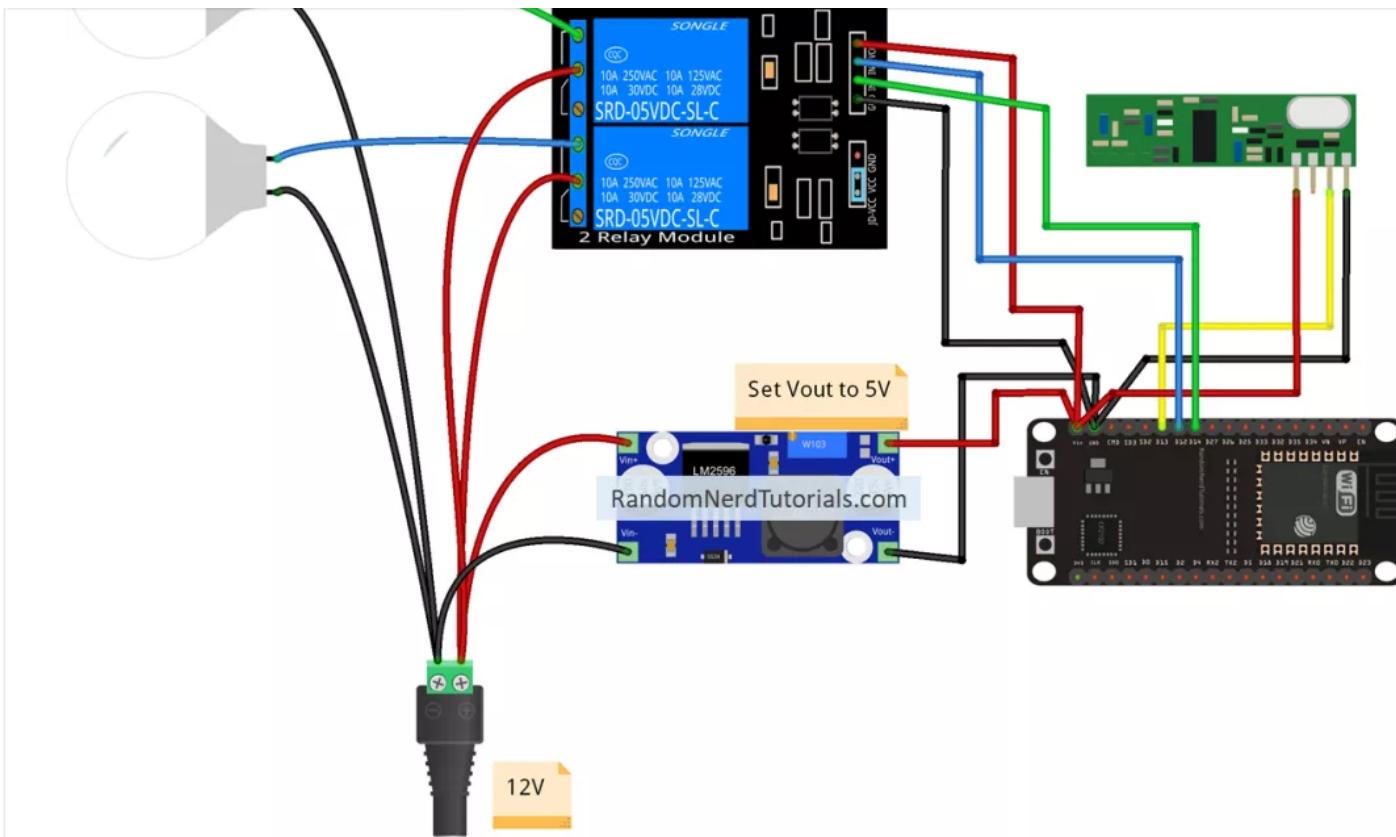
## Installing the AsyncTCP Library

You also need to install the AsyncTCP Library. Follow the next instructions to install it:

1. [Click here to download the AsyncTCP library](#). You should have a .zip folder in your Downloads
2. Unzip the .zip folder and you should get **AsyncTCP-master** folder
3. Rename your folder from **AsyncTCP-master** to **AsyncTCP**
4. Move the AsyncTCP folder to your Arduino IDE installation **libraries** folder
5. Finally, re-open your Arduino IDE

## Schematic

If you're using an ESP32 board, assemble your circuit by following the next schematic diagram – you can click the image to zoom.



If you're having trouble following the circuit diagram, you can use the following table as a reference:

ESP32	Connect to
GPIO 13	433 MHz receiver data pin
GPIO 14	Relay IN1 pin

**IMPORTANT NOTE:** before applying power, make sure you set your step-down buck converter output voltage to 5V! Otherwise, you may damage your ESP.



## Code

Copy the following code to your Arduino IDE, but don't upload it yet! You need to make some changes to make it work for you.

Rui Santos

\* Complete Project Details <https://randomnerdtutorials.com>

\*/

```
#include <Arduino.h>
#ifndef ESP32
    #include <WiFi.h>
    #define RF_RECEIVER 13
    #define RELAY_PIN_1 12
    #define RELAY_PIN_2 14
#else
    #include <ESP8266WiFi.h>
    #define RF_RECEIVER 5
    #define RELAY_PIN_1 4
    #define RELAY_PIN_2 14
#endif
#include "fauxmoESP.h"
```

```
#include <RCSwitch.h>
```

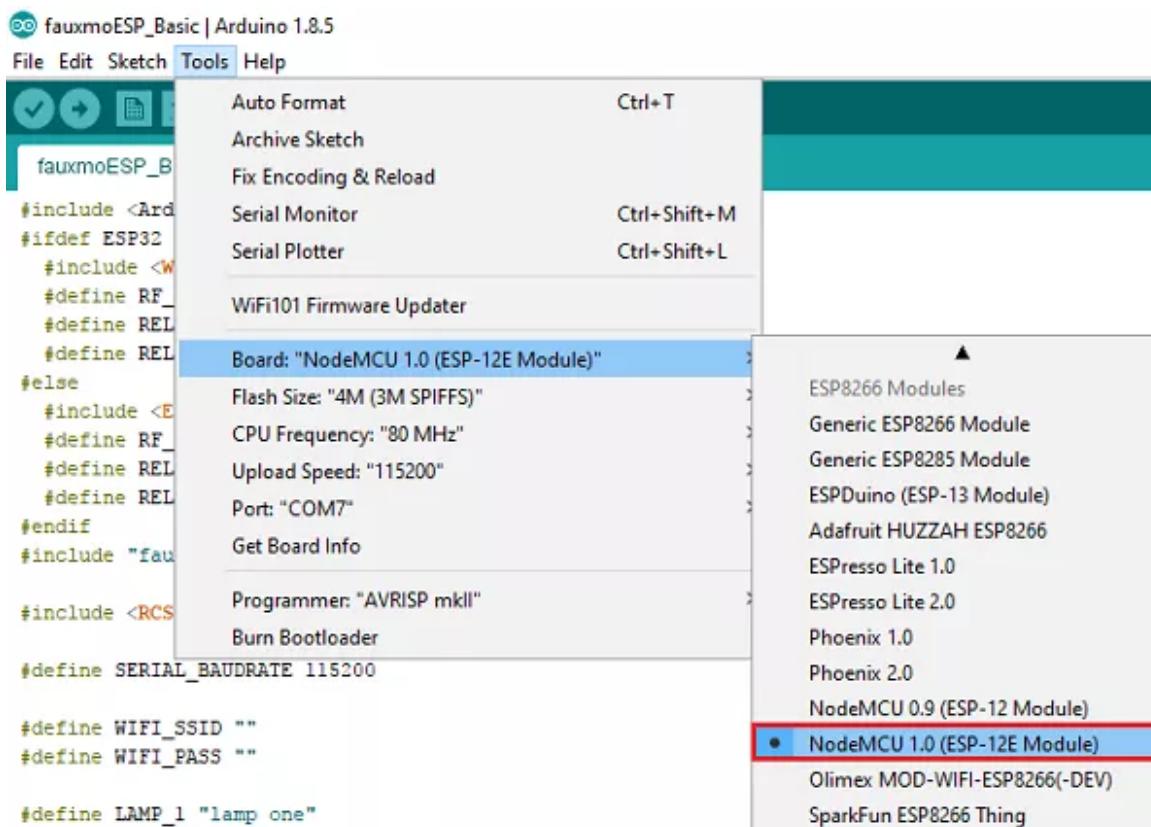
```
#define SERIAL_BAUDRATE 115200
```

```
#define WIFI_SSID "REPLACE_WITH_YOUR_SSID"
```

```
#define WIFI_PASS "REPLACE_WITH_YOUR_PASSWORD"
```

## Selecting the right board

This code works both with ESP32 and ESP8266. To make it work for your board, you need to select the board you're using in **Tools > Board**. Select your ESP8266 or ESP32 model.



