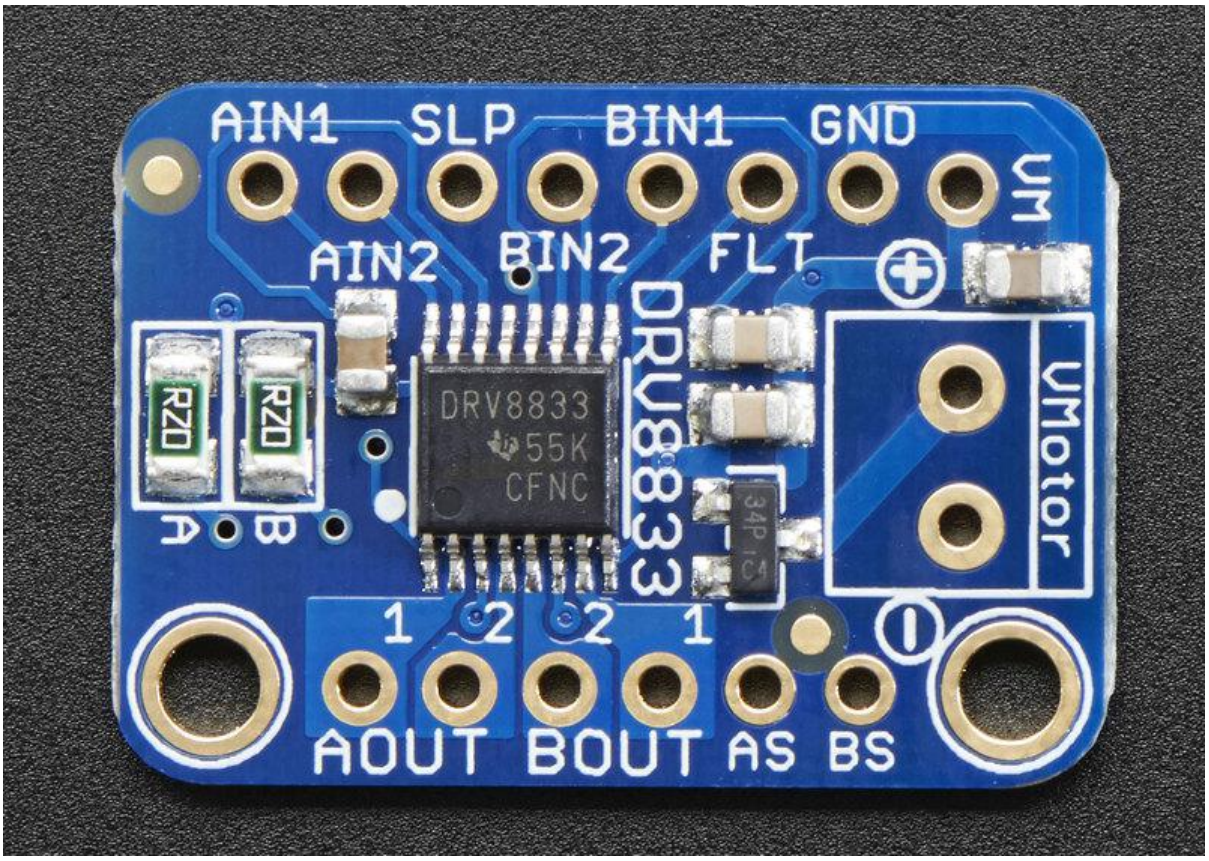
Adafruit DRV8833 DC/Stepper Motor Driver Breakout Board

Created by lady ada

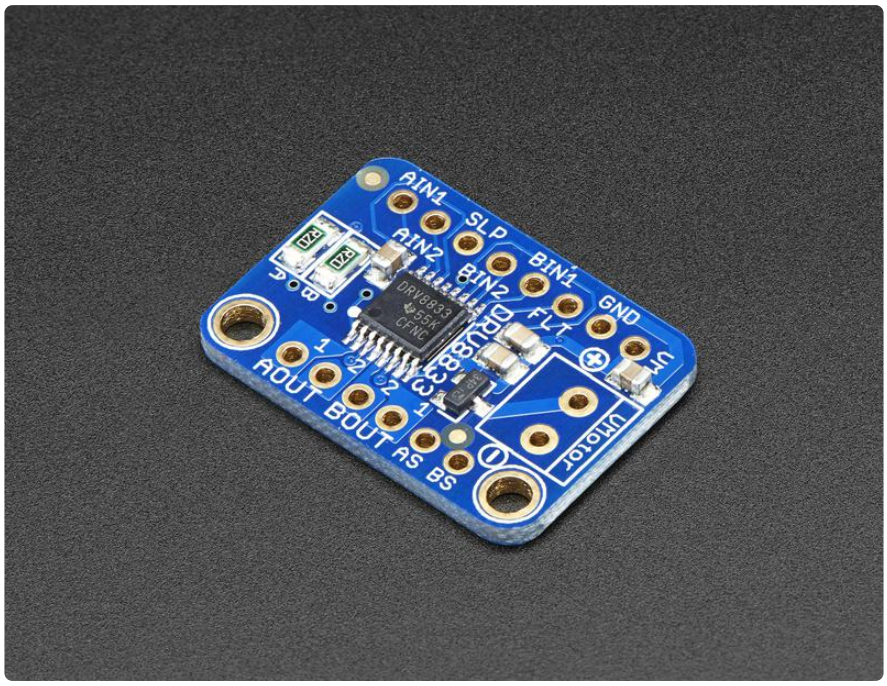


https://learn.adafruit.com/adafruit-drv8833-dc-stepper-motor-driver-breakout-board

