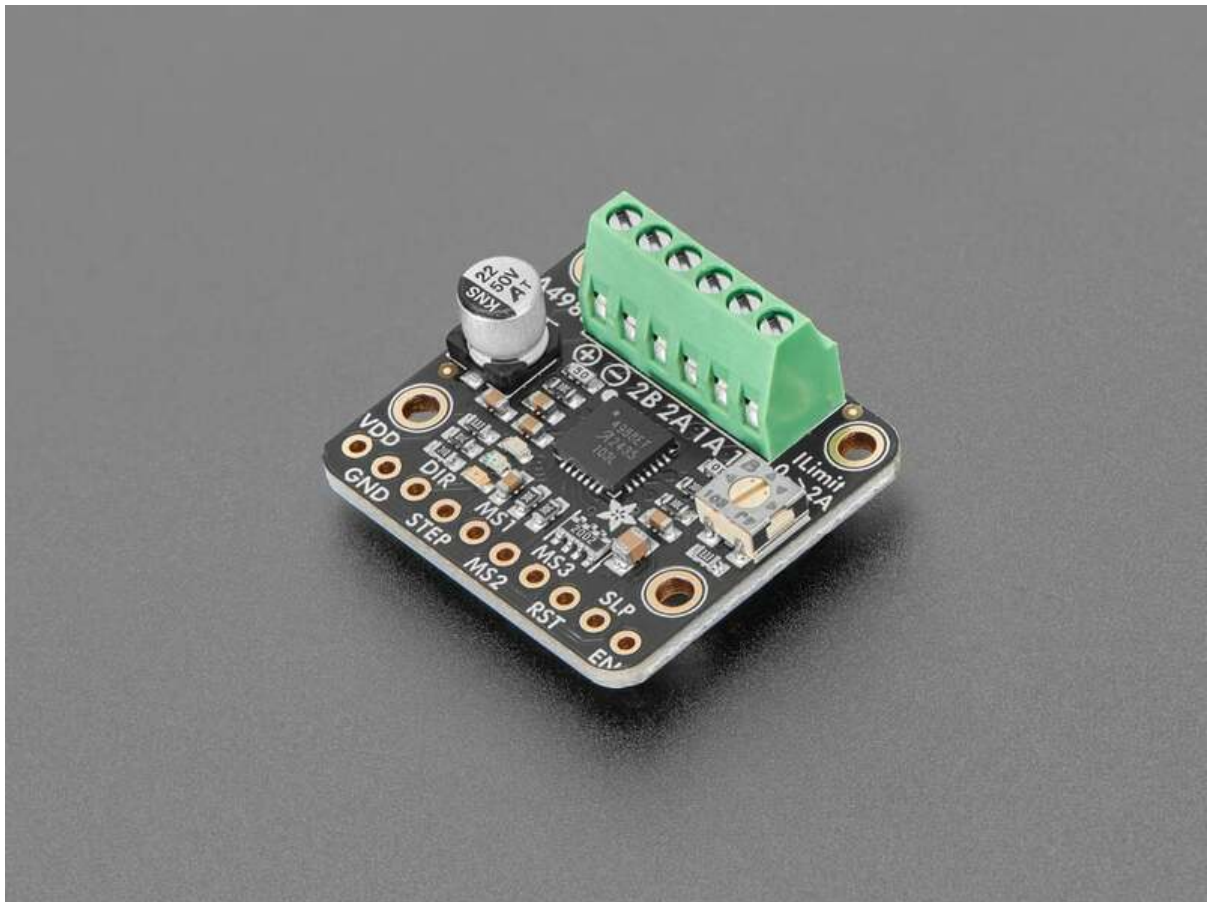




Adafruit A4988 Stepper Motor Driver Breakout Board

Created by Liz Clark



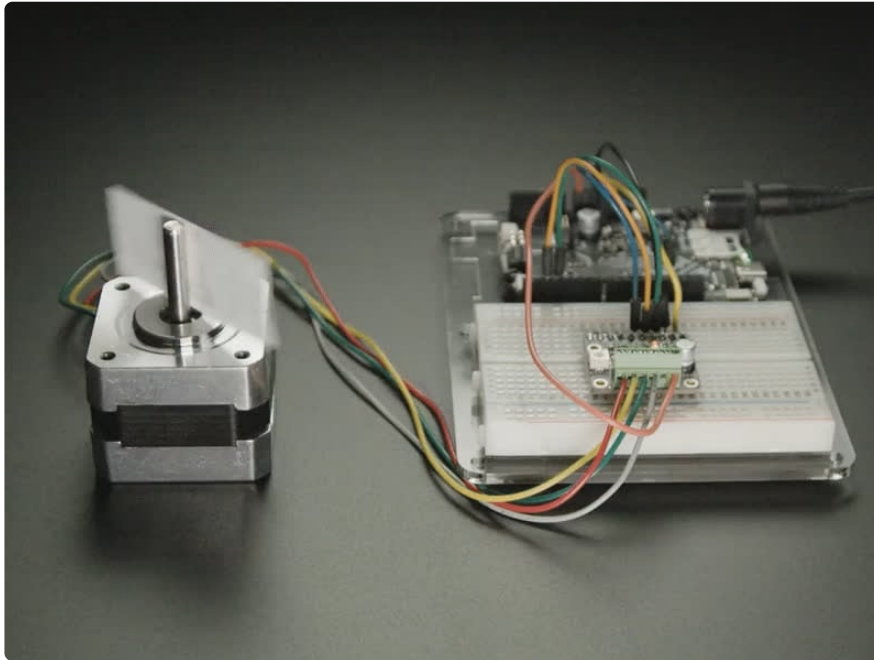
<https://learn.adafruit.com/adafruit-a4988-stepper-motor-driver-breakout-board>

Last updated on 2025-01-16 01:45:17 PM EST

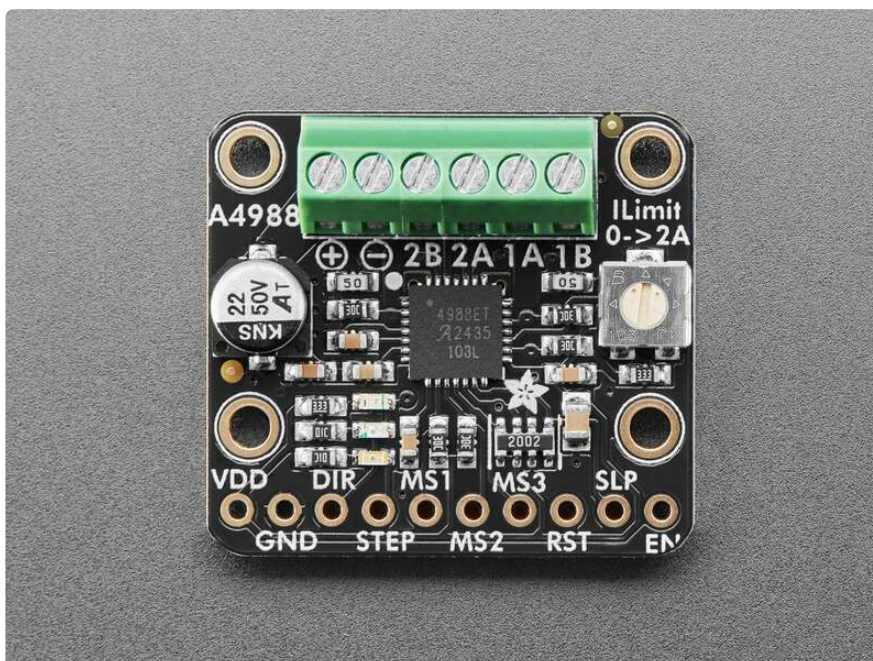
Table of Contents

Overview	3
Pinouts	6
<ul style="list-style-type: none">• Power• Current Potentiometer• Terminal Block Bipolar Stepper Motor Connections• Stepper Control Pins• Microstep Selection Pins• A4988 Control Pins• LEDs• ROSC Jumpers	
CircuitPython and Python	8
<ul style="list-style-type: none">• CircuitPython Microcontroller Wiring• Python Computer Wiring• Python Blinka Setup• CircuitPython Usage• Python Usage• Example Code	
Arduino	13
<ul style="list-style-type: none">• Wiring• Example Code	
Downloads	15
<ul style="list-style-type: none">• Files• Schematic and Fab Print	