You need to modify the following lines to include your network credentials.

```
#define WIFI_SSID "REPLACE_WITH_YOUR_SSID"  
#define WIFI_PASS "REPLACE_WITH_YOUR_PASSWORD"
```

## Add your 433 MHz signal codes

You also need to include the signals you've decoded previously for your wall panel switches.

Replace the value highlighted in red with the value you've gotten for the switch that controls lamp 1:

```
if (mySwitch.getReceivedValue()==6819768) {  
    digitalWrite(RELAY_PIN_1, !digitalRead(RELAY_PIN_1));  
}
```

And the value for lamp 2 in the following:

```
if (mySwitch.getReceivedValue()==9463928) {  
    digitalWrite(RELAY_PIN_2, !digitalRead(RELAY_PIN_2));  
}
```

After making all the necessary changes, you can upload code to your ESP. Make sure you have the right COM port selected, in **Tools > Port**. For demonstration purposes, you can open your Serial Monitor at a baud rate of 115200, while you prepare your Echo Dot.

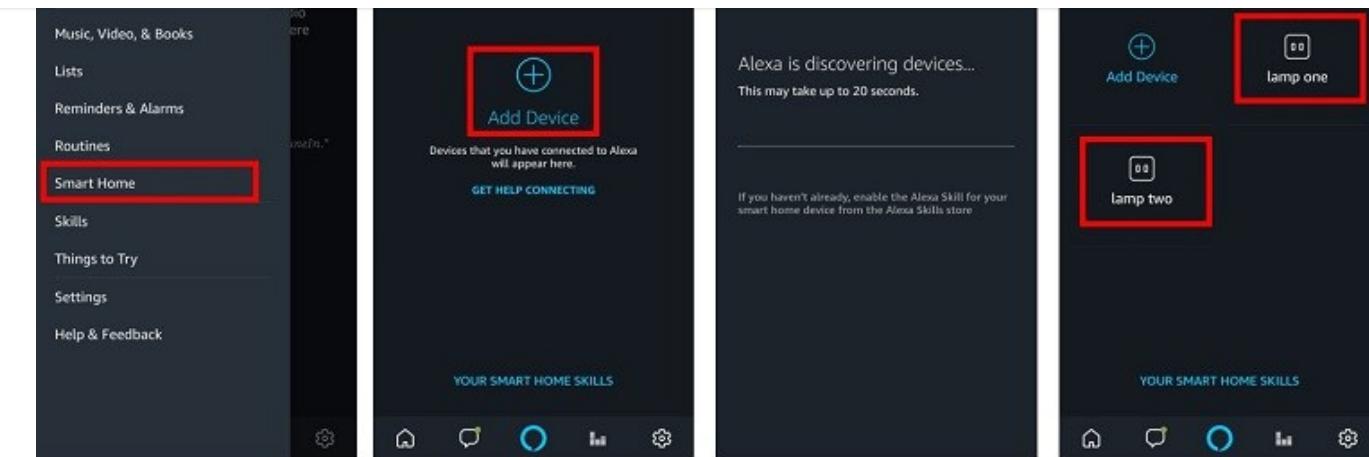
## Alexa, Discover Devices

With the circuit ready, and the code uploaded to your ESP8266 or ESP32, you need to ask alexa to discover devices.

Say: "Alexa, discover devices". It should answer as shown in the figure below.



Alternatively, you can also discover devices using the Amazon Alexa app, by following the steps shown in the figure below.



Then, ask Alexa to turn on/off the lamps. You'll also get information about the lamps state on the Serial Monitor.

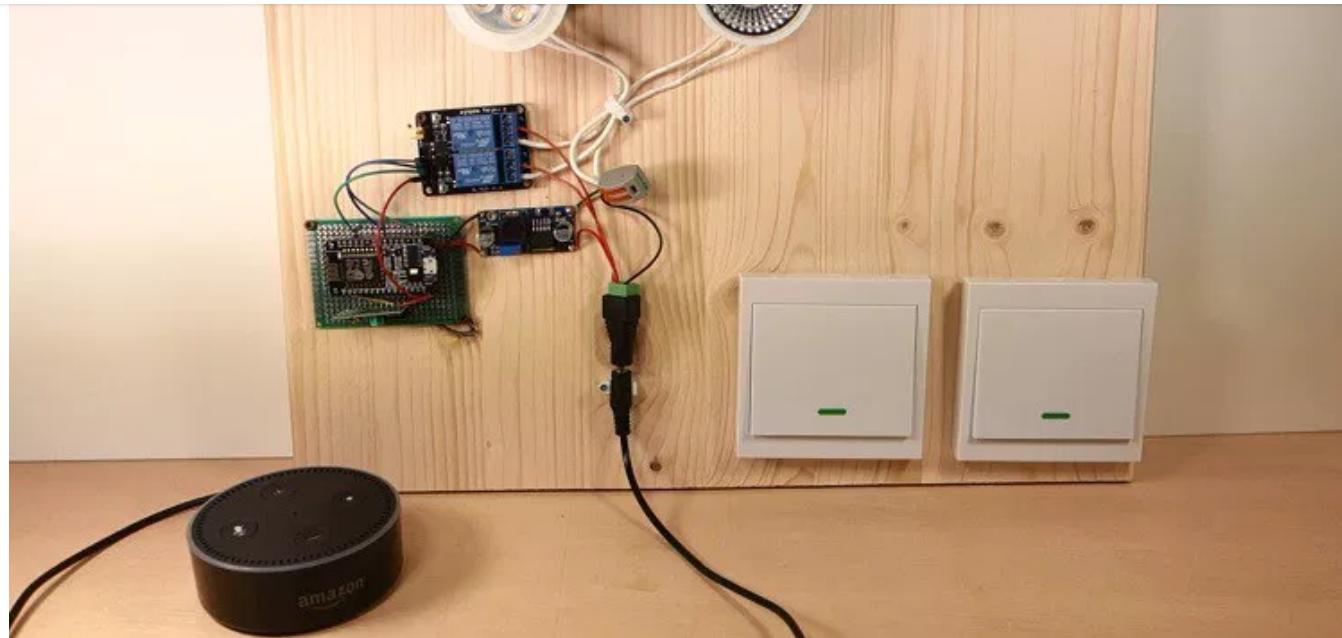
```
[WIFI] Connecting to MEO-620B4B .....  
[WIFI] STATION Mode, SSID: MEO-620B4B, IP address: 192.168.1.132  
[MAIN] Device #0 (lamp one) state: ON  
RELAY 1 switched by Alexa  
[MAIN] Device #0 (lamp one) state: OFF  
RELAY 1 switched by Alexa  
[MAIN] Device #1 (lamp two) state: ON  
RELAY 2 switched by Alexa  
[MAIN] Device #1 (lamp two) state: OFF  
RELAY 2 switched by Alexa  
[MAIN] Device #0 (lamp one) state: ON  
RELAY 1 switched by Alexa  
[MAIN] Device #0 (lamp one) state: OFF  
RELAY 1 switched by Alexa  
[MAIN] Device #1 (lamp two) state: OFF  
RELAY 2 switched by Alexa
```

Autoscroll No line ending 115200 baud Clear output

After making sure everything is working properly, you can turn your circuit into a permanent solution.