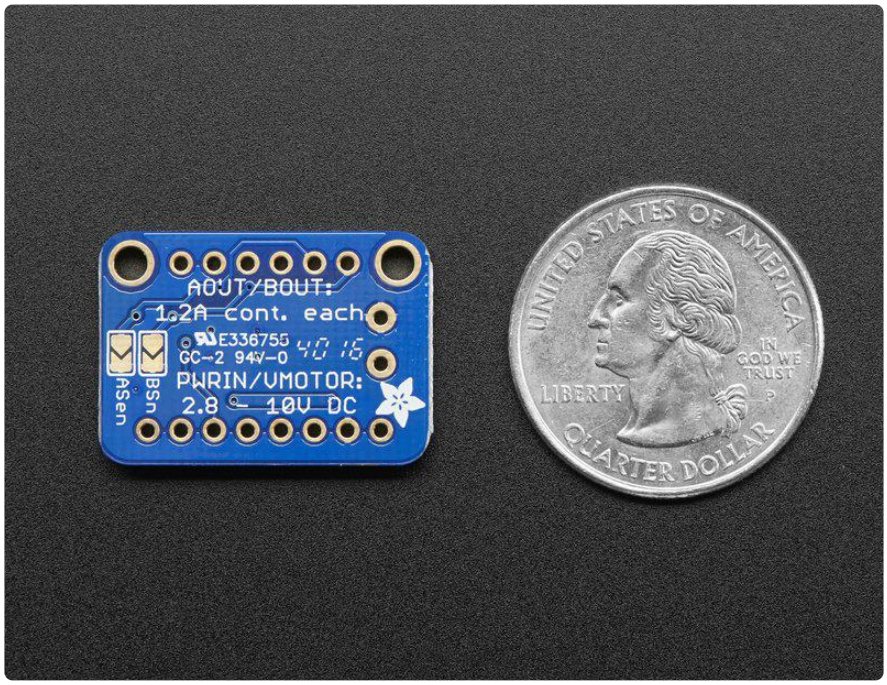
Last updated on 2021-11-15 06:48:50 PM EST

Table of Contents

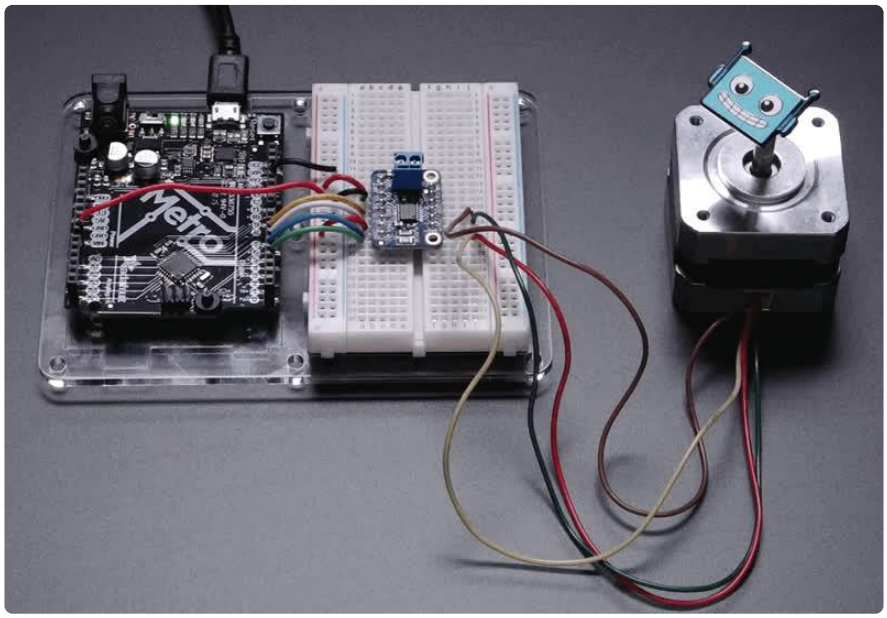
Overview



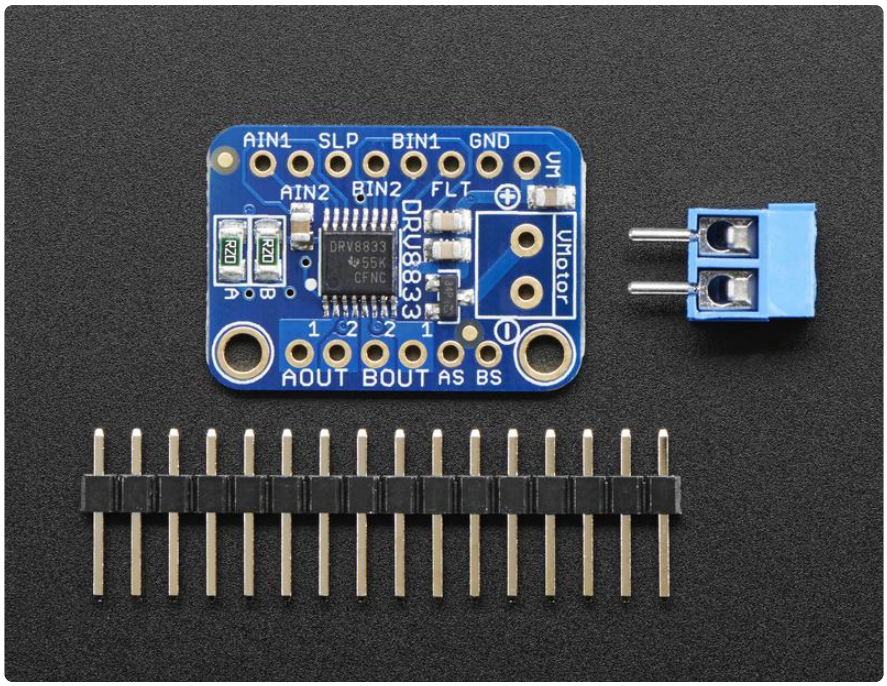
Spin two DC motors or step one bi-polar or uni-polar stepper with up to 1.2A per channel using the DRV8833. This motor driver chip is a nice alternative to the TB6612 driver. Like that chip, you get 2 full H-bridges, but this chip is better for low voltage uses (can run from 2.7V up to 10.8V motor power) and has built in current limiting capability. We set it up for 1A current limiting so you don't get more than 2A per chip, but you can also disable the current limiting, or change it to a different limit!



We solder on DRV8833 onto a breakout board for you here, with a polarity protection FET on the motor voltage input. Each breakout chip contains two full H-bridges (four half H-bridges). That means you can drive two DC motors bi-directionally, or one stepper motor. Just make sure they're good for about 1.2 Amp or less of current, since that's the limit of this chip. They do handle a peak of 2A but that's just for a short amount of time, if you turn off the current limiting we set. What we like most about this particular driver is that it comes with built in kick-back diodes internally so you dont have to worry about the inductive kick damaging your project or driver! You also don't have to worry as much about burning out the chip with overdriving since there is current limiting.