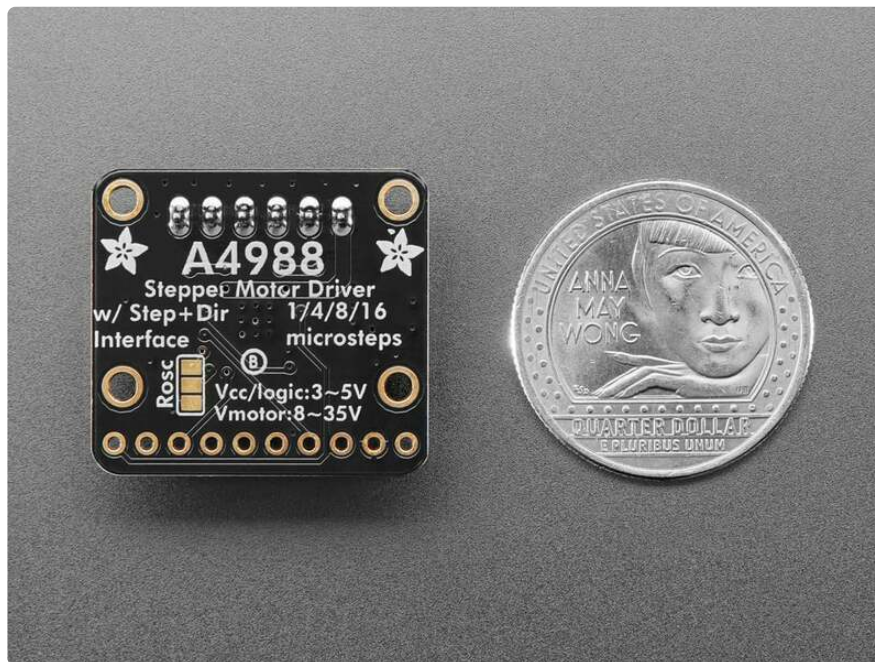
Overview



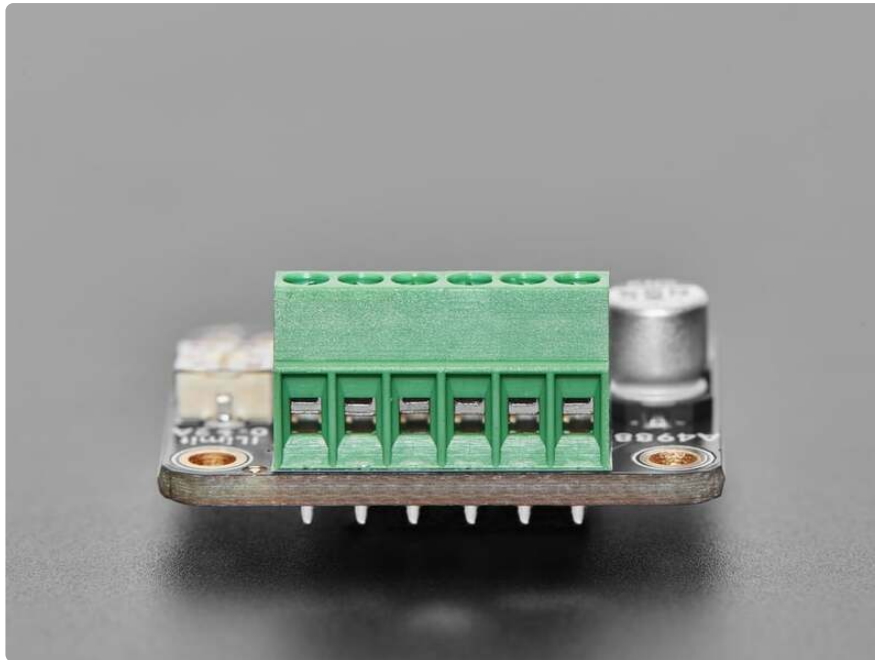
Stepper motors are used for CNC machines, 3D printers, and whenever else one needs precise, powerful motion. But to get good behavior from steppers, you need a motor driver chip that can provide high bursts of current. And for smooth motion, the chip should be able to PWM that current for microstepping support. You can DIY with a lot of timers, a microcontroller and an H-Bridge chip - or you could take the easy way out and use an **Adafruit A4988 Stepper Motor Driver Breakout Board** which makes controlling stepper motors easy-breezy.



