Home (/blog) Bazaar (http://seeedstudio.com) Fusion (https://www.seeedstudio.com/fusion.html) Wiki (http://wiki.seeedstudio.com)

Projects (https://project.seeedstudio.com/) Forum (http://Forum.seeedstudio.com) Solution (https://solution.seeed.cc/) About Us (/blog/forum-2/about-us-2/)

LATEST OPEN TECH FROM SEEED STUDIO (/BLOG/)

FOR EMERGING IOT, AI AND AUTONOMOUS APPLICATIONS ON THE EDGE

LATEST (/BLOG)   GROVE (/BLOG/TAG/GROVE/)   PCB (/BLOG/TAG/PCB/)   RASPBERRY PI (/BLOG/TAG/RASPBERRY-PI/)   ARDUINO (/BLOG/TAG/ARDUINO/)

IOT (/BLOG/TAG/IOT/)   NEW PRODUCT (/BLOG/TAG/NEW-PRODUCT/)   AI (/BLOG/TAG/AI/)   SHENZHEN (/BLOG/TAG/SHENZHEN/)

RESPEAKER (/BLOG/TAG/RESPEAKER/)   LORA (/BLOG/TAG/LORA/)

ARDUINO (/BLOG/CATEGORY/ARDUINO/)   BAZAAR (/BLOG/CATEGORY/BAZAAR/)   GUIDE (/BLOG/CATEGORY/GUIDE/)

Dummies guide on driving a 28BYJ-48 Stepper Motor with a ULN2003 Driver Board and Arduino

By Helen (/blog/author/helen/)   1 year ago

SALE on Arduino Motor Pack (https://www.seeedstudio.com/Motor-Pack-for-Arduino-p-4000.html?utm_source=blog&utm_medium=blog)

Before we start, Seeed is having a big sale on our very own Arduino Motor Pack! This pack is a perfect kit for motor learning with Arduino. Whether your project requires a DC motor, a stepper motor, or a steering gear, it can be found in this kit. This kit includes:

Seeed Fusion Open Parts Library for PCBA

(https://www.seeedstudio.com/opl.html)

Subscribe for Updates

Email address:

Your email address

SUBSCRIBE NOW

(https://...)
JaRWXc8cBc1gm5kK

(https://...)
&text=seeedstudio)

Recent Posts

Linux 5.7 Released: New features and Main Changes (/blog/2020/06/13/linux-5-7-released-new-features-and-main-changes/)

What Is A Doppler Radar And How Does It Work? (/blog/2020/06/12/what-is-a-

- 6V DC motor(13000 rpm ±14%) x2
- Small stepper motor x1
- Standard servo x1(Speed: 0.17/60°@4.8V；0.14/60°@6.0V）/(Torque: 3.5kg.cm@4.8V; 4.8kg.cm@6.0V)
- Micro servo x1(Speed: 0.12/60°@4.8V；0.10/60°@6.0V）/(Torque: 1.5kg.cm@4.8V; 1.8kg.cm@6.0V)

Interested? Use code: **MOTOR40** to get 40 percent off our Arduino motor pack (https://www.seeedstudio.com/Motor-Pack-for-Arduino-p-4000.html? utm_source=blog&utm_medium=blog) now!