## Demonstration

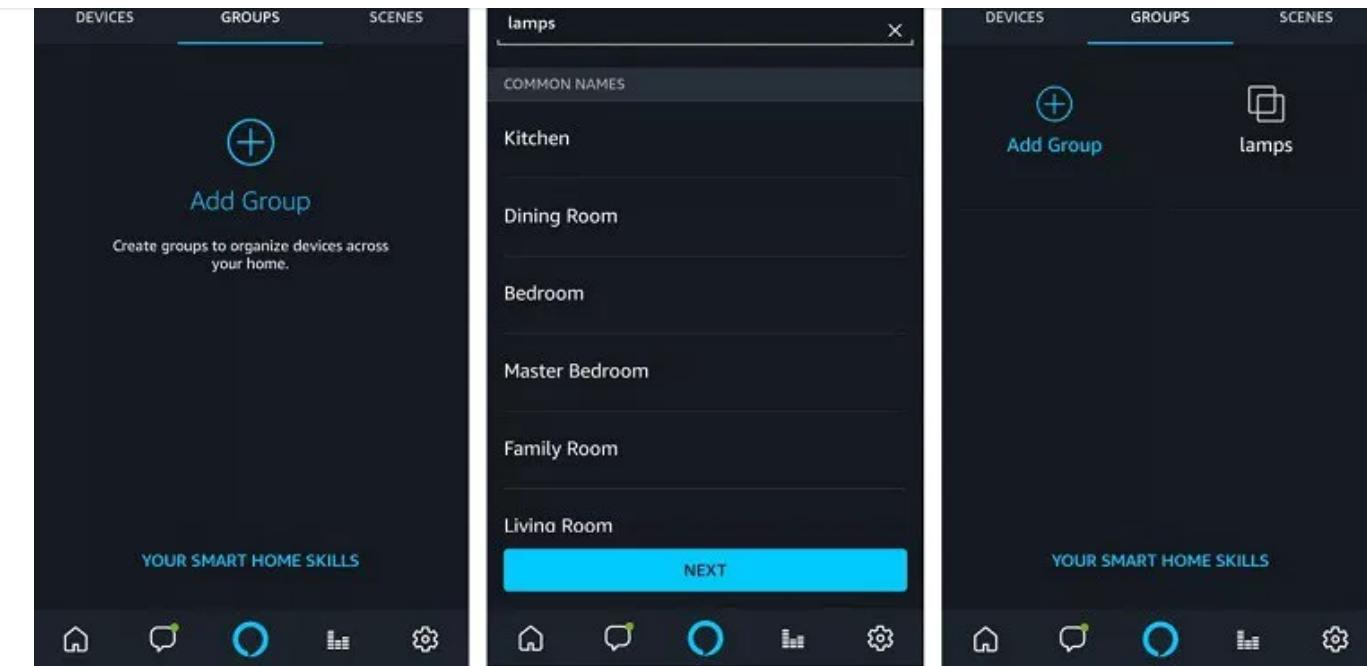
For demonstration purposes, we've built our circuit in a prototyping stripboard, and attached everything in a wooden board, as shown in the figure below:



Now you can ask Alexa to control your lamps with the following voice commands:

- “Alexa, turn on lamp 1”
- “Alexa, turn off lamp 1”
- “Alexa, turn on lamp 2”
- “Alexa, turn off lamp 2”

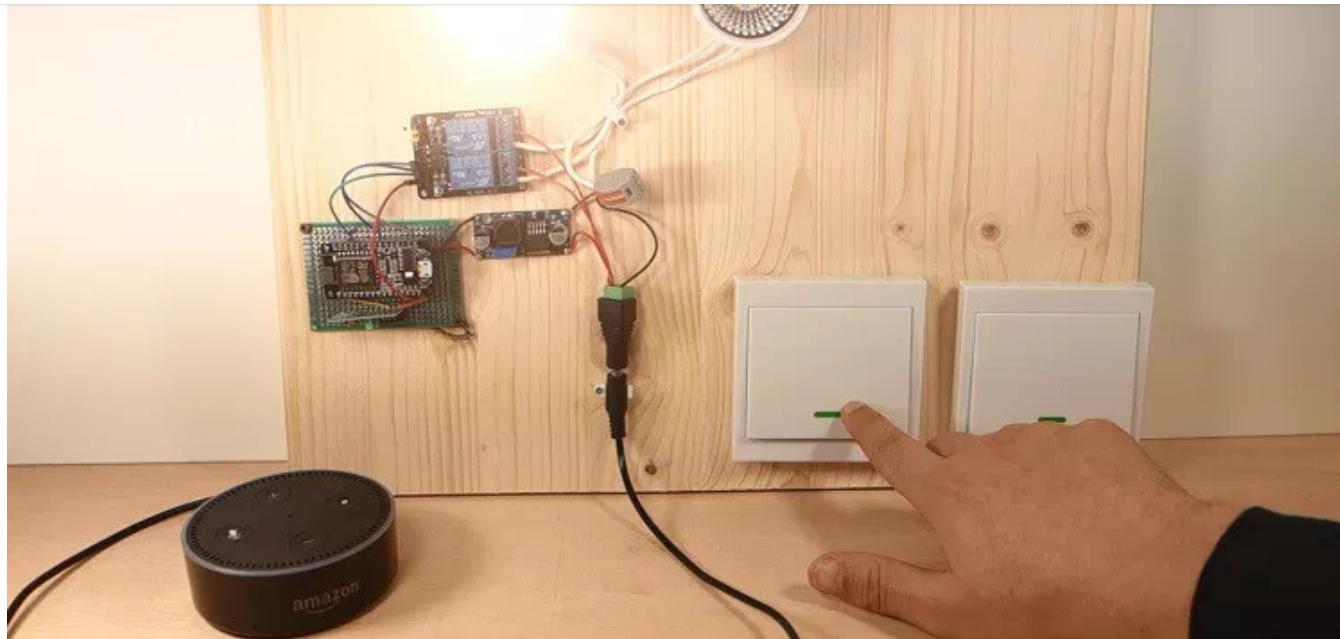
You can also control both lamps at the same time by creating a group in the Amazon Alexa app. We called it “lamps”.



Now, you can control both lamps at the same time, using the following voice commands.

- *"Alexa, turn on lamps"*
- *"Alexa, turn off lamps"*

You can also physically control your lamps using the 433 MHz wall panel switches.



## Wrapping Up

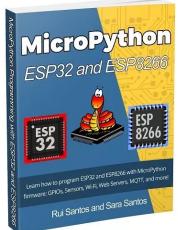
In this project we've shown how to control your ESP8266 and your ESP32 with voice commands using Amazon Echo. As an example, we've controlled two 12V lamps using a relay. Instead of 12V lamps, you can control any other electronics appliances. We've also shown you how you can remotely control your lamps using a 433 MHz wall panel switch.

We hope you've found this project useful. If you liked this post, you may also like:

- [Build a Home Automation System](#)
- [Home Automation using ESP8266](#)

■ [ESP8266 Daily Task – Publish temperature readings to thingspeak](#)

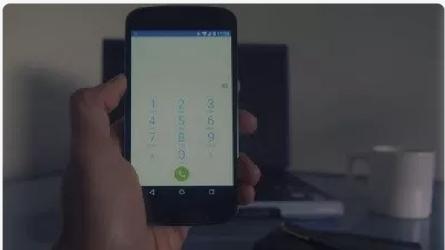
Thanks for reading.