All you need is two output pins, no timers, PWM or real-time microcontroller. Set the DIRection pin high or low to set the spin orientation. Then toggle the STEP pin to take one step or microstep at a time. You can set whether you want to go fast with single-step mode or improve the motion precise with 1/4, 1/8 or 1/16 microstepping per STEP toggle. By default the driver is set to 1/16 microstep mode, you can change it by tying the MS1/MS2/MS3 pins low, either with jumpers or with 3 more output pins. The step/microstep mode can even be adjusted on the fly! LEDs on the DIR and STEP pins let you get visual feedback of your motor signal.



The Allegro A4988 is a popular driver chip, with small breakout boards used in many 3D printers. Those breakouts are great for plugging into motherboards, but are a little tough to use for prototyping. Our version comes with terminal blocks for the motor power and stepper wires, plus nicely labeled pins for control and mounting holes.

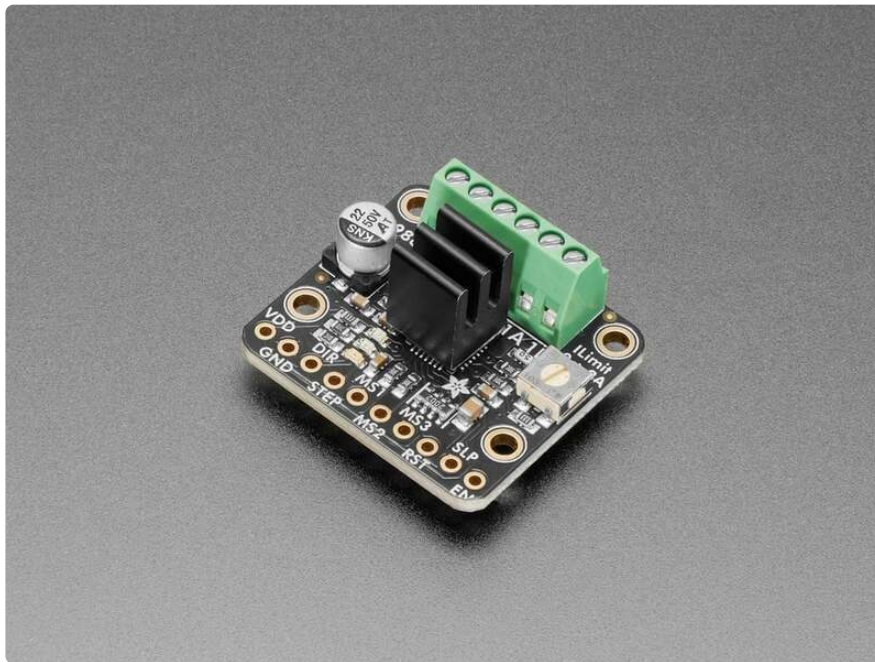


We fabricated the board with 2 oz copper to give it a hand with the 2A-max current that this driver can handle. To use the current limiting capability, twist the onboard potentiometer: when all the way to the right we can get to up to 2A max. Note that the higher currents will heat up both the motor driver and stepper, so you may need to add heatsinking to the chip. We don't include a heatsink but you can get [a tall ~80°C/W](http://adafru.it/1493) (<http://adafru.it/1493>) or [short ~90°C/W heatsink](http://adafru.it/1515) (<http://adafru.it/1515>) to attach on top.

Features:

- [Allegro A4988](https://adafru.it/1abb) (<https://adafru.it/1abb>) DMOS microstepping driver with translator and overcurrent protection
- Motor voltage from 8V to 35VDC
- Vdd/Logic voltage from 3V to 5V, use with anything from an Arduino-compatible or ESP32 to Raspberry Pi other Single Board Computer
- Terminal screw block connections for easy VMotor power and 4-wire bipolar stepper motor connection with 26-20AWG slots, 2.54mm / 0.1" spacing
- Control steppers using only two pins: DIRection and STEP
- Defaults to 1/16 microstep mode, change by grounding MS1/MS2/MS3 pins (see [A4988 datasheet](https://adafru.it/1abc) (<https://adafru.it/1abc>) for pin configuration)
- Red and Green LEDs on DIR signal to let you know forward or backward motion
- Yellow LED on STEP to let you know that motor is being driven
- Sleep, Reset and Enable control lines for low power / deactivation
- Potentiometer to set current limiting, up to 2A
- 22uF 50V electrolytic capacitor on motor power
- 2 Oz copper for better current carrying and heatsinking

