



## Alexa Enabled Garage Door Controller for \$25

By pdubinsky (/member/pdubinsky/) in Microcontrollers (/circuits/microcontrollers/projects/)

3,662

43

4

Featured



Download

Favorite



<https://edub.instructables.com/ET3/II/8V/IWGN7ZBZ/ET3/II/8V/IWGN7ZBZ.LARGE.jpg>

### Why bother:

It's really ironic that one of the easiest home automation devices to control, the home electric garage door opener, eludes the control capabilities of the basic Alexa task. Anyone can add a simple momentary switch to the control terminals of the garage door opener and in 10 minutes, have a simple switch to open and close a garage door.

The problem for Alexa is that while Alexa can turn on or turn off almost any switch or relay type device, she can't pulse a device. Most garage door openers have two terminals that when shorted by a simple push button will cause the garage door to open or close. The door controller assumes that the shorting will only be "momentary" and not permanent. Some will not even operate until the short is removed. Alexa can't really do that.

The answer is to buy a retro fit add on device for your garage door opener that is "Alexa enabled." They install pretty much the same was as any other device in the "Alexa enabled" firmament. They're just quite pricey, anywhere from \$100 to \$300+ which is odd considering that an "Alexa enabled" outlet can be had for less than \$10. Certainly, many of those devices have many more capabilities beyond just opening the door. But if all you want is the ability to open or close a garage door with Alexa, you're stuck. So if you want a simple project that does just that, open or close a garage door, here's mine which will set you back about \$25.

We built our house in the '80s with a network installed during the construction. The first network was coax ethernet which was eventually replaced by twisted pair ethernet which was eventually replaced by WIFI ethernet. My daughter will happily regale the "joys" of being an 8 year old pulling cable through dark and dank crawl spaces under the house. But when we were done, we had a full-fledged network in our home - something quite rare in 1985. And a home with a network was just begging to have a home automation network installed.

In the 1980's, home automation networks were also quite rare and eventually, I settled on an X10 network. It did a few things and it some of those few things well. If all you needed to do was control light bulbs, X10 was a fairly decent solution. As the years passed, I tried various different home automation schemes hoping to expand the capabilities of our network but each ended up having limitations that turned out to be a deal breaker.

Then came Alexa from Amazon and everything changed. Well, actually not much changed. The network built around Alexa was still really only good at controlling a relay but, because Alexa is an Amazon concept, suddenly there was a seeming treasure trove of devices that were "Alexa enabled" right out of the box. "Hello Alexa." "Goodbye X10."

Alexa could find the "Alexa enabled" devices without too much drama. The devices that were "Alexa enabled" usually had an Android or iOS app so that you could control the devices from the app. And you could build routines and then schedule the routines to operate independently. Once the "Alexa enabled" device was installed on the same network as the Alexa hub device like an Echo or Dot (sometimes not a trivial task), Alexa would take care of the networking and messaging. It really is a huge improvement over past attempts at networked home automation.

All of this is fine if all you want to do is control a bought device controller. I wanted to get into the weeds and figure out to do this from scratch. Just how do you make an "Alexa enabled" device?

#### In to the weeds:

When you ask Alexa to do something, say "Alexa turn on the bedroom lights", Alexa forwards the command to Alexa Central where software parses the command looking for the device name, "bedrooms lights", and what to do with the device, "turn on". If Alexa Central can find the device name in the list of devices that it knows reside on your home network and if it understands the command (enunciate clearly), Alexa issues the appropriate command to the named device over the home network and, in this example, the bedroom lights turn on.