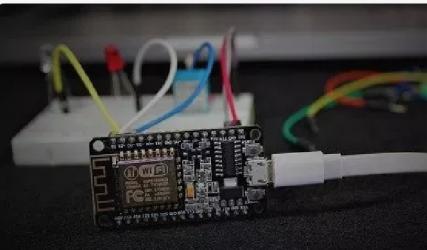
## [eBook] MicroPython Programming with ESP32 and ESP8266

Learn how to program and build projects with the ESP32 and ESP8266 using MicroPython firmware [DOWNLOAD »](#)

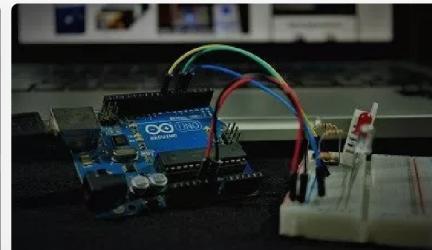
## Recommended Resources



[Build a Home Automation System from Scratch »](#) With Raspberry Pi, ESP8266, Arduino, and Node-RED.



[Home Automation using ESP8266 eBook and video course »](#) Build IoT and home automation projects.



[Arduino Step-by-Step Projects »](#) Build 25 Arduino projects with our course, even with no prior experience!

## WHAT TO READ NEXT...

[ESP8266 Deep Sleep with Arduino IDE](#)

[MicroPython: Interrupts with ESP32 and ESP8266](#)

[ESP32/ESP8266: DHT Temperature and Humidity Readings in OLED Display](#)

[Getting Started with ESP8266 WiFi Transceiver \(Review\)](#)

[Getting Started with MicroPython on ESP32 and ESP8266](#)

[ESP32 with PIR Motion Sensor using Interrupts and Timers](#)

**Enjoyed this project? Stay updated by subscribing our weekly newsletter!**

 [SUBSCRIBE](#)

**85 thoughts on “Alexa (Echo) with ESP32 and ESP8266 – Voice Controlled Relay”**



**John Haywood**

March 7, 2018 at 11:09 am | Reply

bluetooth) capability on the board already.

**Sara Santos**

March 7, 2018 at 5:42 pm | Reply

Hi John.

This project works with ESP32.

We give the instructions on how to make it work with the ESP32.

We've tested this example using that exact same ESP32 module  
(DOIT V1).

**John Haywood**

March 7, 2018 at 8:22 pm | Reply

Yes, you are right, sorry! I must have missed part of the board description and got confused with it saying the board had 36 GPIOs, when the board itself only has 30 pins.

Looking forward to making this tomorrow 😊



March 8, 2018 at 9:30 am | [Reply](#)

Hi John.

Note that the most recent versions of the DOIT board come with 36 pins.



**Lance Benson**

March 7, 2018 at 3:49 pm | [Reply](#)

What 12V LED lamps did you use? What wattage?



**John Haywood**

March 7, 2018 at 4:25 pm | [Reply](#)

Wattage would be governed by the relay contacts and any limitations of current supplied by your 12v power supply. The relay module shown has contacts rated at 10 amps so I think your absolute maximum rating for 12 volt LED lamps would be (10A x

**Sara Santos**March 7, 2018 at 5:41 pm | [Reply](#)

Hi Lance.

We're using GU5.3 12V lamps 5.5W.

We use these lamps because many people in our audience are beginners, so we prefer to show how to use relays using a 12V lamp (instead of mains voltage).

However, this tutorial could be easily applied to control any AC lamp.

Regards,

Sara

**chuck**March 7, 2018 at 5:04 pm | [Reply](#)

please show how this can be done without the echo / google devices for those of us that are skeptical about privacy concerns with these devices,

**Rui Santos**March 14, 2018 at 6:23 pm | [Reply](#)

Unfortunately, I haven't found a good solution to what you're looking for.

Regards,  
Rui

**Michael**November 2, 2018 at 7:58 pm | [Reply](#)

You can use the BitVoicer software. You can install it on your local computer in your home and do voice. [bitsophia.com/en-US/Home.aspx](http://bitsophia.com/en-US/Home.aspx)

**Sara Santos**November 4, 2018 at 11:00 am | [Reply](#)

**Kevin Hogan**March 7, 2018 at 8:23 pm | [Reply](#)

This is a fantastic post!!! Definitely, add it to your course! This is exactly a project I wanted to duplicate and was having difficulty with. Thanks!

**Sara Santos**March 8, 2018 at 9:29 am | [Reply](#)

Hi Kevin.

Thanks! We're glad you've found this project useful.

Regards,

Sara

**Jeff Young**March 7, 2018 at 8:41 pm | [Reply](#)

**Sara Santos**

March 8, 2018 at 9:29 am | Reply

Hi Jeff.  
Thank you 😊

**langtupt**

March 8, 2018 at 6:32 am | Reply

WOW, good job.  
Hi John,  
How to control device Lamp 1 and Lamp 2 in Google Home?  
Can you help share the tutorial.

Thank you very much

**Sara Santos**

March 8, 2018 at 9:33 am | Reply

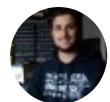
we don't have a Google Home yet.  
So, we can't help you with that project at the moment.

**niels**

March 10, 2018 at 10:15 pm | Reply

hi

I compiled the program, and uploaded it to my NodeMCU, on the monitor I got the information that it has connected to my WiFi.  
But Alexa did not find any new devices when asked! I have Belkin Wemo smart home skill enabled  
What should I do next?

**Rui Santos**

March 14, 2018 at 5:54 pm | Reply

Hi Niels,  
I'm not sure why that's happening, but a few others are reporting the same problem: [github.com/kakopappa/arduino-esp8266-alexa-multiple-wemo-switch/issues/22](https://github.com/kakopappa/arduino-esp8266-alexa-multiple-wemo-switch/issues/22)

**Michael**October 26, 2018 at 7:25 pm | [Reply](#)

I know there is some issues with 2nd Generation Echo Dot with Fauxmo. You might try the solution they mention on this link. If that doesn't work there is a client called Sinric.

[bitbucket.org/xoseperez/fauxmoesp/issues/35/device-not-responding-but-works](https://bitbucket.org/xoseperez/fauxmoesp/issues/35/device-not-responding-but-works)

Here is the link to Sinric:

[hackster.io/kakopappa/sinric-147d27](https://hackster.io/kakopappa/sinric-147d27)

Hope one of those works for you. I have a 2nd Gen but haven't got a chance to try it myself.

**Sara Santos**October 27, 2018 at 9:15 am | [Reply](#)

Thanj you Michael, for sharing that solution 😊

**Mike**November 16, 2018 at 5:14 am | [Reply](#)

example, as I just tried this as well (am going through the ESP32 course) and have had a few stumbles along the way... but Fauxmo not working with the latest generation of Echos makes the tutorial frustrating as you don't find out until the end when you (excitedly) go to discover your device to have it not show up, and have troubleshooted all the other issues along the way.



**Sara Santos**

November 16, 2018 at 10:14 am

Hi Mike.  
I'm sorry for the trouble.  
We need to update this tutorial for the new generation of Echos.  
I don't know when it will be, because at the moment we just have the 2nd generation. We need to get the 3rd generation to test everything.  
Meanwhile, I'll leave a note at the beginning of the post to inform that this just works with the 2nd generation.  
Regards,  
Sara 😊

**lizaboo**

March 16, 2018 at 8:23 pm | Reply

Thank you a great job. It looks great.

**Sara Santos**

March 17, 2018 at 11:43 am | Reply

Thanks 😊

**Duncan Amos**

March 23, 2018 at 7:25 pm | Reply

Watch out for the men in uniforms – 433MHz is not quite as 'free' as many people believe it to be, certainly within Europe.

In Asia, where almost all these devices originate, it really is more or less a free to use for anything band.

It's highly unlikely that RF switches would be noticed, but you never know!

Permitted use restrictions are incredibly complex in, their conditions and I don't pretend to understand them – this was just a 'heads-up'...

**Rajiv Shankar**April 2, 2018 at 11:30 am | [Reply](#)

Alexa could not find new devices. I tried to get it discovered by Amazon Alexa app as mentioned in your tutorial but without any success.

**Matej**April 4, 2018 at 12:41 pm | [Reply](#)

Hi

Do I really need the RF 433MHz Transmitter/Receiver for all of this to work?  
Will it work if I connect it only by Amazon dot?