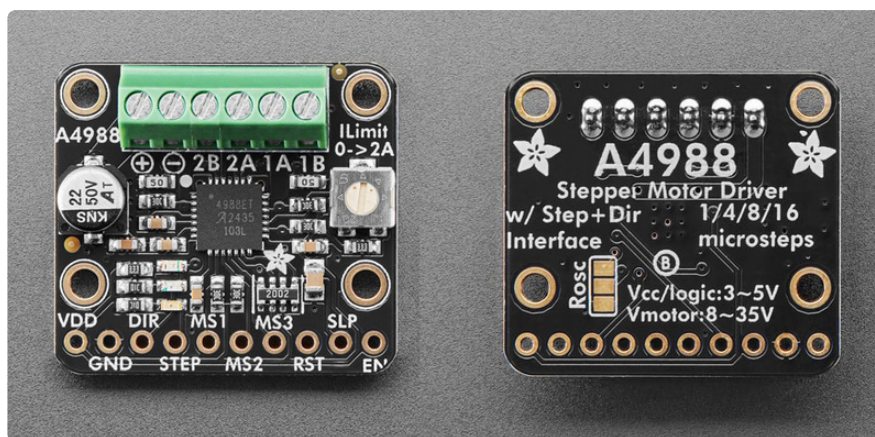
- Four mounting holes



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.

Microcontroller, motors, and power supply not included. You will need some sort of driver board that will toggle the DIR/STEP pins for you.

Pinouts



Power

- **VDD** - This is the logic voltage input. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V micro like Arduino, use 5V. It can be powered between from 3.3V to 5V.
- **+ -** - This is the motor voltage input. Supply 8V to 35VDC for your motor.
- **GND** / - (terminal block) - common ground for power and logic.

Current Potentiometer

On the right side of the board is the current potentiometer. This potentiometer limits the current output to the motor. When the potentiometer is all the way to the right, you can get up to 2A max. Note that the higher currents will heat up both the motor driver and stepper, so you may need to add heatsinking to the chip.

Terminal Block Bipolar Stepper Motor Connections

- **1A** and **1B** - The first set of motor control outputs connected to DMOS bridge 1.
- **2A** and **2B** - The second set of motor control outputs connected to DMOS bridge 2.

Stepper Control Pins

- **DIR** - This is the direction control pin. You can set this pin high or low to set the spin orientation. Pulling it high turns the motor clockwise. Pulling it low turns it counterclockwise.
- **STEP** - This is the microstep control pin. Toggle this pin to take one step or microstep at a time. By default, the driver is set to 1/16 microstep mode.

Microstep Selection Pins

There are three pins (**MS1**, **MS2** and **MS3**) that can be pulled low to change the default 1/16 microstep mode where all three pins are open/high. The table below shows the pin combinations to change to the different modes. For example, if you wanted to use 1/4 microstep mode, you would tie **MS1** and **MS3** to ground (closed/low) and leave **MS2** as-is (open/high).

STEP	MS1	MS2	MS3
1	L	L	L
1/2	H	L	L
1/4	L	H	L
1/8	H	H	L
1/16	H	H	H

A4988 Control Pins

- **RST** - This is the reset pin. Pull this pin low to reset the A4988 to its home state.
- **SLP** - This is the sleep pin. Pull this pin low to put the breakout into a low power sleep mode. Pull the pin high to disable sleep mode.
- **EN** - This is the enable pin. Pull this pin high to disable the output to the motors.

LEDs

- **Green LED** - The green LED is tied to the DIR pin. It is lit when the motor is being driven counterclockwise when the DIR pin is low.
- **Red LED** - The red LED is tied to the DIR pin. It is lit when the motor is being driven clockwise when the DIR pin is high.
- **Yellow LED** - The yellow LED is tied to the STEP pin. It is lit when the motor driver is being moved.

ROSC Jumpers

On the back of the board is a set of jumpers that are tied to the **ROSC** pin. This pin sets the off-time for the internal PWM current control that determines the off time and decay mode for the motors (how quickly the A4988 reduces current being driven to the motor).