Without further ado, let us jump right into Dummies guide on driving a 28BYJ-48 Stepper Motor with a ULN2003 Driver Board and Arduino!
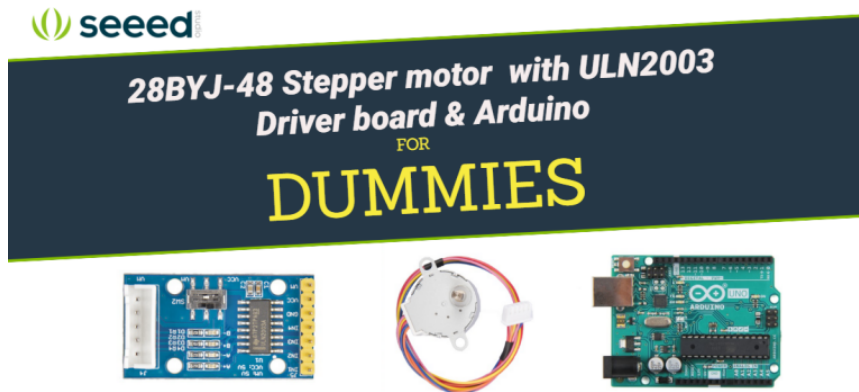
The 28BYJ-48 stepper motor is widely used to control a myriad of common devices we see every day. From blinds, car side mirror tilts and DVD players to security cameras and precise control machines, stepper motors are closer to us than we actually think.

Ironically, not many humans in our world actually know how to work with them. And if you are a totally clueless beginner when it comes to electronics, we understand that reading through tons of technical jargon online can be rather daunting and off-putting at times. Hence, we have tried our hand at simplifying that information in this summary for you, on how to interface a 28BYJ-48 stepper motor with an Arduino, using a ULN2003 driver board.

First of all, let's start with an introduction on the 28BYJ-48 stepper motor and the ULN2003 driver board.
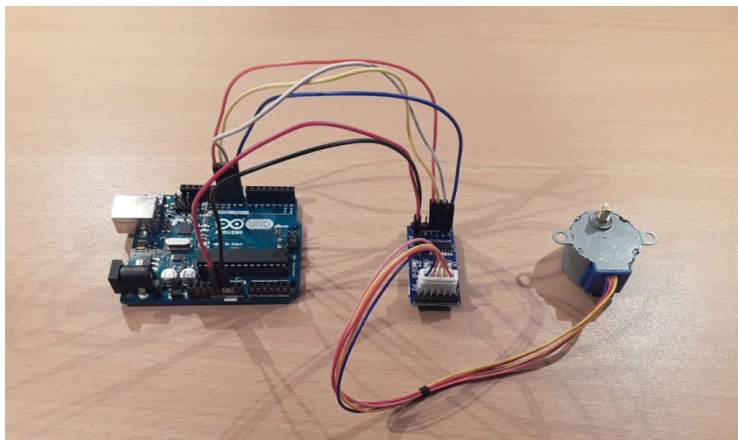
A 28BYJ-48 stepper motor connected to a ULN2003 driver board and an Arduino (Source: Mschoeffler)

The 28BYJ-48 Stepper Motor

Latest Posts

Tags

What is the 28BYJ-48 Stepper motor?

The 28BYJ-48 stepper motor is a commonly used stepper motor, which converts electrical pulses into discrete mechanical rotation. Why is the name stepper used, you may ask. That is because when electrical signals are applied, the stepper motor rotates in accurate and fixed angle increments known as steps. The motor consists of 4 coils that make a ring around the rotor. These coils are known as the stator, as they are stationary and static. Each coil is rated at +5V, making it easy to control with any microcontroller, such as an Arduino.

The GIF below illustrates this working property.
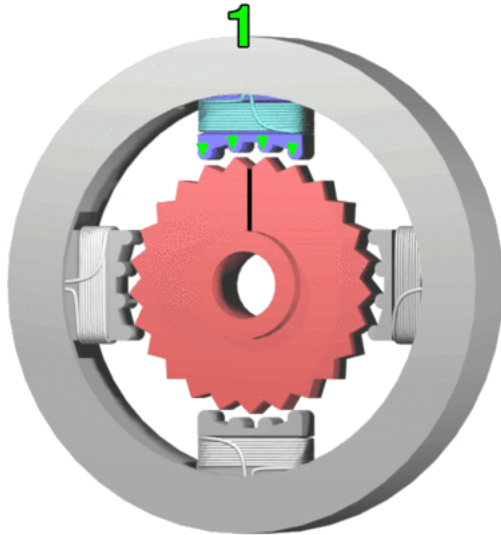
What is the difference between stepper motors and standard DC motors?