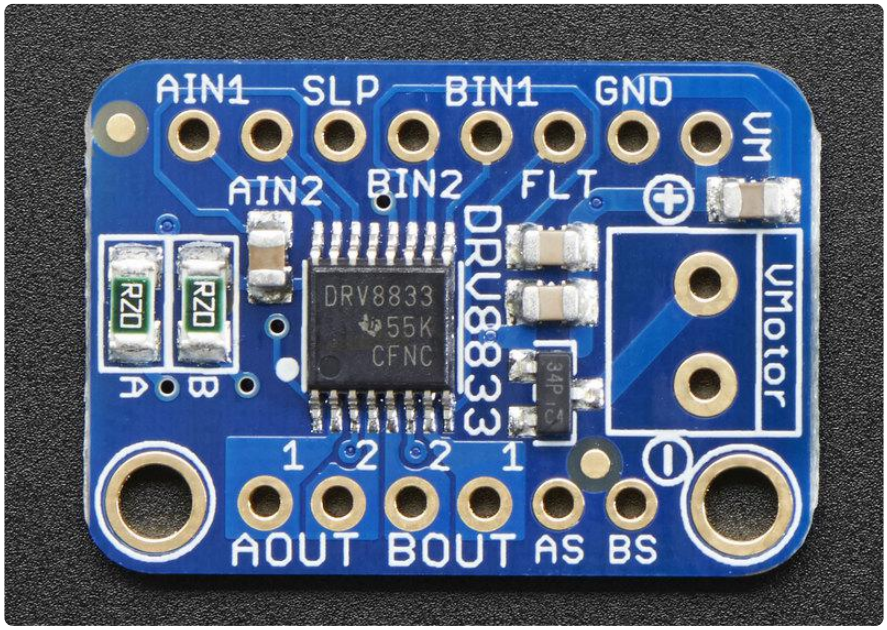


There's two digital inputs per H-bridge (one for each half of the bridge), you can PWM one of the inputs to control motor speed. Runs at 2.7V-10.8V logic/motor power. The motor voltage is the same as the logic voltage, but logic voltage from 2.7V or greater will work so no need to worry if you are powering the motors from 9V and using 3.3V logic. For higher voltages, check out the TB6612. (https://adafru.it/sdc) For much higher voltages and currents check out the DRV8871! (http://adafru.it/3190)



Comes as one assembled and tested breakout plus a small strip of header. You'll need to do some light soldering to attach the header onto the breakout PCB. Arduino, motors, and power supply not included.

Pinouts



Power Pins

- Vmotor - This is the voltage for the motors, not for the logic level. Keep this voltage between 2.7V and 10.8V. This power supply will get noisy so if you have

a system with analog readings or RF other noise-sensitive parts, you may need to keep the power supplies seperate (or filtered!). The terminal block has a simple polarity protection on the + pin that feeds into VM. The VM pin is not protected, but VMotor is!
- GND - This is the shared logic and motor ground. All grounds are connected

Signal in Pins

These are all '2.7V or higher logic level' inputs

- AIN1, AIN2 - these are the two inputs to the Motor A H-bridges. If you want to use speed control, PWM the pin that is normally high. If you dont need PWM control, connect them to logic high/low.
- BIN1, BIN2 - these are the two inputs to the Motor B H-bridges. If you want to use speed control, PWM the pin that is normally high. If you dont need PWM control, connect them to logic high/low.
- FLT -This is the Fault output, which will drive low if there's a thermal shutdown or overcurrent. Note it is open drain so connect a pullup resistor to your desired logic voltage!
- SLP - this is the sleep pin for quickly disabling the driver. By default it is pulled low with an internal 500K resistor, so the chip is not active! Connect to a logic high pin either directly or via a pullup resistor to enable the motor control!

Current Limit Pins

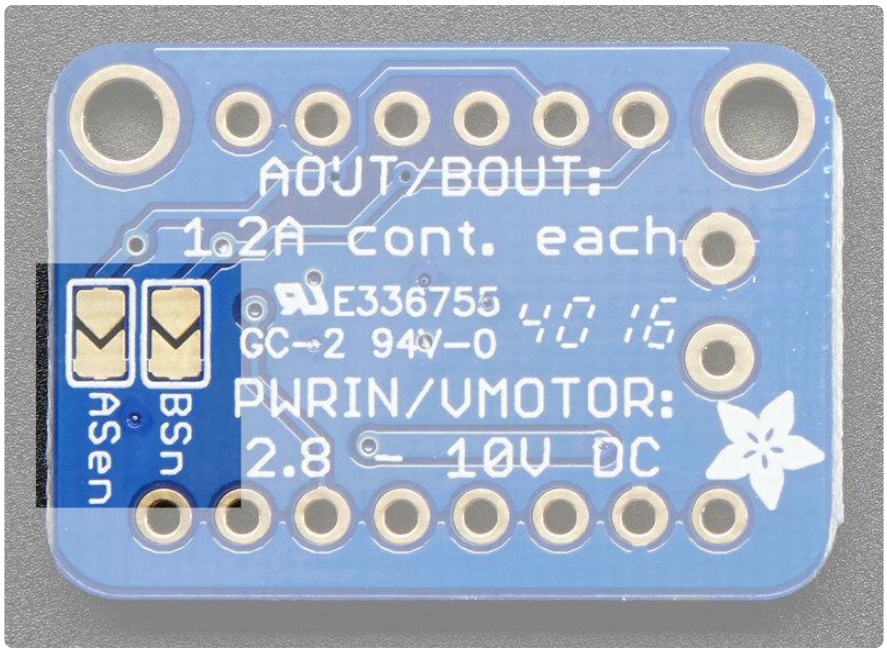
The DRV8833 can perform current limiting for each motor H-bridge. Basically a resistor is connected between Asen and ground to set the Motor A limit (ditto for Bsen and Motor B)

The current limiting rule is: LimitCurrent (amps) = 0.2 V / RSENSE

By default, there are two 1206-sized 0.2Ω resistors on the board for both motors. That means you have a limit of 1 Amp per

If you'd like to raise the limit, you can put a 0.2Ω ohm from Asen to ground, which will then make the RSENSE equal to 0.1Ω (2 parallel 0.2Ω resistors) for a limit of 2A.

You can also totally disable current limiting by soldering closed the two jumpers on the back.