By default, with the jumpers open, the **ROSC** pin is tied to ground with a 20K resistor, setting the off time to approximately 24.2 μ s (resistor value divided by 825). The decay mode is automatic mixed (slow and fast), except when driving the motor in full-step mode. Full-step mode uses slow decay.

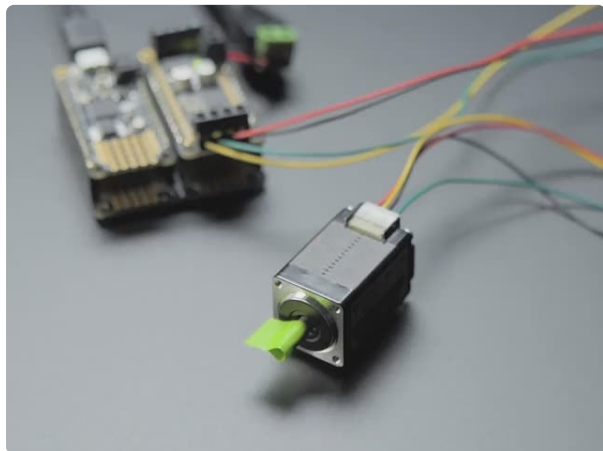
If you solder the **upper jumper closed**, the **ROSC** pin is tied to **VDD**. This sets the off time internally to 30 μ s. The decay mode is automatic mixed (slow and fast), except when driving the motor in full-step mode. Full-step mode uses slow decay.

If you solder the **lower jumper closed**, the **ROSC** pin is tied to **GND**. This sets the off time internally to 30 μ s. The decay mode is automatic mixed (slow and fast) for all step modes.

CircuitPython and Python

It's easy to use the **A4988 Stepper Motor Driver** with CircuitPython and the [digitalio \(https://adafru.it/19ID\)](https://adafru.it/19ID) core module. This module allows you to easily write Python code to access input and outputs on GPIO pins.

You can use the example code 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) (<https://adafru.it/BSN>).



Mini Stepper Motor - 200 Steps - 20x30mm NEMA-8 Size

This tiny stepper motor is the same quality and step-size as the big NEMA-17's we stock, but so cute so it will work with compact CNC builds. This 4-wire bipolar stepper has...

<https://www.adafruit.com/product/4411>



Adjustable Power Supply with 2.1mm / 5.5mm DC - 3V to 12V at 5A

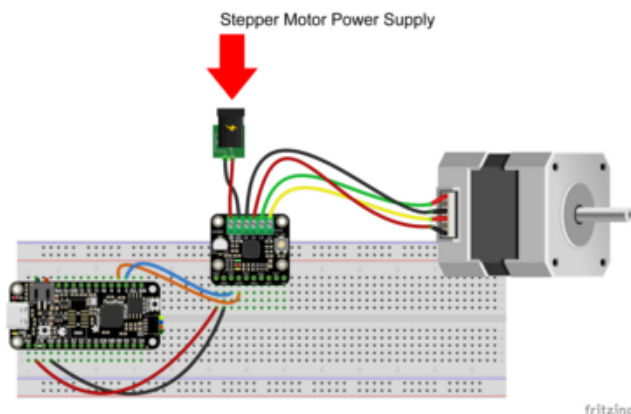
Put your Snap! cassette on and sing along to "I got the power!" with this super useful power supply adapter where...

<https://www.adafruit.com/product/4880>

CircuitPython Microcontroller Wiring

Here is how you'll wire the breakout to a Feather RP2040 and stepper motor:

Check your stepper motor wiring - your motor may have different wire colors or wire order.



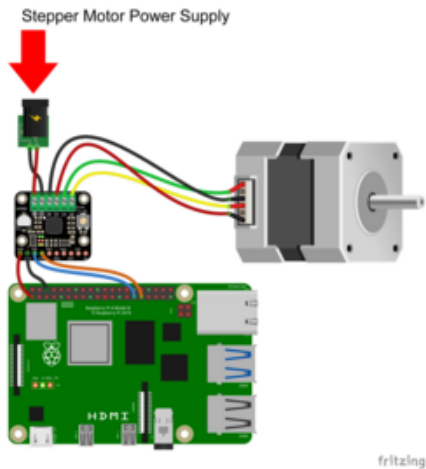
Stepper motor power supply positive to breakout terminal block + (red wire)
 Stepper motor power supply negative to breakout terminal block - (black wire)
 Breakout VDD to Feather 3.3V (red wire)
 Breakout GND to Feather GND (black wire)
 Breakout DIR to Feather pin 5 (blue wire)
 Breakout STEP to Feather pin 6 (orange wire)
 Breakout 1A to stepper motor coil 1 positive (green wire)
 Breakout 1B to stepper motor coil 1 negative (yellow wire)
 Breakout 2A to stepper motor coil 2 positive (red wire)
 Breakout 2B to stepper motor coil 2 negative (black wire)