Not sure if the 28BYJ-48 stepper motor is the motor that you need? Before deciding on which type of motor is more suitable for your project, we will have to first understand some differences between the two motors. Here are some basic differences between the two.

1. The rotation of stepper motors is incremental, slow and precise, while DC motors have a fast, continuous motion.

2. Stepper motors are known to generate some noise during operation while DC motors are quiet and relatively vibration free.

3. The response time of the stepper motor is slower than the DC motor.

4. Stepper motors can be easily controlled with microprocessors like the Arduino. Compared to DC motors, they are more mechanically simple and easy to design and build. In contrast, DC motors are not so easily controlled with microprocessors

What is the difference between stepper motors and servo motors?

Perhaps you may have heard of the servo motors, another motor similar to the stepper motor. It can be a little difficult to choose between them as there are so many considerations to look at: speed, cost, circuitry and more. So here's a brief description on how they're different.

1. Stepper motors can move more accurately and precisely than the servo motor, and are much more easier to control.

2. Stepper motors are more suitable for applications with lower speeds of less than 2000 rpm (revolutions per minute), and servos are best suited for applications with high speeds greater than 2000 rpm.

3. Stepper motors operate at lower speeds than servo motors.

4. Stepper motors are cheaper than servo motors as they are less mechanically complex.

When would I choose a stepper motor over the other types?

Although the stepper motor does have its disadvantages when compared against the other two, as long as its performance satisfies the needs of your device, you should consider the stepper motor for it offers several advantages over the servo motor and the standard DC motor. Here are some areas where the stepper motor triumphs over the other two.

- **Precise Positioning** – Stepper motors move in precise steps. As such, they do well in applications that require precise positioning, such as 3D printers and camera platforms.

- **Precise Speed control** – Precise increments in movements enable excellent control of rotational speed for process automation and robotics.

- **High Torque at low speeds** – Stepper motors are best suited for applications with low speed (less than 2000 rpm) as they have maximum torque at low speeds. In contrast, normal DC motors and servo motors do not have so much torque at low speeds.

- **Easy to control** – Stepper motors can easily be controlled using a microcontroller like an Arduino. In fact, its ease of use has also been a major reason for its continued usage by stepper motor users.

- **Cheap** – In applications where stepper motors would suffice, and using one could help you to realize better cost savings.
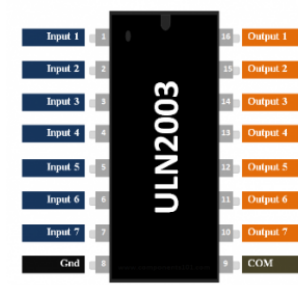
In addition, the stepper motor also shares some of the same advantages as the other types of motors. For example, similar to DC motors, stepper motors are widely available and come in many different sizes and styles, which means it can accommodate for size constraints. This means that you do not have to compromise on the advantages of the other motors when you choose to use a stepper motor.

The ULN2003 Driver Board

What is the ULN2003 Driver Board?

The ULN2003 is one of the most common motor driver ICs that houses an array of 7 Darlington transistor pairs, each capable of driving loads up to 500mA and 50V. Basically, a Darlington pair is a pair of transistors, where the second transistor amplifies the output current of the first transistor. The ULN2003 IC is needed to drive the motor with an Arduino, but they come in convenient and cheap driver boards that are readily available complete with indicator LEDs. So it's better to grab one of these rather than breadboarding the IC itself.

As shown in the diagram below, a ULN2003 driver board consists of a ULN2003 soldered onto a board, together with resistors, capacitors, and other bits and bobs that help create the circuit that takes the pulse signals from the controller and converts them into stepper motor motion.

Pinout of ULN2003
(Source: Component101)

A ULN2003 driver board
(Source: electronicoaldas)

Why is the ULN2003 needed?