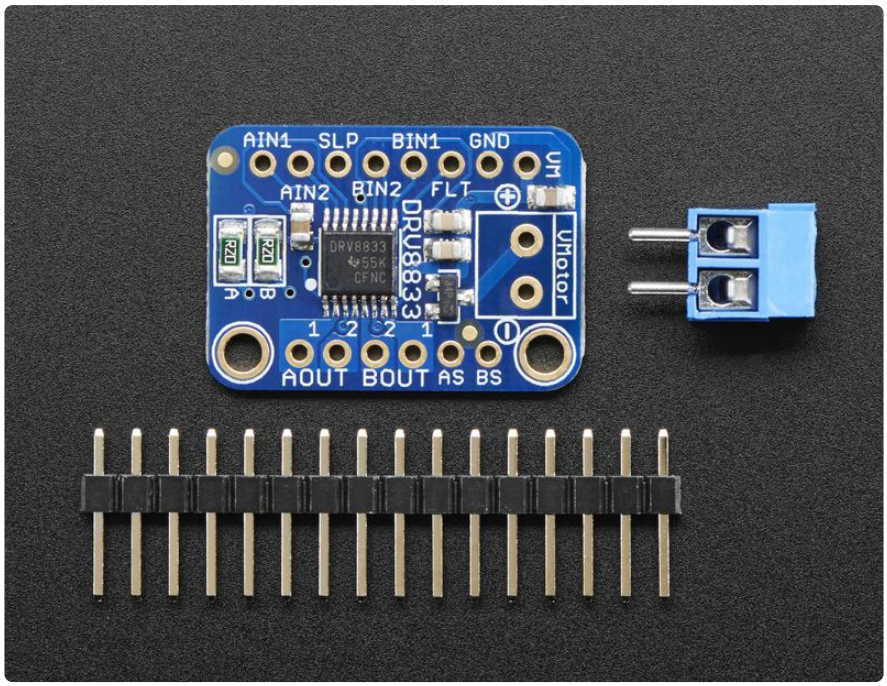
If you want a lower current limit, remove/destroy the 0.2Ω resistor on the board and add your own resistor value between Asen or Bsen and ground.

Motor Out Pins

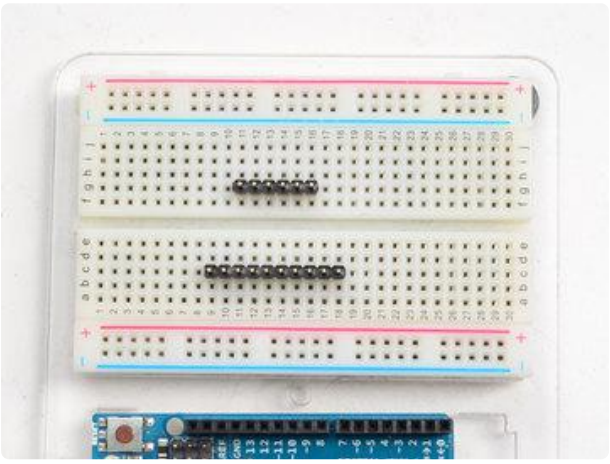
These are motor power outputs

- Motor A - these are the two outputs for motor A, controlled by AIN1 and AIN2
- Motor B - these are the two outputs for motor B, controlled by BIN1 and BIN2

Assembly

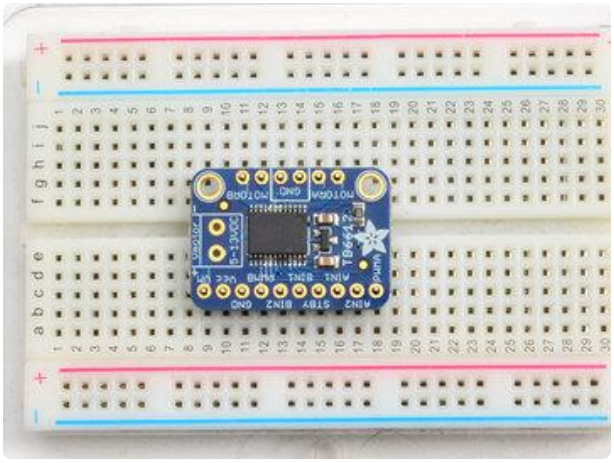


The assembly images below show the similar TB6612 instead of the DRV8833 breakout but the procedure is identical!



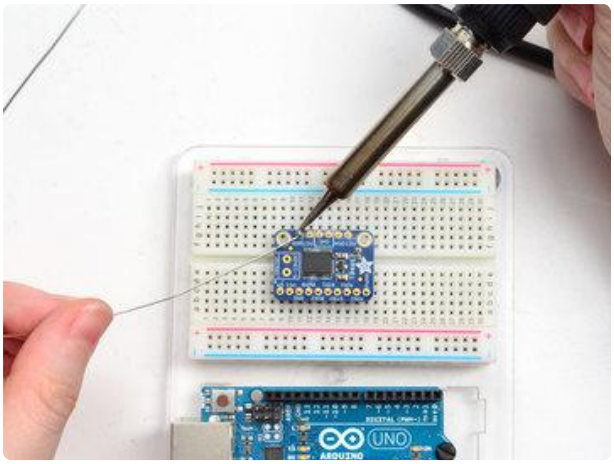
Prepare the header strip:

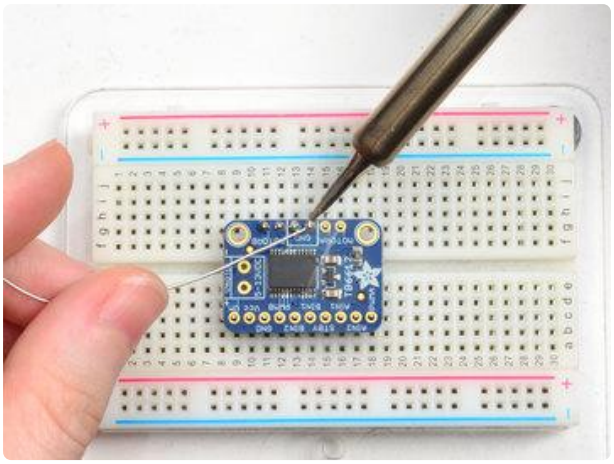
Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - long pins down