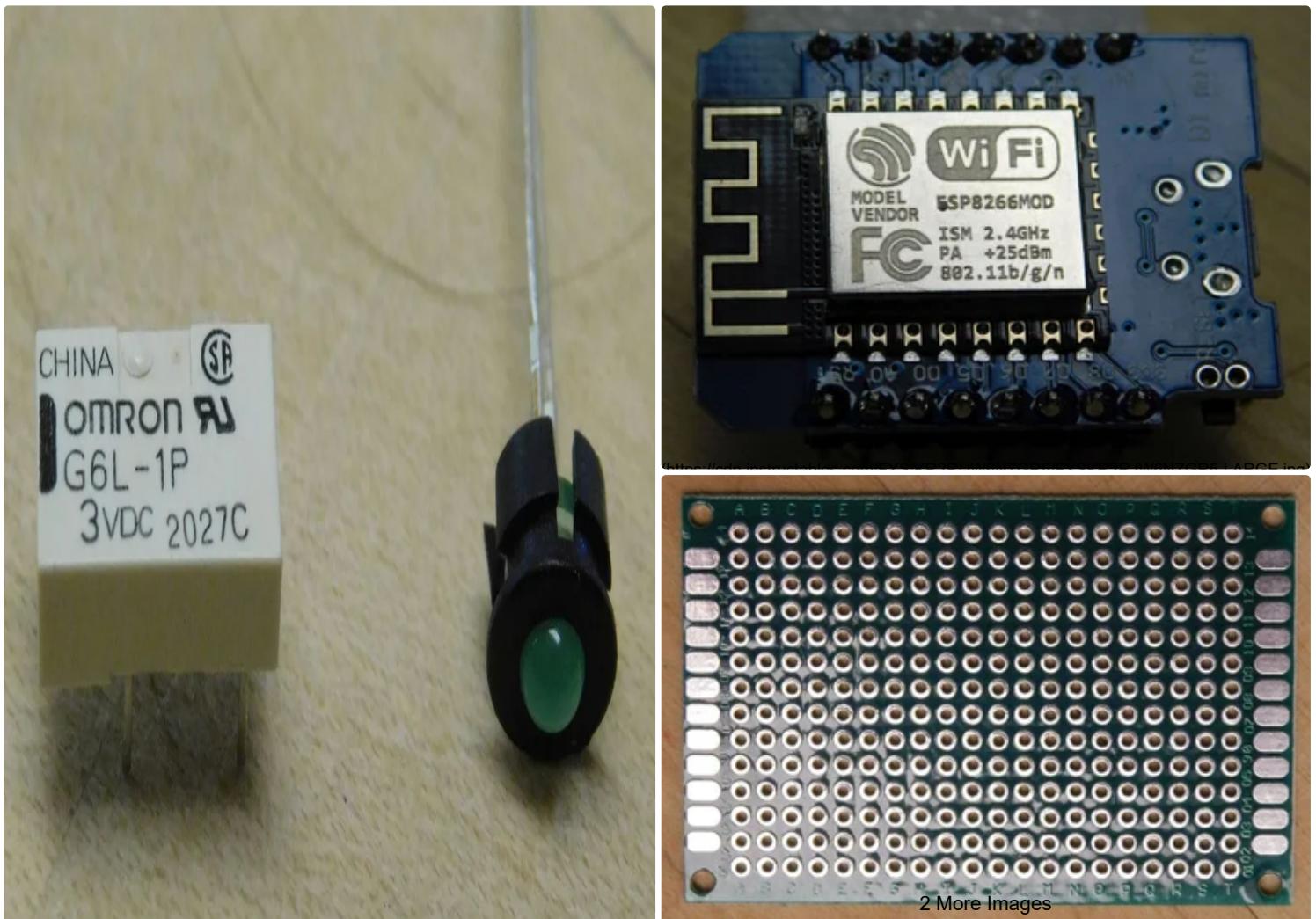
What I wanted to do was create a device that would be recognized by Alexa that I could program to do what I wanted, ie. close a relay contact, run a servo, get data from a sensor, etc. That turns out to be a bit more complicated than I first thought.

The first stop was Google (isn't it always, now?) "How do I control an Alexa enabled device?" That query just returned 18,100,000 results in 0.97 seconds. Yes, there's info out there that can help you do anything with Alexa. The trick is sorting through the 18,000,000+ citations to find the one that I need. Fortunately, I stumbled on a project from [Rui Santos](https://randomnerdtutorials.com) (<https://randomnerdtutorials.com>) that used a ESP8266 NodeMCU microcontroller to control a 5VDC normally open relay. Here's the link to his [project](https://randomnerdtutorials.com/alex Echo with esp32 and esp8266/). (<https://randomnerdtutorials.com/alex Echo with esp32 and esp8266/>) **Plug for Rui:** if you're interested in IoT, have a project idea or just want to learn about small microcontrollers like the NodeMCU, go to his site and sign up for his newsletter. It will be worth your effort.