The 28BYJ-48 stepper motor consumes high current and hence, we will need to use a driver IC like the ULN2003 in order to control the motor with a microcontroller like the Arduino. Known for its high current and high voltage capacity, the ULN2003 gives a higher current gain than a single transistor and enables the low voltage and low current output of a microcontroller to drive a higher current stepper motor. For example, a stepper motor that needs 9V and 300mA to operate cannot be powered by an Arduino. Hence, we connect this IC to source for enough current and voltage for the motor. If you have to power anything more than 5V and 80mA, the ULN2003 driver board should be used.

Take note that powering the stepper motor directly from the Arduino is not recommended, which will be explained in the following section

Pin Arrangement
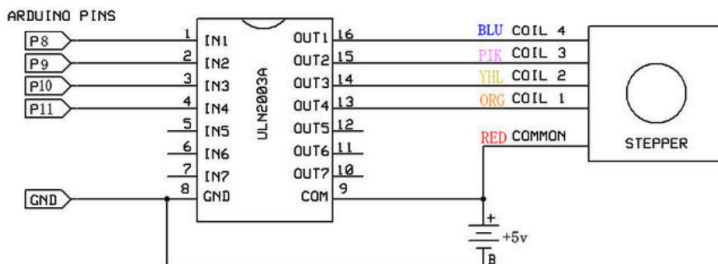
Schematic of the pin arrangement
(Source: Geeetech wiki)

Connecting the 28BYJ-48 stepper motor to the ULN2003 driver board

Usually, the 28BYJ-48 stepper motor comes with a 5-pin connector that will fit to the ULN2003 driver board

Connecting the ULN2003 driver board to the Arduino

Connect the ULN2003 driver IN1, IN2, IN3, IN4 to the Arduino digital pins 8, 9, 10, and 11 respectively. The driver board has two pins which are labeled GND and VCC, which are two pins for the power supply. The ULN2003 driver board's GND pin must be connected to the Arduino's GND pin. Similarly, the driver board's VCC pin should be connected to the Arduino's 5V pin.

Important note: Using this setup, we are powering the stepper motor directly from the Arduino. Although this is the simplest way to provide power to the motor, it is not recommended to do so (connecting the driver board VCC pin to the 5V pin of the Arduino). This is because if the motor

draws too much power, your Arduino can potentially be damaged. Make sure not more than 300mA is drawn out of your Arduino. If more power is needed, simply connect your driver board to an external voltage supply rather than using the Arduino's onboard power supply.

But since we are at the initial learning stage of familiarizing ourselves with the setup, we will be powering the motor from the Arduino in this example to keep the connections simple. Once you have familiarized yourself with the connections, be sure to connect your motor to an external power source instead if more power is needed.

Example Code

The Arduino Integrated Development Environment (IDE), or Arduino software, comes pre-installed with a default stepper library. Hence, there is no need to download the library. On the first line of the code, we will first include the header file of this built-in stepper library.

We define the number of steps that will be made for one revolution. Working out this number might be a little tricky, so here's how it's done. Stepper motors can be driven in different modes and they have a specific gear ratio. Both factors have an influence on the number of steps per revolution. For this example, we shall drive the motor in a mode known as the full step mode, with each step corresponding to a rotation of 11.25 degrees according to the datasheet. That means there are 32 steps per revolution (360/11.25 = 32). In addition, the manufacturer has specified a gear ratio of 64:1 for the 28BYJ-48 stepper motor. To obtain the final number of steps, the gear ratio must be multiplied by the number of steps per revolution, 32. The more accurate value for the gear ratio is actually about 63.68395 : 1. Therefore, we set the final number of steps to 2038 (32 x 63.68395 = 2037.8864).