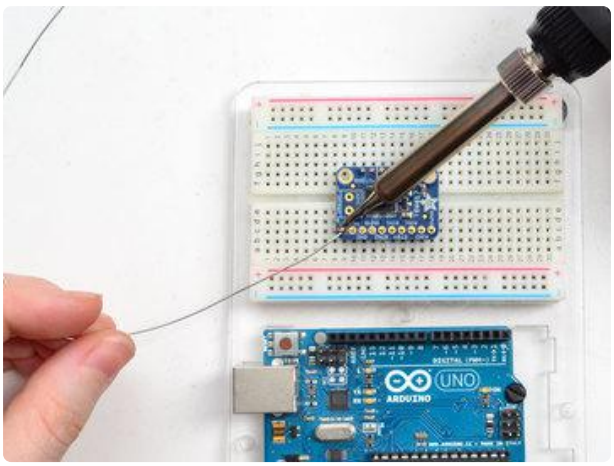


Add the breakout board:

Place the breakout board over the pins so that the short pins poke through the breakout pads



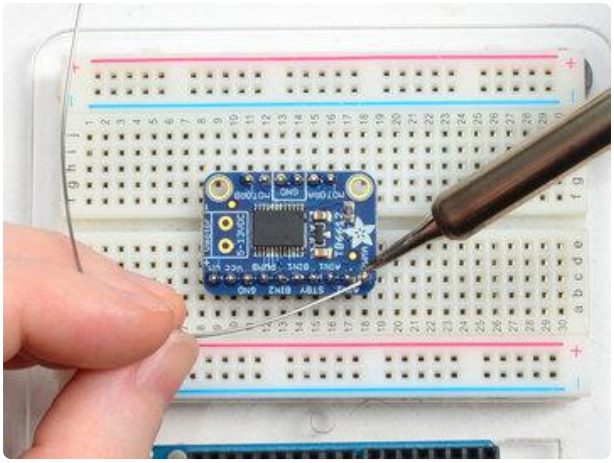




And Solder!

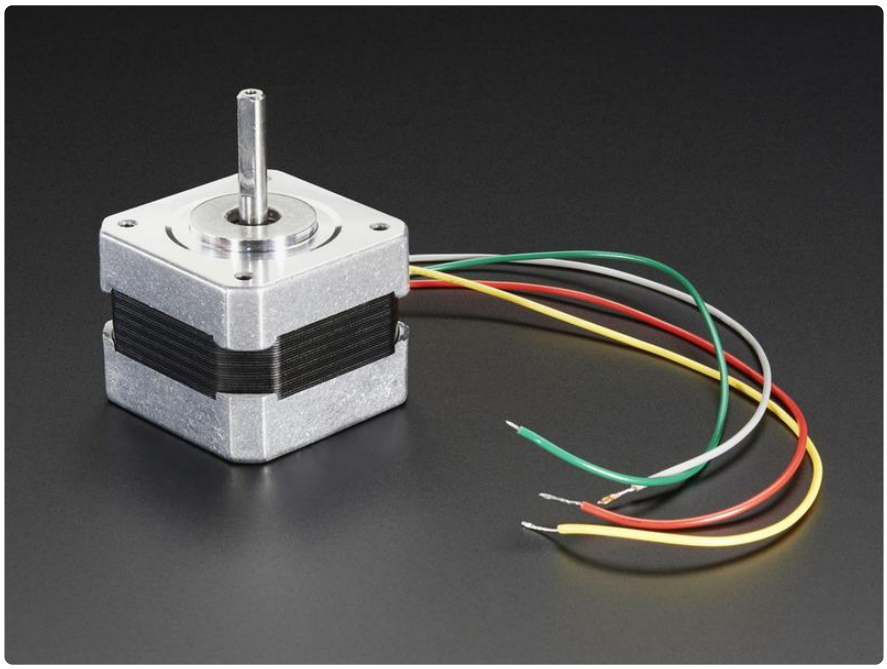
Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our Guide to Excellent Soldering (https://adafru.it/aTk)).



Arduino Usage

In this example we'll wire up and use a bi-polar stepper motor with recommended 9V motor voltage, and 200 steps per rotation.



Wiring

We'll wire it to a Metro, but you can use any microcontroller you like!

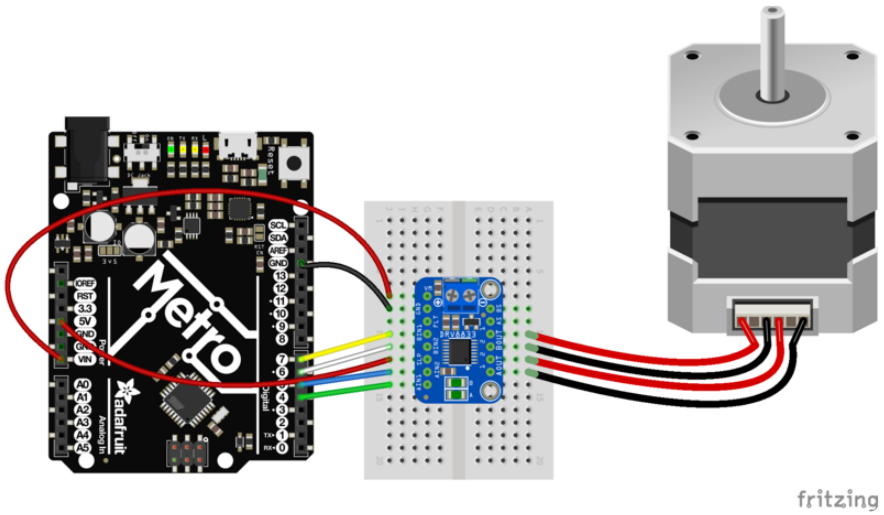
Connect:

- Vmotor to 9V (red wire)
- GND to ground

- SLP to > 2.7V power pin
- AIN1 to Digital 4
- AIN2 to Digital 5
- BIN2 to Digital 6
- BIN1 to Digital 7