ESP8266 NodeMCU microcontrollers are particularly well suited for IoT applications because of their extremely low power consumption, small size and very low cost. The NodeMCU used in this project is just about \$5. There are built-in 3.3 VDC GPIO, at least 1 ADC, at least 1 mb of ram (the one in this project has 4 mb of ram), an onboard 5 to 3.3 VDC converter/regulator, PWM capability, I2C and serial UART communication capabilities. That's a lot capability for a \$5 board that can run on a Li-ion battery.

 Add Tip  Ask Question  Comment Download

## Step 1: The Project Parts:



2 More Images

1 - \$5.00 - **D1 Mini NodeMcu Lua ESP8266 ESP-12FF** [Find it](https://www.amazon.com/gp/product/B076F53B6S/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1)

1 - \$3.14 - **3 VDC, SPST-NO Relay** [Find it](https://www.amazon.com/gp/product/B076F53B6S/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1)

1 - \$0.80 - **6 cm X 8 cm perf board** [Find it](https://www.amazon.com/gp/product/B07FK48ZMN/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1)

1 - \$9.99 - **5 VDC Power Supply**[Find it](https://www.amazon.com/CanaKit-Raspberry-Supply-Adapter-Listed/dp/B00MARDJZ4) (https://www.amazon.com/CanaKit-Raspberry-Supply-Adapter-Listed/dp/B00MARDJZ4).

1 - \$0.14 - **LEDFind It** (<https://www.mouser.com/ProductDetail/Lite-On/LTL-5233?qs=sGAEpiMZZMt mwHDZQCdlqefIX0SwY%252BjQ8feB6%252Bmc srg%3D>).

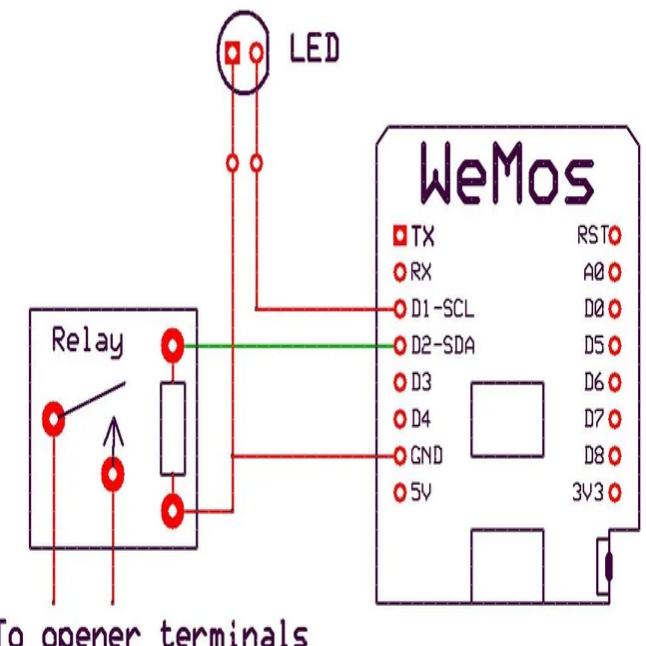
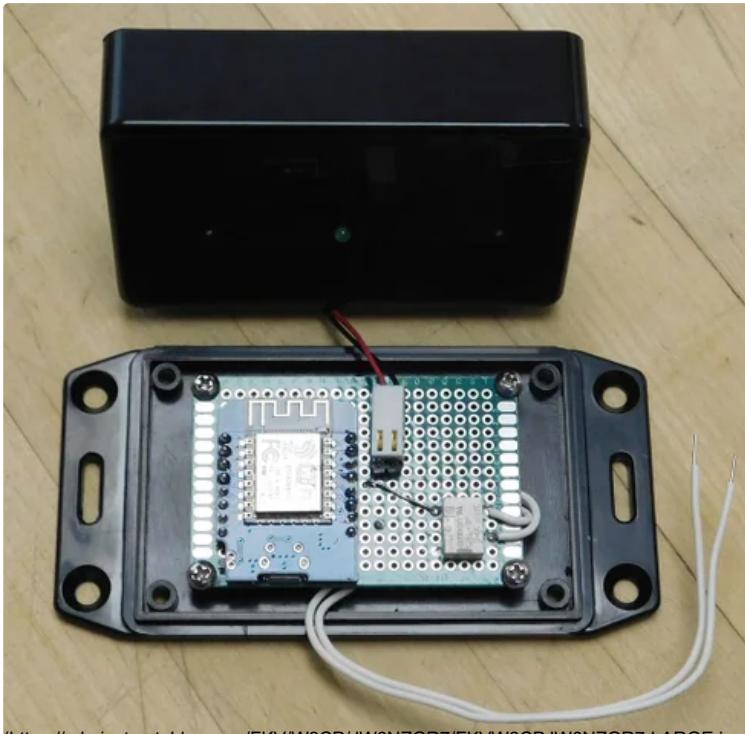
1 - \$0.40 - **LED Mounting Clip**[Find it](https://www.mouser.com/ProductDetail/VCC/CLP_127_BLK?qs=sGAEpiMZZMt lylJihmhKq8Gz6waqu52xhkUc%2Flgd%252B2rb%252BL1tE AIM9g%3D%3D) ([https://www.mouser.com/ProductDetail/VCC/CLP\\_127\\_BLK?qs=sGAEpiMZZMt lylJihmhKq8Gz6waqu52xhkUc%2Flgd%252B2rb%252BL1tE AIM9g%3D%3D](https://www.mouser.com/ProductDetail/VCC/CLP_127_BLK?qs=sGAEpiMZZMt lylJihmhKq8Gz6waqu52xhkUc%2Flgd%252B2rb%252BL1tE AIM9g%3D%3D)).