Next, we initialize the stepper. Within the brackets are the parameters. The first parameter is the number of steps. As it was already defined in the previous line, we can write STEPS to represent this number. The other parameters correspond to the Arduino pins that are used to connect the ULN2003 driver board. As stated previously, the pins are 8, 9, 10 and 11.

The next loop function is where we will write our code to drive the motor. For the first line of the loop, we set the speed of one revolution to one per minute. Next, we tell the stepper motor to do 2038 steps. Since one revolution corresponds to 2038 steps as we have calculated previously, that means the motor shaft should move a full revolution within about one minute. Next, we set a delay of one second. In the next two lines, we do the same thing again – setting the number of revolutions per minute and commanding the stepper to do a number of steps. However, this time, we set the speed to 6 rounds per minute and move the shaft in the other direction by setting a negative number of steps. This means that the motor will move 6 times faster, and should complete a full revolution in about 10s (60s / 6 = 10s).

Here's the example code. Upload the following code into your Arduino programme and try it out yourself!

```
1.    #include <Stepper.h>
2.
3.    #define STEPS 2038 // the number of steps in one revolution of your motor
(28BYJ-48)
4.
5.    Stepper stepper(STEPS, 8, 10, 9, 11);
6.
7.    void setup() {
8.    // nothing to do
9.    }
10.
11.    void loop() {
12.    stepper.setSpeed(1); // 1 rpm
13.    stepper.step(2038); // do 2038 steps -- corresponds to one revolution in
one minute
14.    delay(1000); // wait for one second
15.    stepper.setSpeed(6); // 6 rpm
16.    stepper.step(-2038); // do 2038 steps in the other direction with faster
speed -- corresponds to one revolution in 10 seconds
17.    }
```

Code by: Michael Schoeffler

This is merely some simple code to get you started on getting the stepper motor up and running. Starting out in the electronics arena can be pretty daunting and at times can fill you with feelings of doubt and inadequacy, especially when your code doesn't run. So start with a few simple codes. After some more practice to build up your confidence, be sure to try out more complex codes so that you can get the motor to do more things. Github, a platform with millions of open-source projects, is a good place to start looking for more example codes to play around with.

Now that you have gotten a gist of how to operate the 28BYJ-48 stepper motor with the ULN2003 driver board and Arduino, have a browse at the Seeed Bazaar here (https://www.seeedstudio.com/catalogsearch/result/?cat=&q=stepper+motor), where you can find quality stepper motors, driver boards, and Arduino microcontrollers for your next project. Also, check out the stepper motor driver pack (https://www.seeedstudio.com/Gear-Stepper-Motor-Driver-Pack-p-3200.html) that comes with both the 28BYJ-48 stepper motor and a ULN2003 driver board at affordable prices.

So that's all for this guide! Feel free to leave any questions in the comments section, and we'll do our best to get back to them.

Please follow and like us:

Follow (http://www.specificfeeds.com/widgets/emailSubscribeEncFeed/TTlmd1VuNFVUSHhm TGJTdWwzUDR0eXh6TmFPQlZXUEwvWHNrbENTeUpJS1ZTUlpWaWpoc0hDMW9rVUJxQ0M1M2Y xY1pWdU82SGpzRDdTbkJHakZSL2NuUG0rb05RRlpVQ0lyMUd4NFBxVG1GRStFOVVYZGlrbFljL3ZC VmJ3NGR8VFBmaWJiRjFIMlF0OXlYM0dFcTl0ci94NmwzOE9Nalh1amZKVkdwL1QvND0=/OA==)

Like (https://www.facebook.com/sharer/sharer.php?u=%2Fblog%2F2019%2F03%2F04%2Fdriving-a-28byj-48-stepper-motor-with-a-uln2003-driver-board-and-arduino%2F)

Tweet (https://twitter.com/intent/tweet?text=Dummies+guide+on+driving+a+28BYJ-48+Stepper+Motor+with+a+ULN2003+Driver+Board+and+Arduino /blog/2019/03/04/driving-a-28byj-48-stepper-motor-with-a-uln2003-driver-board-and-arduino/)