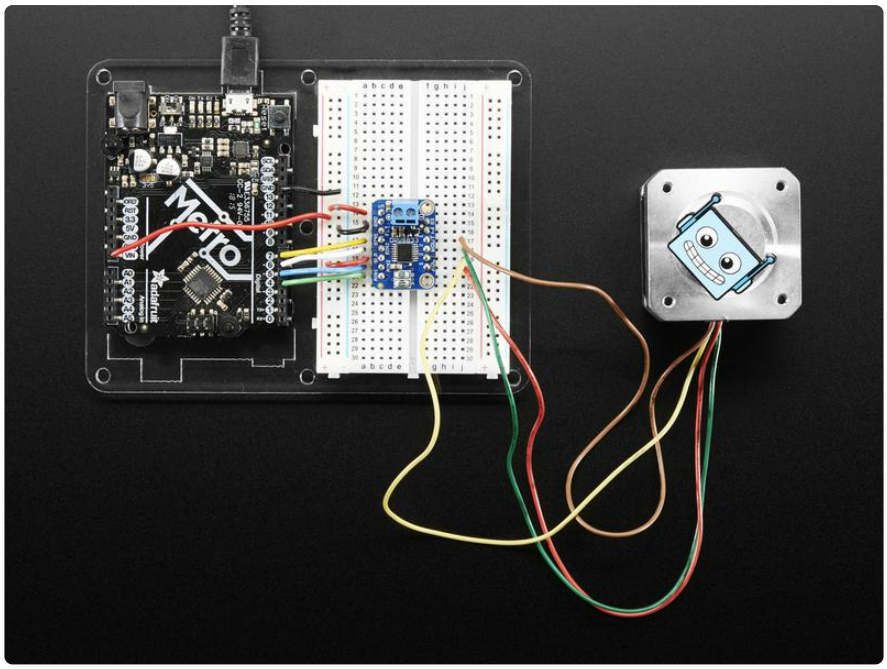
Then hook one stepper motor coil to Motor A (red and yellow) and the second coil to Motor B (green and gray/brown). If you have another motor, you'll need to experiment a little to figure out which wires are which coil. Check any documentation you have! You can use a multimeter to measure between wires, the ones with a small resistance between them are a pair to a coil, for example. If the motor is vibrating but not spinning, check all wires are connected and try flipping around a pair or rechecking the wire pairs.

If you have a unipolar motor, there will be a 5th or 6th wire that is the 'common' wire. Connect these wires to the GND pins in between the Motor A and B outputs on the breakout.



drv8833 fritzing diagram

https://adafru.it/sdd



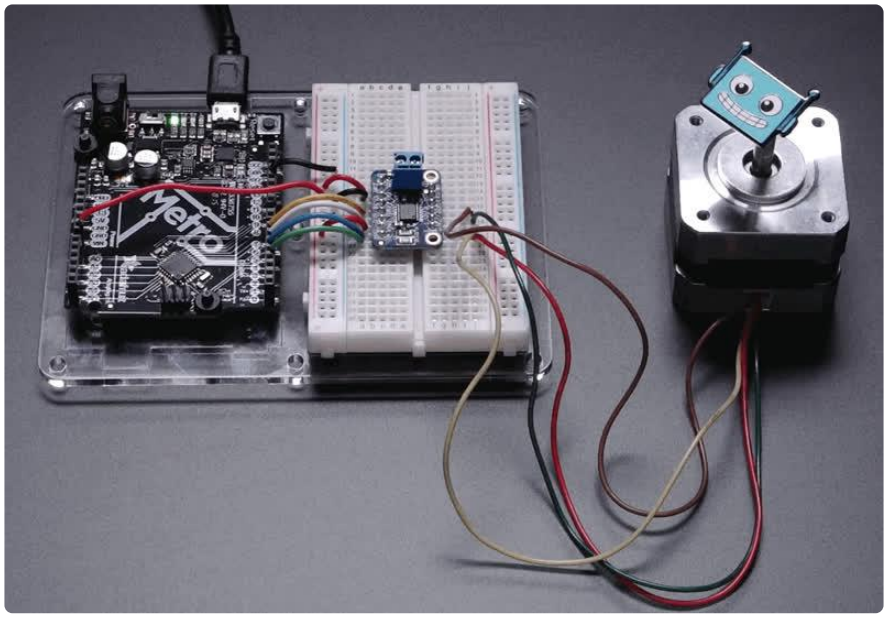
Software

We'll use the built-in Arduino Stepper library (https://adafru.it/eRw), but you can manually toggle the AIN1/AIN2/BIN1/BIN2 pins with your own favorite microcontroller setup

```
#include &lt;Stepper.h&gt;

// change this to the number of steps on your motor
#define STEPS 200

// create an instance of the stepper class, specifying
// the number of steps of the motor and the pins it's
// attached to
Stepper stepper(STEPS, 4, 5, 6, 7);


void setup()
{
  Serial.begin(9600);
  Serial.println("Stepper test!");
  // set the speed of the motor to 30 RPMs
  stepper.setSpeed(60);
}

void loop()
{
  Serial.println("Forward");
  stepper.step(STEPS);
  Serial.println("Backward");
  stepper.step(-STEPS);
}
```

Basically after you make the Stepper object with the 4 control pins, you can set the rotational speed (in RPM) with setSpeed(rpm) and then step forward or backwards with .step(steps) where steps is positive for 'forward' and negative for 'backward'

For more details, check out the Stepper library (https://adafru.it/eRw)



Python & CircuitPython: Stepper Motors

It's easy to use the DRV8833 DC/Stepper Motor Driver or the TB6612 DC/Stepper Motor Driver breakouts with CircuitPython and Python using the Adafruit CircuitPython Motor (https://adafru.it/BCK) library to control stepper motors. We'll show you how to wire them up and use the library to control a stepper motor. The code is the same for both breakouts, however the pinouts on each breakout are slightly different.

You can use this breakout with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library (https://adafru.it/BSN).

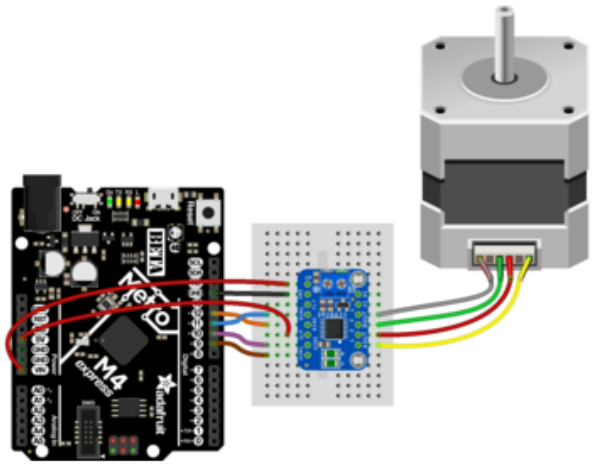
CircuitPython Microcontroller Wiring

You can use any CircuitPython microcontroller board, but bear in mind that motors require external power to run. The Metro M0 and M4 have a convenient pin available for providing sufficient power, so this example will use the Metro M4.