=====

**\$23.97 - Total**

Add Tip Ask Question Comment [Download](#)

## Step 2: The Build:



(<https://cdn.instructables.com/EKV/W3CB/IW6NZCB7/EKIVW3CB/IW6NZCB7.LARGE.jpg>) (<https://cdn.instructables.com/E7I/C617/IWAV7EDV/E7I6C617/IWAV7EDV1.ARGE.jpg>)

The NodeMCU and the N/O relay are mounted on the perf board to facilitate wringing and mounting in the flanged case.

There are only 3 NodeMCU pins used: D1 (GPIO 5) to control the LED, D2 (GPIO 4) to control the relay and the GND pin for the relay and the LED.

There are 2 pigtails connected to the relay that are connected to the door opener switch terminals.

Power is provided by a standard USB 5VDC power adapter through the NodeMCU onboard microUSB connector.

 Add Tip

 Ask Question

 Comment

Download

Favorite

## Step 3: The Software:

First, your Arduino programmer needs to have the correct ESP8266 libraries added and the correct board selected. Rather than go through that entire procedure here, go to <http://randomnerdtutorials.com> (<http://randomnerdtutorials.com>) and search for the Alexa Controlled Relay IoT project for detailed instructions. I selected the WeMos D1 R1 board for the programmer.

The software that makes this device respond to Alexa is the included fauxmoESP.h module. This is the code that decodes the messages from the Alexa system and reports the commands to the program so that the program can execute code and control the NodeMCU GPIO. The code makes the NodeMCU appear to be a Hue color controller and we will use it to toggle a GPIO pin to pulse the garage door opener.

Before uploading the software, make sure to change "Your\_Network\_SSID" to the correct network SSID for the same network that your Alexa device is attached to. Also, change the value of WIFI\_PASS to the correct password for your network. Then compile and upload to the NodeMCU.

After you upload the software, power cycle the NodeMCU to restart it. If you are still connected to the computer that is running the Arduino programmer, open the serial monitor and you can watch the NodeMCU connect to the network and see the ip address that is assigned by your network's DHCP server.