Save

TAGS: 28BJY-48 STEPPER MOTOR (/BLOG/TAG/28BJY-48-STEPPER-MOTOR/), ARDUINO (/BLOG/TAG/ARDUINO/), STEPPER MOTOR DRIVER BOARDS (/BLOG/TAG/STEPPER-MOTOR-DRIVER-BOARDS/), STEPPER MOTORS (/BLOG/TAG/STEPPER-MOTORS/), ULN2003 (/BLOG/TAG/ULN2003/), ULN2003 DRIVER BOARD (/BLOG/TAG/ULN2003-DRIVER-BOARD/)

PREVIOUS   NEXT

5 THOUGHTS ON "DUMMIES GUIDE ON DRIVING A 28BYJ-48 STEPPER MOTOR WITH A ULN2003 DRIVER BOARD AND ARDUINO"

jayson says:

December 19, 2019 at 3:54 pm (/blog/2019/03/04/driving-a-28byj-48-stepper-motor-with-a-uln2003-driver-board-and-arduino/#comment-245220)

Thanks for explaining why darlington transistors are needed.

j says:

November 18, 2019 at 5:07 am (/blog/2019/03/04/driving-a-28byj-48-stepper-motor-with-a-uln2003-driver-board-and-arduino/#comment-239373)

Flip IN2 and IN3 to enable reverse.

Ivan says:

October 5, 2019 at 9:53 pm (/blog/2019/03/04/driving-a-28byj-48-stepper-motor-with-a-uln2003-driver-board-and-arduino/#comment-231231)

Hi Helen.
Thanks for the blog.
I am new to stepper motors.
I built up the same setup like you with the ULN2003 board, and using the Stepper library.
As long as I dont call stepper.step(), the 28BYJ-48 consumes practically no current.
But if I call send(), after the motor stops rotating, the power consumption jumps up to constant 370mA on 5V.
Is this a problem with the Stepper driver?
Thanks,
Ivan

Chris says:

June 2, 2019 at 9:16 pm (/blog/2019/03/04/driving-a-28byj-48-stepper-motor-with-a-uln2003-driver-board-and-arduino/#comment-214958)

Hello Helen
I am very inexperienced with electronics, I been looking this project and the one part that is missing is how to power the project.
In this project I am guessing for power do I use a USB computer to Arduino?
To power th 28BYJ-48 stepper motor through the ULN2003 driver board, can I use a battery source?

I am not grasping how to Powering projects do you have a guides on it?

Helen says:

June 14, 2019 at 5:31 pm (/blog/2019/03/04/driving-a-28byj-48-stepper-motor-with-a-uln2003-driver-board-and-arduino/#comment-215855)

Hello Chris

Indeed, you can power the Arduino by connecting the USB port of the Arduino to your laptop or desktop. For more portability, you can also power it using a battery. Try connecting a 9V battery, with the positive terminal to the Vin pin and the negative terminal to the GND pin of the Arduino. Also, you can connect an AC to DC adaptor to the barrel connector of the Arduino.

There is another way of powering the Arduino with the battery using the 5V and GND pins. But, this method is generally avoided because the 5V pin bypasses the voltage regulator that is built into the board. So if your voltage exceeds 5V, your Arduino can potentially be damaged.

Hope this helps! Do tell us if it works out!

Comments are closed.