For the VIN pin on the Metro to provide sufficient power to the stepper motor, you must plug in a 9V power supply into the barrel jack built into the Metro, and ensure the switch is "on".

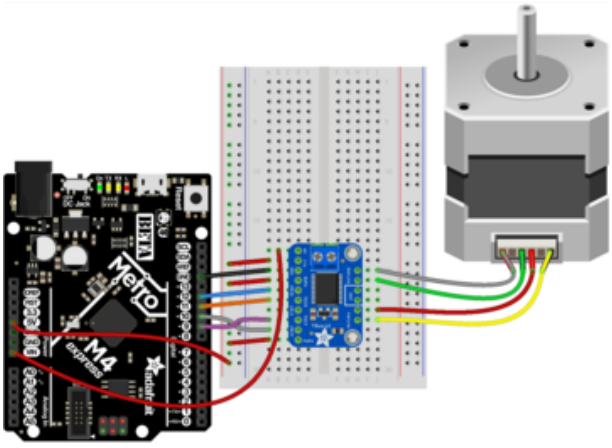
We're using this stepper motor (https://adafru.it/BxE) with this 9V power supply (https://adafru.it/dO6). With the Metro board, the 9V power supply will power both the Metro and the stepper motor.

Connect a Metro M4, the DRV8833 breakout and a stepper motor as follows:



- Board VM to Metro VIN
- Board GND to Metro GND
- Board SLP to Metro 5V
- Board AIN1 to Metro D9
- Board AIN2 to Metro D10
- Board BIN1 to Metro D11
- Board BIN2 to Metro D12
- Board AOUT (1 and 2) to stepper coil (red and yellow stepper wires)
- Board BOUT (1 and 2) to stepper coil (green and grey stepper wires)
- 9V power supply to barrel jack on Metro

For use with the TB6612, connect the Metro M4, breakout and stepper motor as follows:



- Board VM to Metro VIN
- Board VCC to breadboard power rail
- Board GND to Metro GND
- Board PWMB to breadboard power rail
- Board BIN2 to Metro D12
- Board BIN1 to Metro D11
- Board AIN1 to Metro D9
- Board AIN2 to Metro D10
- Board PWMA to breadboard power rail
- Breadboard power rail to Metro 5V
- Board MOTORA (two pins) to stepper coil (red and yellow stepper wires)
- Board MOTORB (two pins) to stepper coil (green and grey stepper wires)
- 9V power supply to barrel jack on Metro

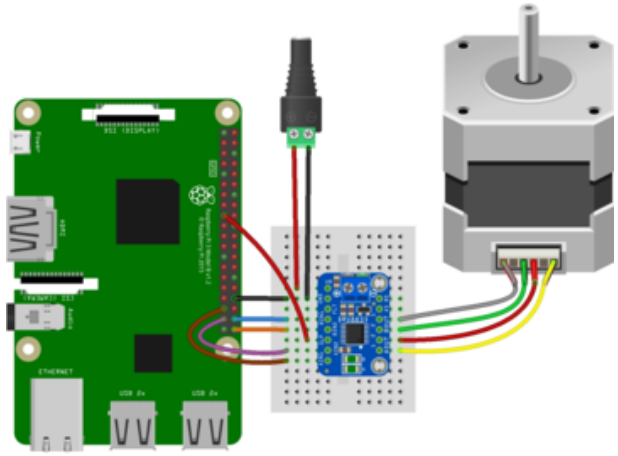
Python Computer Wiring

Since there's dozens of Linux computers/boards you can use we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported (https://adafru.it/BSN).

> The Raspberry Pi will not provide sufficient power to run the stepper motor. You must power the driver board externally with a 9V power supply for the motor to work.

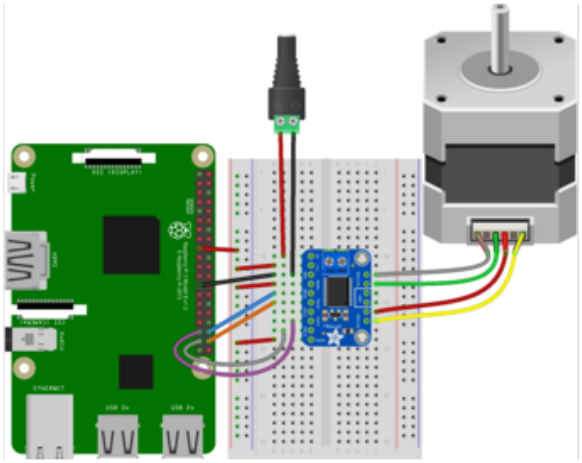
We're using this stepper motor (https://adafru.it/BxE), and powering the driver board with this 9V power supply (https://adafru.it/dO6). With the Raspberry Pi, you'll need a separate 5V power supply for the Pi (on USB as normal).

Here's the Raspberry Pi, breakout and stepper motor wired up to the DRV8833:



- Board VM to positive terminal on barrel jack
- Board GND to negative terminal on barrel jack
- Barrel jack to 9V power supply
- Board GND to Pi GND
- Board SLP to Pi 3.3V
- Board AIN1 to Pi D19
- Board AIN2 to Pi D26
- Board BIN1 to Pi D20
- Board BIN2 to Pi D21
- Board AOUT (1 and 2) to stepper coil (red and yellow stepper wires)
- Board BOUT (1 and 2) to stepper coil (green and grey stepper wires)

And, the Raspberry Pi, breakout and stepper motor wired up to the TB6612:



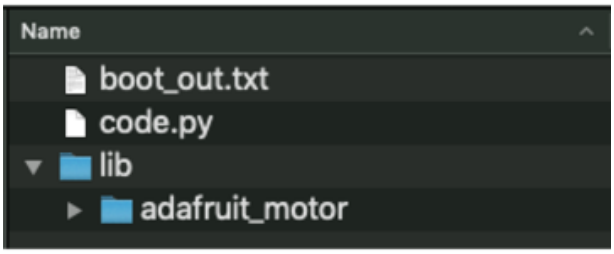
- Board VM to positive terminal on barrel jack
- Board GND to negative terminal on barrel jack
- Barrel jack to 9V power supply
- Board VCC to breadboard power rail
- Board GND to Pi GND
- Board PWMB to breadboard power rail
- Board BIN2 to Pi D21
- Board BIN1 to Pi D20
- Board AIN1 to Pi D19
- Board AIN2 to Pi D26
- Board PWMA to breadboard power rail
- Breadboard power rail to Pi 3.3V
- Board MOTORA (two pins) to stepper coil (red and yellow stepper wires)
- Board MOTORB (two pins) to stepper coil (green and grey stepper wires)

CircuitPython Installation of Motor Library

Next you'll need to install the Adafruit CircuitPython Motor (https://adafru.it/BCK) library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython (https://adafru.it/Em8) for your board.