Hi.

We use the 433 MHz transmitter because we've added a 433 MHz button to our setup.

You don't need to use the transmitter if you just want to use Amazon dot to control your lamps.

I hope this helps.

**Matej**

May 5, 2018 at 1:59 pm | [Reply](#)

Hi.

I have compiled the program with and without the transmitter codes, with those codes program doesn't even compile properly and without them, i can compile but can't upload to my module, any idea what to do? I followed all the steps and it doesn't work.

▼▼ June 25, 2018 at 12:13 am | Reply

Hi!!!

This project works great.Thanks.

Currently, I am looking for something to control 433 mhz outlets with alexa, can you recommend me where to look?

Best Regards.



**Sara Santos**

June 25, 2018 at 11:13 am | Reply

Hi Jose.

We usually recommend these 433 mhz outlets:

<https://makeradvisor.com/tools/rf-remote-controlled-sockets-433mhz/>

I hope this helps.

Regards,

Sara 😊



**Jose L Gonzalez**

June 26, 2018 at 11:54 pm | Reply

the remote control, and Blynk app, using a "sonoff basic", with a 433mhz transmitter. As I am not a programmer, I had to use copy and paste, of several different projects, until I got it to work. I use it, for lights, irrigation, air conditioning, etc. What I'd like to do is use Alexa, as a third way to use my devices, but I do not know how to do it in my sketch, or in yours.

Thanks.

**Duncan Amos**June 27, 2018 at 11:23 am | [Reply](#)

Yes, I was surprised that control of RF sockets wasn't included in. the project.

**Sara Santos**June 27, 2018 at 4:57 pm | [Reply](#)

Hi Jose.  
First, you need to add a 433MHz transmitter to the

[transmitter-module](#)), so that the ESP is able to send RF signals to control the sockets.

Then, you need to find the values sent by the transmitter to turn the sockets on and off. You can take a look at the first part of this tutorial to learn how to do that: [ESP8266 Remote Controlled Sockets](#)

Finally, you need to modify the code in the Alexa tutorial as follows.

In the following part of the code, in the first if statement, instead of using digitalWrite(), you should write the command to send the right RF signals to turn the socket on. And in the second if statement you should send the RF command to turn the socket off.

```
if (state) {  
    digitalWrite(RELAY_PIN_1, LOW);  
} else {  
    digitalWrite(RELAY_PIN_1, HIGH);  
}
```

I hope this helps.

You may also find useful taking a look at these next tutorials:

- [Decode and Send 433 MHz RF Signals with Arduino](#)
- [Complete Guide for RF 433MHz Transmitter/Receiver Module With Arduino](#)

▼▼ June 27, 2018 at 10:53 pm

Hi,Sara.  
I think this will help me so much.  
Thaks a lot.



**Jose L Gonzalez**

July 1, 2018 at 1:24 am

Hi Sara.  
I have added the code to use my etekcity  
433mhz outlets, and now I can use it, to control  
them with my Alexa Echo Dot. I have connected  
a transmitter, and I am using the project, for  
the lights of my bedside tables, and at the  
same time controlling the lights from the living  
room, kitchen, entrance, backyard, etc.  
I guess you can control other outlets, and other  
devices, just by sniffing codes.  
I hope this can be useful to other people.  
Greetings.

```
#include
#define RF_RECEIVER 13
#define RELAY_PIN_1 12
#define RELAY_PIN_2 14
#else
#include //I am using ESP8266-12E module
#define RF_RECEIVER 5
#define RELAY_PIN_1 4
#define RELAY_PIN_2 14
#endif
#include "fauxmoESP.h"

#include

#define SERIAL_BAUDRATE 115200

#define WIFI_SSID "XXXXXXXXXX"
#define WIFI_PASS "XXXXXXXXXX"

#define LAMP_1 "lamp one"
#define LAMP_2 "lamp two"
#define LAMP_3 "living light"
fauxmoESP fauxmo;

RCSwitch mySwitch = RCSwitch();
```

```
// Set WiFi module to STA mode
WiFi.mode(WIFI_STA);

// Connect
Serial.printf("[WIFI] Connecting to %s",
WIFI_SSID);
WiFi.begin(WIFI_SSID, WIFI_PASS);

// Wait
while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(100);
}
Serial.println();

// Connected!
Serial.printf("[WIFI] STATION Mode, SSID: %s, IP
address: %s\n", WiFi.SSID().c_str(),
WiFi.localIP().toString().c_str());
}

void setup() {
// Init serial port and clean garbage
Serial.begin(SERIAL_BAUDRATE);
Serial.println();
mySwitch.enableTransmit(12); // pin GPIO12
for 433mhz transmpter
```

```
// WiFi connection
wifiSetup();

// LED
pinMode(RELAY_PIN_1, OUTPUT);
digitalWrite(RELAY_PIN_1, HIGH);

pinMode(RELAY_PIN_2, OUTPUT);
digitalWrite(RELAY_PIN_2, HIGH);

mySwitch.enableReceive(RF_RECEIVER); //
Receiver on interrupt 0 => that is pin #2

// You can enable or disable the library at any
moment
// Disabling it will prevent the devices from
being discovered and switched
fauxmo.enable(true);

// Add virtual devices
fauxmo.addDevice(LAMP_1);
fauxmo.addDevice(LAMP_2);
fauxmo.addDevice(LAMP_3);
// you can add more devices

// fauxmoESP 2.0.0 has changed the callback
signature to add the device_id,
```

```
tauxmo.onSetState(LJ(unsigned char device_id,
const char * device_name, bool state) {
Serial.printf("[MAIN] Device # %d (%s) state:
%s\n", device_id, device_name, state ? "ON" :
"OFF");

if ( (strcmp(device_name, LAMP_1) == 0) ) {
// this just sets a variable that the main loop()
does something about
Serial.println("RELAY 1 switched by Alexa");
//digitalWrite(RELAY_PIN_1,
!digitalRead(RELAY_PIN_1));
if (state) {
digitalWrite(RELAY_PIN_1, LOW);
} else {
digitalWrite(RELAY_PIN_1, HIGH);
}
}

if ( (strcmp(device_name, LAMP_2) == 0) ) {
// this just sets a variable that the main loop()
does something about
Serial.println("RELAY 2 switched by Alexa");
if (state) {
digitalWrite(RELAY_PIN_2, LOW);
} else {
digitalWrite(RELAY_PIN_2, HIGH);
}
```

```
    if (state) {
        mySwitch.send("010100000101010100110011")
        ; // etekcity outlet #1 connected to my living
        light "ON" binary code
    } else {
        mySwitch.send("010100000101010100111100")
        ; // etekcity outlet #1 connected to my living
        light "OFF" binary code
    }
}

});

// Callback to retrieve current state (for
GetBinaryState queries)
/*fauxmo.onGetState([](unsigned char
device_id, const char * device_name) {
//return !digitalRead(RELAY_PIN_1);
return
});*/
}

void loop() {
// Since fauxmoESP 2.0 the library uses the
"compatibility" mode by
// default, this means that it uses WiFiUdp class
instead of AsyncUDP.
```

```
// whilst the former works fine with current
// stable 2.3.0 version.
// But, since it's not "async" anymore we have
// to manually poll for UDP
// packets
fauxmo.handle();

/*static unsigned long last = millis();
if (millis() - last > 5000) {
last = millis();
Serial.printf("[MAIN] Free heap: %d bytes\n",
ESP.getFreeHeap());
}*/

if (mySwitch.available()) {
/*Serial.print("Received ");
Serial.print( mySwitch.getReceivedValue() );
Serial.print(" / ");
Serial.print( mySwitch.getReceivedBitlength() );
Serial.print("bit ");
Serial.print("Protocol: ");
Serial.println( mySwitch.getReceivedProtocol()
);*/
if (mySwitch.getReceivedValue()==6819768) {
digitalWrite(RELAY_PIN_1,
!digitalRead(RELAY_PIN_1));
}
```

```
!digitalRead(RELAY_PIN_2));  
}  
delay(600);  
mySwitch.resetAvailable();  
}  
}
```



### John Haywood

June 26, 2018 at 7:18 am | Reply

Depending on your exact needs, this project would work fine except for the relay board. You would need to look at other relay boards to find ones that have suitably rated relays (for the amperage you wish to use).