Tags

AI (19) (/BLOG/TAG/AI/)   ARDUINO (110) (/BLOG/TAG/ARDUINO/)   BAZAAR (18) (/BLOG/TAG/BAZAAR/)   COMMUNITY (51) (/BLOG/TAG/COMMUNITY/)

CONTEST (13) (/BLOG/TAG/CONTEST/)   DIY (17) (/BLOG/TAG/DIY/)   DSO (18) (/BLOG/TAG/DSO/)   DSO NANO (18) (/BLOG/TAG/DSO-NANO/)

EDGE COMPUTING (14) (/BLOG/TAG/EDGE-COMPUTING/)   FUSION (24) (/BLOG/TAG/FUSION/)   GROVE (115) (/BLOG/TAG/GROVE/)

GROVE SENSOR (17) (/BLOG/TAG/GROVE-SENSOR/)   IIOT (14) (/BLOG/TAG/IIOT/)   IOT (52) (/BLOG/TAG/IOT/)   LORA (15) (/BLOG/TAG/LORA/)

MACHINE LEARNING (16) (/BLOG/TAG/MACHINE-LEARNING/)   MAKER (34) (/BLOG/TAG/MAKER/)   MAKER FAIRE (19) (/BLOG/TAG/MAKER-FAIRE/)

NEW ARRIVAL (14) (/BLOG/TAG/NEW-ARRIVAL/)   NEW PRODUCT (26) (/BLOG/TAG/NEW-PRODUCT/)   NPP (39) (/BLOG/TAG/NPP/)

NVIDIA JETSON NANO (14) (/BLOG/TAG/NVIDIA-JETSON-NANO/)   ODYSSEY - X86J4105 (23) (/BLOG/TAG/ODYSSEY-X86J4105/)

OPEN HARDWARE (46) (/BLOG/TAG/OPEN-HARDWARE/)   OPEN SOURCE HARDWARE (24) (/BLOG/TAG/OPEN-SOURCE-HARDWARE/)

OSCILLOSCOPE (22) (/BLOG/TAG/OSCILLOSCOPE/)   PCB (129) (/BLOG/TAG/PCB/)   PCB DFM (15) (/BLOG/TAG/PCB-DFM/)   PROJECT (17) (/BLOG/TAG/PROJECT/)

PROTOTYPE (17) (/BLOG/TAG/PROTOTYPE/)   PROTOTYPING (13) (/BLOG/TAG/PROTOTYPING/)   RAINBOW CUBE (17) (/BLOG/TAG/RAINBOW-CUBE/)

RAINBOWDUINO (21) (/BLOG/TAG/RAINBOWDUINO/)   RASPBERRY PI (73) (/BLOG/TAG/RASPBERRY-PI/)   RASPBERRY PI 4 (33) (/BLOG/TAG/RASPBERRY-PI-4/)

RASPBERRY PI 4 MODEL B (16) (/BLOG/TAG/RASPBERRY-PI-4-MODEL-B/)   SEEEDUINO (69) (/BLOG/TAG/SEEEDUINO/)   SENSECAP (18) (/BLOG/TAG/SENSECAP/)

SENSOR (19) (/BLOG/TAG/SENSOR/)   SHENZHEN (16) (/BLOG/TAG/SHENZHEN/)   SHIELD (28) (/BLOG/TAG/SHIELD/)   STEAM (19) (/BLOG/TAG/STEAM/)

STEM EDUCATION (13) (/BLOG/TAG/STEM-EDUCATION/)   WIO TERMINAL (28) (/BLOG/TAG/WIO-TERMINAL/)   XADOW (15) (/BLOG/TAG/XADOW/)

Categories

Select Category ▾

Recent Posts

Linux 5.7 Released: New features and Main Changes (/blog/2020/06/13/linux-5-7-released-new-features-and-main-changes/)

What Is A Doppler Radar And How Does It Work? (/blog/2020/06/12/what-is-a-doppler-radar-and-how-does-it-work-m/)

The Biggest Ever Open Parts Library! Over 150,000 parts for Seeed Fusion PCBA (/blog/2020/06/12/the-biggest-ever-open-parts-library-over-150000-parts-for-seeed-fusion-pcba/)

Newsletter from Seeedstudio

Email address:

Your email address

SUBSCRIBE NOW

Seeed Fusion Open Parts Library for PCBA

(https://www.seeedstudio.com/opl.html)