Next you'll need to install the necessary library to use the hardware--carefully follow the steps to find and install the library from Adafruit's CircuitPython library bundle (https://adafru.it/ENC). Our introduction guide has a great page on how to install modules from the library bundle (https://adafru.it/ABU).



You'll need to manually install the necessary library from the bundle:

- adafruit_motor

Before continuing make sure your board's lib folder or root filesystem has the adafruit_motor folder copied over.

Python Installation of Motor Library

You'll need to install the Adafruit_Blinka library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready (https://adafru.it/BSN)!

Once that's done, from your command line run the following command:

- `pip3 install adafruit-circuitpython-motor`

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

CircuitPython & Python Usage

To demonstrate the usage of the DRV8833 and the TB6612, we'll use a complete code example to control a stepper motor. Code is the same for both boards.

Save the following code to your CIRCUITPY drive as code.py:

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

# Use this example for digital pin control of an H-bridge driver
# like a DRV8833, TB6612 or L298N.

import time
import board
import digitalio
from adafruit_motor import stepper

DELAY = 0.01
STEPS = 200

# You can use any available GPIO pin on both a microcontroller and a Raspberry Pi.
# The following pins are simply a suggestion. If you use different pins, update
# the following code to use your chosen pins.

# To use with CircuitPython and a microcontroller:
coils = (
    digitalio.DigitalInOut(board.D9),  # A1
    digitalio.DigitalInOut(board.D10),  # A2
    digitalio.DigitalInOut(board.D11),  # B1
    digitalio.DigitalInOut(board.D12),  # B2
)

# To use with a Raspberry Pi:
# coils = (
#     digitalio.DigitalInOut(board.D19),  # A1
#     digitalio.DigitalInOut(board.D26),  # A2
#     digitalio.DigitalInOut(board.D20),  # B1
#     digitalio.DigitalInOut(board.D21),  # B2
# )

for coil in coils:
    coil.direction = digitalio.Direction.OUTPUT

motor = stepper.StepperMotor(coils[0], coils[1], coils[2], coils[3],
microsteps=None)

for step in range(STEPS):
    motor.onestep()
    time.sleep(DELAY)

for step in range(STEPS):
    motor.onestep(direction=stepper.BACKWARD)
    time.sleep(DELAY)

for step in range(STEPS):
    motor.onestep(style=stepper.DOUBLE)
    time.sleep(DELAY)

for step in range(STEPS):
    motor.onestep(direction=stepper.BACKWARD, style=stepper.DOUBLE)
    time.sleep(DELAY)
```

```
for step in range(STEPS):
    motor.onestep(style=stepper.INTERLEAVE)
    time.sleep(DELAY)

for step in range(STEPS):
    motor.onestep(direction=stepper.BACKWARD, style=stepper.INTERLEAVE)
    time.sleep(DELAY)

motor.release()
```

Once saved, watch your stepper motor move!

Let's take a look at the code. First, import the necessary libraries. Set the `DELAY` in seconds for the time between each motor control statement, and the number of `STEPS` used in each control block.

Next, set up the pins used by the driver board. If you followed the diagrams above, these will already be correct. If you did not, change these to match the pins you used. If you're using CircuitPython on a microcontroller, no changes are necessary. If you're using a Raspberry Pi with Adafruit Blinka, then you need to comment out the microcontroller pins, and uncomment the Raspberry Pi pins.

Then, set all the pins to `OUTPUT`, and instantiate `motor` with each of the four coils.

Now you can begin controlling the motor using the many options available in the Adafruit CircuitPython Motor library. You can check out the documentation (https://adafru.it/BNE) for details.

That's all there is to controlling a stepper motor with a DRV8833 or the TB6612!

Python Docs

Python Docs (https://adafru.it/C5o)

Downloads

Files

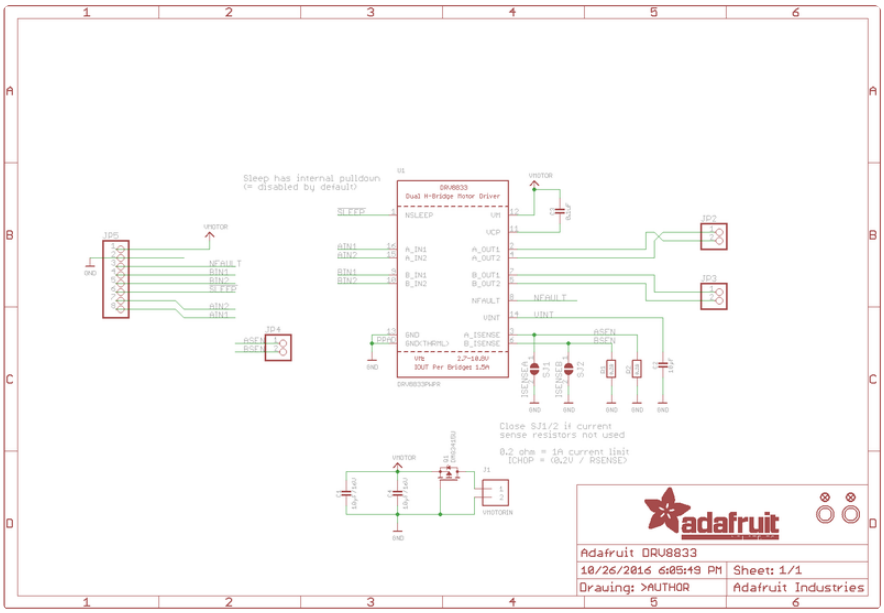
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/aP3)
- EagleCAD PCB files (https://adafru.it/sdf)

This motor driver is a fairly simple breakout of the DRV8833 motor chip, so do check out the datasheet for the DRV8833 for any details you need about pin voltage limits, capacitance, etc!

drv8833 datasheet

https://adafru.it/uNA

Schematic



Fabrication print

Dimensions in inches