In order for Alexa to communicate with the NodeMCU, Alexa first has to discover the NodeMCU. Start your Alexa app and tell Alexa to discover devices by saying, "Alexa, discover devices". This can take a while but if everything goes correctly, Alexa should find the NodeMCU device named MINI\_1 and add it to the list of devices that Alexa knows.

Once Alexa finds the device, you can toggle the relay by saying "Alexa, turn on Mini 1". Alexa will respond with "Ok" and the relay contacts on the NO relay should close for the amount of time specified in the pulseTime variable and the LED should light for the same time. There is no need to issue a "turn off" command because the the software will turn off the relay when the pulseTime expires.

**Finally, the code:**

```

/*
 * Thanks to Rui Santos
 *
 * Derived from his Alexa Controlled Relay Project at <a href="http://randomnerdtutorials.com">http://randomnerdtutorials.com</a>
 *
 * This software is published with absolutely no warranty whatsoever.
 * This software simply shows how I did my project and it is not implied expressly
 * or otherwise to suitable in any other application.
 * Use at your own risk.
 */

#define MINI_D1
#include
#include
#define RELAY_PIN_1 4      // Controller Relay D2
#define RELAY_PIN_2 5      // Indicator LED    D1
#define MINI_1 "mini one"
#include "fauxmoESP.h"

#define SERIAL_BAUDRATE 115200
#define WIFI_SSID "Your_Network_SSID"
#define WIFI_PASS "your_wifi_password"

pulseTime = 2000; // Controls Relay Active Time in milliseconds, 2000 = 2 sec
boolean toggle1 = false;

```

[Add Tip](#)[Ask Question](#)[Comment](#)[Download](#)

## Step 4: The Install:



Lastly, at the garage door opener, connect the 2 pigtail leads to the terminals that the opener switch is connected to. If there already are wires attached to the opener terminals, make sure that you connect these wires in parallel with the existing wires.

When first installing the opener, I recommend that you connect the opener to a lap top that has the Arduino programmer software installed. This will allow you to use the programmer's serial monitor to watch the opener connect to your Wifi network. Because a

Alexa Enabled Garage Door Controller for \$25 by pdubinsky (member/pdubinsky) Follow Download Favorite  
garage isn't usually an area that people are concerned about Wifi coverage, it's possible that the opener may have trouble connecting to the network. If that's the case, you'll have to adjust your Wifi network to improve the coverage for the garage opener. Think mesh routers, new and improved routers, bridges, etc.

When everything is working correctly, disconnect the opener from the computer and connect it to the 5 volt power adapter and you should be ready to go.

 Add Tip  Ask Question  Comment Download

## Step 5: Last Thoughts:

In my install, I only use my opener to close the garage door in the evening using a scheduled Alexa routine. The routine runs at 9:00 PM every night and in order to ensure that the routine doesn't open an already closed garage door (the "Alexa enabled" opener doesn't know if the garage door is open or closed), I had to add one more switch that would only be closed when the garage door is open.

The switch is installed in series with one of the pigtail leads so that when Alexa asks the NodeMCU to close the relay contacts, the circuit can only be made if the second switch is also closed - a simple AND gate, if you will. I used a simple momentary leaf switch that is tripped by the door closer carrier that is connected to the garage door.

You can do the same or use a simple magnetic reed switch that is used in common household security systems. I've used both types of switches and both types work well so if you need this type of functionality, go with what you have or like. Enjoy.

PD

 Add Tip  Ask Question  Comment Download

## Share

Did you make this project? Share it with us!

I Made It!

## Recommendations

# Ramen Bowl Lamp



(/id/Ramen-Bowl-Lamp/)

Ramen Bowl Lamp with Yellow LEDs (/id/Ramen-Bowl-Lamp/)

by ramenkingandi (/member/ramenkingandi/) in Circuits (/circuits/projects/)



(/id/VFD-Alarm-Clock/)

VFD Alarm Clock (/id/VFD-Alarm-Clock/)

by ChristineNZ (/member/ChristineNZ/) in Arduino (/circuits/arduino/projects/)