OR you could simply build your own power relay board and use this project to turn on your home made relay board instead of 12 volt bulbs.



### Duncan Amos

July 1, 2018 at 8:11 am | Reply

other RF frequencies), this video by Andreas Spiess ("the guy with the Swiss accent") is excellent...

<https://youtu.be/L0fSEbGEY-Q>



**Trần Viết Phương**

July 8, 2018 at 3:13 am | Reply

Thank you, please share



**Rui Santos**

July 11, 2018 at 10:29 am | Reply

You're welcome. Thanks for reading!

Rui



**Fabian**

July 29, 2018 at 11:54 pm | Reply

how the code read this instruction in the ESP8266 ?

**Sascha**

October 26, 2018 at 7:42 am | Reply

Hello, i have a Problem in the Arduino IDE, but i can`t find it. Can you please help me?

Alexa:76: error: expected primary-expression before 'unsigned'

```
fauxmo.onSetState((unsigned char device_id, const char * device_name,  
bool state) {
```

^

Alexa:76: error: expected ')' before 'unsigned'

Alexa:139: error: expected ';' at end of input

}

^

```
exit status 1  
expected primary-expression before 'unsigned'
```

**Sara Santos**

October 29, 2018 at 9:59 am | Reply

Hi Sascha.

Did you modify our code example?

It seems you have a syntax error: you have a ) and } missing on in the wrong place.

Please double-check your code indentation is correct. See if you have line 97 on your code:

[https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/Alexa/ESP32\\_ESP8266\\_2\\_Devices.ino](https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/Alexa/ESP32_ESP8266_2_Devices.ino)

Regards,

Sara

**Icarusx**

October 30, 2018 at 9:48 am | Reply

```
Is missing... ", unsigned char value" in the  
fauxmo.onSetState...
```

**Giovanni Roberto Mariani**November 7, 2018 at 11:15 am | [Reply](#)

I compiled the sketch for esp development module and I've received several error with fauxmoesp. Did somebody already tested for this environment?

**Sara Santos**November 8, 2018 at 10:11 am | [Reply](#)

Hi Giovanni.  
What kind of error are you getting?  
Regards,  
Sara 😊

**Duncan**November 8, 2018 at 1:04 pm | [Reply](#)

to trying it) – the error I get is:  
'class fauxmoESP' has no member named 'onSetState'

I get the same error if I try the example sketch that comes  
with the fauxmo library.

Wemos D1 Mini, fauxmo v3.0.2



**Sara Santos**

November 9, 2018 at 1:05 pm | Reply

Hi Duncan.  
There are other people reporting that error.  
Honestly I don't know why it is happening. Maybe it  
has something to do with some library updates.  
Meanwhile if you find what is causing the error, please  
let us know.  
Regards,  
Sara 😊



**Duncan Amos**

November 9, 2018 at 1:51 pm

1 hour ago.

In my case, I had two versions of the faumo library in different places and although I'd updated one of them, it was trying to use the other one (there must be some method for the order of looking for libraries but I have no idea what it is). Overwriting the old version with the new version fixed that particular problem.

But...

The method used by the library has changed significantly, which means that earlier examples of programs that use the library now throw other errors...

Your example will run on a D1 Mini but won't run on a Sonoff (Board: Generic ESP8266 and pins changed to suit). I tried your earlier Sonoff and Alexa example and that continually throws watchdog resets.

I think it would be useful to include the version of all libraries you use as a comment. Once the code is compiled, uploaded and running, it's all



**Sara Santos**

November 10, 2018 at 10:53 am

Hi Duncan. Thank you for sharing your solution.

You are right, sometimes when the libraries update, older examples no longer work. So, it would be useful to include the library version.

Thank you.

Regards,

Sara 😊



**Duncan Amos**

November 10, 2018 at 4:13 pm

Heads up..

Neither of my two new 3rd Generation Echo Dots would find new Fauxmo devices – ask the 2nd Gen Dot and it found everything immediately. Once ‘on the list’ the Gen 3s control them OK.

Amazon are aware of the problem, but...

**Sara Santos**

November 9, 2018 at 1:08 pm | Reply

Can you follow the suggestions in this topic and see if it works?  
[github.com/xoseperez/espurna/issues/406](https://github.com/xoseperez/espurna/issues/406)

**Phil Nicholson**

November 9, 2018 at 9:12 am | Reply

function for call to 'fauxmoESP::onSetState(setup()):'

**Sara Santos**

November 9, 2018 at 1:09 pm | [Reply](#)

Hi Phil.

Can you follow the suggestions on this topic and see if it works?

[github.com/xoseperez/espurna/issues/406](https://github.com/xoseperez/espurna/issues/406)

Regards,

Sara

**Rui Santos**

November 10, 2018 at 7:07 pm | [Reply](#)

Are you using the latest version of the FauxmoESP library? Can you remove the old installation, re-download the latest version? Does it work now? Thanks!



December 9, 2018 at 6:33 pm | Reply

I tried compiling the above code but failed , the error was

alexax\_esp:97: error: no matching function for call to  
'fauxmoESP::onSetState(setup()):'

then i read over and found this

<https://github.com/xoseperez/espurna/issues/1141>

went over to bitbucket and downloaded version 2.4 of fauxmo here

<https://bitbucket.org/xoseperez/fauxmoesp/get/2.4.0.zip>

and it compiled



**Sara Santos**

December 10, 2018 at 9:56 am | Reply

Hi, thank you for sharing that solution.

The code we have in the example no longer works with the old  
version of the library.