BY  
FRANCO  
MOLINA

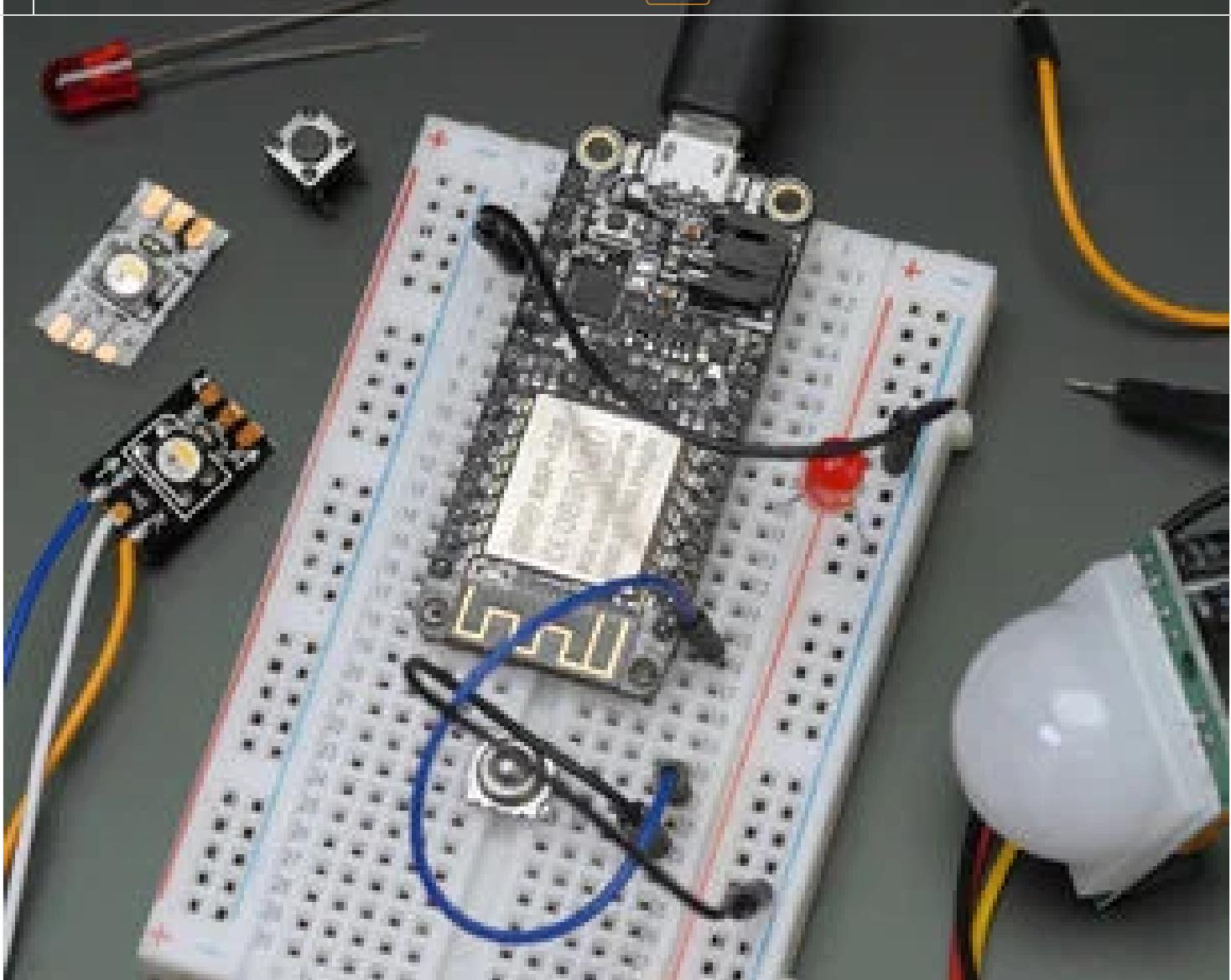
# DRUMCUBE

## A ROBOT DRUMMER

(/id/DrumCube-an-Arduino-Robot-Drummer/)

DrumCube, an Arduino Robot Drummer (/id/DrumCube-an-Arduino-Robot-Drummer/)

by FrancoMolina (/member/FrancoMolina/) in Art (/circuits/art/projects/)



(/class/Internet-of-Things-Class/)

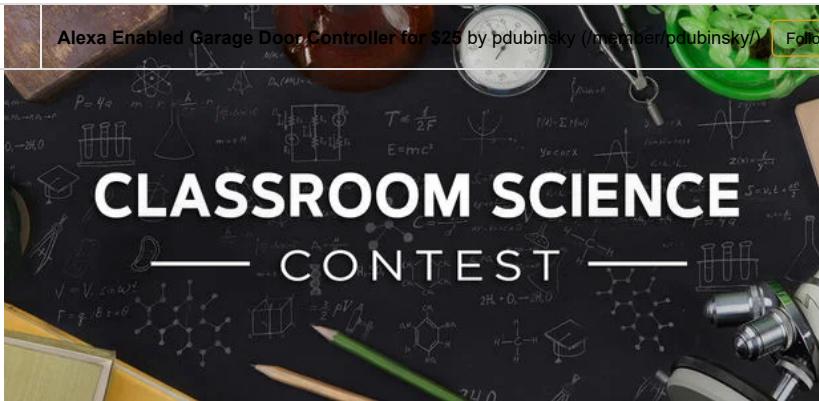


Internet of Things Class (/id/Internet-of-Things-Class/)

19,510 Enrolled



(/contest/frozentreats2019/)



Alexa Enabled Garage Door Controller for \$25 by pdubinsky (/member/pdubinsky/)

Follow

Download

Favorite

# CLASSROOM SCIENCE — CONTEST —

(/contest/classroomscience/)



# BEAUTY TIPS — CONTEST —

(/contest/beautytips/)



 Add Tip

 Ask Question

 Post Comment

We have a **be nice** policy.  
Please be positive and constructive.

[Add Images](#) [Post](#)

## 4 Discussions



(/member/gkern/) gkern (/member/gkern/) 7 hours ago

Reply

▲ Upvote

Or buy this Ewelink switch for 20 bucks, on Amazon, google and Amazon Echo. MHCOZY WiFi Momentary Inching Self Locking Wireless Smart Switch Relay Module for Smart Home 5V 5V/12V, Be Applied to Garage Door Opener



(/member/PhantasticOne/) PhantasticOne (/member/PhantasticOne/) Question 21 hours ago on Step 5

Answer

▲ Upvote

I'm new to this so pardon this ridiculous question.

I have my home set up with Google Assistant. Is there any way to program this to work with it?

1 answer ▾

Reply

▲ Upvote

Your relay link in the parts list is a duplicate of the **Mini NodeMcu Lua** instead of a 3V relay. All the other links seem correct. I'd like to find \$3 3V relays that can be driven with 20mA (max sink of 8266) to simplify these projects. There is no kickback diode in your circuit. Is it an SSR (solid state relay) so it doesn't need one? I've been using 5V relays and switching the 5V with a transistor. Also, I like the box you've picked. I'll be using those!

Your solution to the problem of only closing the door and not opening an already closed door is inspired. Thanks for the great ideas!

---

[Post Comment](#)

## Categories

-  Circuits (</circuits/>)
-  Living (</living/>)
-  Workshop (</workshop/>)
-  Outside (</outside/>)
-  Craft (</craft/>)
-  Teachers (</teachers/>)
-  Cooking (</cooking/>)

## About Us

- [Who We Are](#) (</about/>)
- [Why Publish?](#) (</create/>)
- [Jobs](#) (</community/Positions-available-at-Instructables/>)

## Resources

- [Sitemap](#) (</sitemap/>)
- [Help](#) (</id/how-to-write-a-great-instructable/>)
- [Contact](#) (</contact/>)



(<https://www.instagram.com/instructables/>)



(<https://www.pinterest.com/instructables>)



(<https://www.facebook.com/instructables>)



(<https://www.twitter.com/instructables>)

---

© 2019 Autodesk, Inc. [Terms of Service](#) (<http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21959721>)

[Privacy Statement](#) (<http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21292079>)

[Privacy settings](#)

[Legal Notices & Trademarks](#) (<http://usa.autodesk.com/legal-notices-trademarks/>)

[Autodesk, Inc.](#) (<http://www.autodesk.com>)