Regards,

Sara 😊

**Mason Jones**

December 11, 2018 at 1:21 am | Reply

Hi i've tried to verify/upload the following code

```
#include
#ifndef ESP32
#include
#define RF_RECEIVER 13
#define RELAY_PIN_1 12
#define RELAY_PIN_2 14
#else
#include
#define RF_RECEIVER 5
#define RELAY_PIN_1 4
#define RELAY_PIN_2 14
#endif
#include "fauxmoESP.h"

#include
#define SERIAL_BAUDRATE 115200

#define WIFI_SSID "MY_NETWORK"
#define WIFI_PASS "MY_NETWORK_PASSWORD"
```

```
fauxmoESP fauxmo;

RCSwitch mySwitch = RCSwitch();

// Wi-Fi Connection
void wifiSetup() {
// Set WIFI module to STA mode
WiFi.mode(WIFI_STA);

// Connect
Serial.printf("[WIFI] Connecting to %s ", WIFI_SSID);
WiFi.begin(WIFI_SSID, WIFI_PASS);

// Wait
while (WiFi.status() != WL_CONNECTED) {
Serial.print(".");
delay(100);
}
Serial.println();

// Connected!
Serial.printf("[WIFI] STATION Mode, SSID: %s, IP address: %s\n",
WiFi.SSID().c_str(), WiFi.localIP().toString().c_str());
}
```

```
Serial.begin(SERIAL_BAUDRATE);
Serial.println();

// Wi-Fi connection
wifiSetup();

// LED
pinMode(RELAY_PIN_1, OUTPUT);
digitalWrite(RELAY_PIN_1, HIGH);

pinMode(RELAY_PIN_2, OUTPUT);
digitalWrite(RELAY_PIN_2, HIGH);

mySwitch.enableReceive(RF_RECEIVER); // Receiver on interrupt 0 => that is
pin #2

// You can enable or disable the library at any moment
// Disabling it will prevent the devices from being discovered and switched
fauxmo.enable(true);

// Add virtual devices
fauxmo.addDevice(LAMP_1);
fauxmo.addDevice(LAMP_2);

// fauxmoESP 2.0.0 has changed the callback signature to add the
device_id,
// this way it's easier to match devices to action without having to compare
```

```
door state) {  
Serial.printf("[MAIN] Device #%d (%s) state: %s\n", device_id, device_name,  
state ? "ON" : "OFF");  
if ( (strcmp(device_name, LAMP_1) == 0) ) {  
// this just sets a variable that the main loop() does something about  
Serial.println("RELAY 1 switched by Alexa");  
//digitalWrite(RELAY_PIN_1, !digitalRead(RELAY_PIN_1));  
if (state) {  
digitalWrite(RELAY_PIN_1, LOW);  
} else {  
digitalWrite(RELAY_PIN_1, HIGH);  
}  
}  
  
if ( (strcmp(device_name, LAMP_2) == 0) ) {  
// this just sets a variable that the main loop() does something about  
Serial.println("RELAY 2 switched by Alexa");  
if (state) {  
digitalWrite(RELAY_PIN_2, LOW);  
} else {  
digitalWrite(RELAY_PIN_2, HIGH);  
}  
}  
};  
  
});  
  
// Callback to retrieve current state (for GetBinaryState queries)  
/*fauxmo.onGetState([](unsigned char device_id, const char *  
device_name) {
```

```
});^/
}

void loop() {
// Since fauxmoESP 2.0 the library uses the "compatibility" mode by
// default, this means that it uses WiFiUpd class instead of AsyncUDP.
// The later requires the Arduino Core for ESP8266 staging version
// whilst the former works fine with current stable 2.3.0 version.
// But, since it's not "async" anymore we have to manually poll for UDP
// packets
fauxmo.handle();

/*static unsigned long last = millis();
if (millis() - last > 5000) {
last = millis();
Serial.printf("[MAIN] Free heap: %d bytes\n", ESP.getFreeHeap());
}*/

if (mySwitch.available()) {
/*Serial.print("Received ");
Serial.print( mySwitch.getReceivedValue() );
Serial.print(" / ");
Serial.print( mySwitch.getReceivedBitlength() );
Serial.print("bit ");
Serial.print("Protocol: ");
Serial.println( mySwitch.getReceivedProtocol() );*/
if (mySwitch.getReceivedValue()==6819768) {
```

```
if (mySwitch.getReceivedValue() == 9463928) {  
    digitalWrite(RELAY_PIN_2, !digitalRead(RELAY_PIN_2));  
}  
delay(600);  
mySwitch.resetAvailable();  
}  
}
```

and the following message comes up

```
/Users/masonjones/Documents/Arduino/libraries/xoseperez-  
fauxmoesp/src/fauxmoESP.h:106:9: error: 'WiFiEventHandler' does not  
name a type  
WiFiEventHandler _handler;  
^  
exit status 1  
Error compiling for board NodeMCU 1.0 (ESP-12E Module).
```

any change you could help me?



**Sara Santos**

December 14, 2018 at 5:28 pm | Reply

Can you try Belal Quamar suggestions on his comment:

"I tried compiling the above code but failed , the error was

alexu\_esp:97: error: no matching function for call to  
'fauxmoESP::onSetState(setup()):'

then i read over and found this

<https://github.com/xoseperez/espurna/issues/1141>

went over to bitbucket and downloaded version 2.4 of fauxmo here

<https://bitbucket.org/xoseperez/fauxmoesp/get/2.4.0.zip>

and it compiled"

Regards,  
Sara



**Mario**

December 29, 2018 at 6:04 pm | Reply

Hi Rui,

I have compiled the code for ESP32 and I get this error. Can you help me ?

\*\*\* ERROR \*\*\*

no matching function for call to 'fauxmoESP :: onSetState (setup () :: )'

\*\*\* ERROR \*\*\*



**Sara Santos**

December 31, 2018 at 12:44 pm | Reply

Hi Mario.

Our example works with an older version of the library (2.4.0).

<https://github.com/xoseperez/espurna/issues/1141>

You can either use the older version or update the sketch to use the newer version.

You can go to bitbucket and downloaded version 2.4 of fauxmo here

<https://bitbucket.org/xoseperez/fauxmoesp/get/2.4.0.zip>

I hope this helps.

Regards,

Sara

**MiiCode2**

January 4, 2019 at 6:02 am | Reply

Echo Dot 3 Users.

I should note that this now works with FauxmoESP 3.1.0.

You also need to make couple of code changes.

change

```
fauxmo.enable(true);
```

to:

```
fauxmo.setPort(80); // required for gen 3 devices  
fauxmo.enable(true);
```

also change

```
fauxmo.onSetState([](unsigned char device_id, const char * device_name,  
bool state) {
```

to

Also, when adding new devices on the Alexa App, just hit the + symbol while on devices and select Add Device and then select Other at the bottom of the list. Lamp one and Lamp two will be found.

Works perfectly. I am using the code with an ESP8266

**Sara Santos**

January 7, 2019 at 11:36 pm | Reply

Hi.

Thank you so much for sharing this solution. We don't have any Echo Dot 3rd generation to test the code.

However, we'll add a note with this instructions in our tutorial.

Thank you,

Regards,

Sara

**Denis Brion**

January 8, 2019 at 4:47 pm | Reply

Do alexa black boxes recognise Portuguese (or French, or Arabic) sentences?

Can one “train” this black box to cope with someone who is often hoarse?

**Sara Santos**

January 8, 2019 at 5:08 pm | Reply

Hi Dennis.

In my Alexa app, it supports these languages: English (with different accents), French (with different accents), Deutsch, and Chinese.

Regards,

Sara

**David**

January 13, 2019 at 8:04 pm | Reply

Great tutorial, thanks. I am trying to control 4 relays so I have added to your code by duplicating the lamp info and changing its I'd etc. The 3rd relay works fine but I can't get alexa to discover the 4th. Any ideas why that

**Sara Santos**

January 14, 2019 at 10:30 am | Reply

Hi David.

Without any further information, it is difficult to find out what is going on.

Sorry that I can't help.

Regards,

Sara

**David**

January 20, 2019 at 12:26 am | Reply

this id the code i used, I've now lost lamp 3 aswell

```
/*
 * Rui Santos
 * Complete Project Details https://randomnerdtutorials.com
 */
```

```
#include
```

```
#define RELAY_PIN_2 14
#define RELAY_PIN_3 5
#define RELAY_PIN_4 12
#include "fauxmoESP.h"

#define SERIAL_BAUDRATE 115200

#define WIFI_SSID "BTHub6-CFR5"
#define WIFI_PASS "RP3wpV9DcNMh"

#define LAMP_1 "lamp one"
#define LAMP_2 "lamp two"
#define LAMP_3 "lamp three"
#define LAMP_4 "lamp four"
fauxmoESP fauxmo;

// Wi-Fi Connection
void wifiSetup() {
// Set WIFI module to STA mode
WiFi.mode(WIFI_STA);

// Connect
Serial.printf("[WIFI] Connecting to %s ", WIFI_SSID);
WiFi.begin(WIFI_SSID, WIFI_PASS);

// Wait
while (WiFi.status() != WL_CONNECTED) {
```

```
}

Serial.println();

// Connected!
Serial.printf("[WIFI] STATION Mode, SSID: %s, IP address:
%s\n", WiFi.SSID().c_str(), WiFi.localIP().toString().c_str());
}

void setup() {
// Init serial port and clean garbage
Serial.begin(SERIAL_BAUDRATE);
Serial.println();

// Wi-Fi connection
wifiSetup();

// LED
pinMode(RELAY_PIN_1, OUTPUT);
digitalWrite(RELAY_PIN_1, HIGH);

pinMode(RELAY_PIN_2, OUTPUT);
digitalWrite(RELAY_PIN_2, HIGH);

// By default, fauxmoESP creates it's own webserver on the
defined port
// The TCP port must be 80 for gen3 devices (default is 1901)
// This has to be done before the call to enable()
```

```
fauxmo.setPort(80); // This is required for gen3 devices

// You have to call enable(true) once you have a WiFi
connection
// You can enable or disable the library at any moment
// Disabling it will prevent the devices from being discovered
and switched
fauxmo.enable(true);
// You can use different ways to invoke alexa to modify the
devices state:
// "Alexa, turn lamp two on"

// Add virtual devices
fauxmo.addDevice(LAMP_1);
fauxmo.addDevice(LAMP_2);
fauxmo.addDevice(LAMP_3);
fauxmo.addDevice(LAMP_4);

fauxmo.onSetState([](unsigned char device_id, const char *
device_name, bool state, unsigned char value) {
// Callback when a command from Alexa is received.
// You can use device_id or device_name to choose the
element to perform an action onto (relay, LED,...)
// State is a boolean (ON/OFF) and value a number from 0 to
255 (if you say "set kitchen light to 50%" you will receive a 128
here).
// Just remember not to delay too much here, this is a
```

and process it in your main loop.

```
Serial.printf("[MAIN] Device #%d (%s) state: %s value: %d\n",
device_id, device_name, state ? "ON" : "OFF", value);
if ( (strcmp(device_name, LAMP_1) == 0) ) {
// this just sets a variable that the main loop() does
something about
Serial.println("RELAY 1 switched by Alexa");
//digitalWrite(RELAY_PIN_1, !digitalRead(RELAY_PIN_1));
if (state) {
digitalWrite(RELAY_PIN_1, LOW);
} else {
digitalWrite(RELAY_PIN_1, HIGH);
}
}
if ( (strcmp(device_name, LAMP_2) == 0) ) {
// this just sets a variable that the main loop() does
something about
Serial.println("RELAY 2 switched by Alexa");
if (state) {
digitalWrite(RELAY_PIN_2, LOW);
} else {
digitalWrite(RELAY_PIN_2, HIGH);
}
}
if ( (strcmp(device_name, LAMP_3) == 0) ) {
// this just sets a variable that the main loop() does
```

```
if (state) {
    digitalWrite(RELAY_PIN_3, LOW);
} else {
    digitalWrite(RELAY_PIN_3, HIGH);
}
}

if ( (strcmp(device_name, LAMP_4) == 0) ) {
    // this just sets a variable that the main loop() does
    // something about
    Serial.println("RELAY 4 switched by Alexa");
    if (state) {
        digitalWrite(RELAY_PIN_4, LOW);
    } else {
        digitalWrite(RELAY_PIN_4, HIGH);
    }
}
});

}

void loop() {
    // fauxmoESP uses an async TCP server but a sync UDP
    // server
    // Therefore, we have to manually poll for UDP packets
    fauxmo.handle();
}
```

```
last = millis();  
Serial.printf("[MAIN] Free heap: %d bytes\n",  
ESP.getFreeHeap());  
}  
}
```

**Sara Santos**

January 22, 2019 at 10:45 am | Reply

Hi David.  
Your code seems good.  
Except, you have two void #includes at the beginning  
of the code and you don't include the WiFi library. I'm  
not sure if it was an error copying the code. You also  
didn't set the pinMode for relay 3 and relay 4.  
Do you get any errors?

**David**

January 23, 2019 at 6:26 pm

Hi Sara, if I run the code as it is I don't get any  
errors,

error.

Thanks.

**Domenico**January 18, 2019 at 7:56 am | [Reply](#)

Dear Sara,

I am facing some problems when running this project; Let me explain a bit better to allow you to have a clear status.

I used a NodeMCU1.0, uploaded the code successfully and connected to the domestic Wifi without any problem. However, when I try to turn ON and then OFF, or viceversa using the 433Mhz Switch, the system immediately dis-connected by wifi.

It does not always happen but often causing the difficulty of reliability of the system itself. By the way, I tried to use other esp8266 and 433mhz receivers but nothing.

I removed the relay but nothing, just used only two leds as an output devices.

I tried to use alternative 433Mhz switch's but the scenario was not changed.

If you need more details, or needed some other trial runs, pls let me know.

**Sara Santos**

January 19, 2019 at 6:38 pm | Reply

Hi Domenico.

I'm sorry you're getting that issue.

Have you tried using just the switches? Also, are you getting any errors on the serial monitor when the Wi-Fi disconnects?

Regards,

Sara

**Domenico Carvetta**

January 19, 2019 at 3:00 pm | Reply

Dear Sara,

I am facing some problems when running this project;

Let me explain a bit better to allow you to have a clear status.

I used a NodeMCU1.0, uploaded the code successfully and connected to the domestic Wifi

without any problem. However, when I try to turn ON and then OFF, or viceversa using the 433Mhz Switch, the system immediately dis-

the system itself.

By the way, I tried to use other esp8266 and 433mhz receivers but nothing. I removed the relay but nothing, just used only two leds as an output devices.

I tried to use alternative 433Mhz switch's but the scenario was not changed.

If you need more details, or needed some other trial runs, pls let me know. How can I solve this annoying problem? thanks for your appreciated comments. I stay tuned. Domenico



**Domenico**

January 24, 2019 at 1:32 pm | Reply

Dear Sara, I come back to you on this matter. 1st) I tried the 2 switchs on theirself successfully, so that I can exclude any kind of problem with them. 2nd) I didn't see any particolar errors on the serial monitor when the Wi-Fi disconnects, but in case this data should be important to better understand, I can collect this data and drop to you. But, I guess this is not the problem root cause. I guess however that the problem is related to the connection between 433Mhz receiver pin and input pin for the NodeMCU. The first one is working to 5V, while for the second the logical is 3.3V. So that a shift level can be used. I don't believe I am the only guy that is facing this type of annoiyng issue! thanks for your appreciated comments. Now I

Sara



**Sagar**

January 30, 2019 at 4:57 am | Reply

Please anyone help me  
I want to programme esp nodemcu using android phone  
Because I don't have laptop or computer



**Sara Santos**

February 5, 2019 at 12:36 pm | Reply

Hi Sagar.  
I've never tried that.  
But there is some information about that here: [cnx-software.com/2016/05/18/programming-esp8266-boards-with-a-smartphone/](http://cnx-software.com/2016/05/18/programming-esp8266-boards-with-a-smartphone/)  
And here: [instructables.com/id/IoT-Development-With-Mobile-Directly/](http://instructables.com/id/IoT-Development-With-Mobile-Directly/)  
I hope this helps.

**Domenico Carvetta**February 8, 2019 at 3:53 pm | [Reply](#)

Sara any answer?

**Ringo Davis**February 12, 2019 at 12:25 am | [Reply](#)

Where do I find RCSwitch.h?

**Sara Santos**February 14, 2019 at 6:53 pm | [Reply](#)

Hi Ringo.

You can find the RCSwitch library here: [github.com/sui77/rc-switch/](https://github.com/sui77/rc-switch/)

**Ender Duman**February 12, 2019 at 8:15 am | [Reply](#)

I like your Alexa ESP32 project. Really nice. Is it possible to change the voice commands? For example "Alexa, turn on hall lamp 1" and "Alexa, turn off hall lamp 1"..  
Regards,

**Rui Santos**February 12, 2019 at 7:40 pm | [Reply](#)

Yes, you need to configure/set your device name in the ESP code.

**Fernando**February 24, 2019 at 10:21 pm | [Reply](#)

without it I get a lot of errors.

**Sara Santos**

February 28, 2019 at 12:06 pm | [Reply](#)

Hi Fernando.  
What do you mean?

**Michele**

March 3, 2019 at 8:56 am | [Reply](#)

Very informative and interesting tutorial. I have found limitations using the Belkin account registration, which in my country requires the inclusion of a product code that I do not have. But then I solved and I still managed to communicate with Alexa. My question is: can you add more channels? maybe without using on all the RF 433 transmitter?  
Thank you.

Name \*

Email \*

Website

Post Comment

- Notify me of follow-up comments by email.
- Notify me of new posts by email.

[Home](#)[Free eBooks](#)[About](#)[Contact](#)[Projects](#)[More ▾](#)[Get Courses ▾](#)