Python Computer Wiring

Since there are 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 \(https://adafru.it/BSN\)](https://adafru.it/BSN).

Here's the Raspberry Pi wired to the breakout and a stepper motor:

Check your stepper motor wiring - your motor may have different wire colors or wire order.



Stepper motor power supply positive to breakout terminal block + (red wire)
 Stepper motor power supply negative to breakout terminal block - (black wire)
 Breakout VDD to Pi 3.3V (red wire)
 Breakout GND to Pi GND (black wire)
 Breakout DIR to Pi GPIO 5 (blue wire)
 Breakout STEP to Pi GPIO 6 (orange wire)
 Breakout 1A to stepper motor coil 1 positive (green wire)
 Breakout 1B to stepper motor coil 1 negative (yellow wire)
 Breakout 2A to stepper motor coil 2 positive (red wire)
 Breakout 2B to stepper motor coil 2 negative (black wire)

Python Blinka Setup

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 \(https://adafru.it/BSN\)](https://adafru.it/BSN)!

CircuitPython Usage

To use with CircuitPython, you need to update **code.py** with the example script.

In the example below, click the **Download Project Bundle** button below to download the **code.py** file in a zip file. Extract the contents of the zip file, and copy the **code.py** file to your **CIRCUITPY** drive.

There are no additional libraries needed for this example. It uses core modules that are included when you install CircuitPython on your board.

Python Usage

Once you have the library `pip3` installed on your computer, copy or download the following example to your computer, and run the following, replacing `code.py` with whatever you named the file:

```
python3 code.py
```

Example Code

If running CircuitPython: Once everything is saved to the **CIRCUITPY** drive, [connect to the serial console \(https://adafru.it/Bec\)](https://adafru.it/Bec) to see the data printed out!

If running Python: The console output will appear wherever you are running Python.

```
# SPDX-FileCopyrightText: 2024 Liz Clark for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
from digitalio import DigitalInOut, Direction

# direction and step pins as outputs
DIR = DigitalInOut(board.D5)
DIR.direction = Direction.OUTPUT
STEP = DigitalInOut(board.D6)
STEP.direction = Direction.OUTPUT

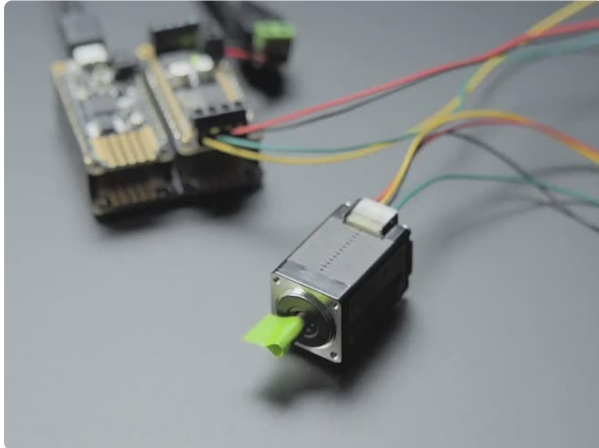
# microstep mode, default is 1/16 so 16
# another ex: 1/4 microstep would be 4
microMode = 16
# full rotation multiplied by the microstep divider
steps = 200 * microMode

while True:
    # change direction every loop
    DIR.value = not DIR.value
    # toggle STEP pin to move the motor
    for i in range(steps):
        STEP.value = True
        time.sleep(0.001)
        STEP.value = False
        time.sleep(0.001)
    print("rotated! now reverse")
    # 1 second delay before starting again
    time.sleep(1)
```

The code starts by setting up the direction and step pins as outputs. In the loop, the direction pin will toggle once every loop to change directions. The step pin toggles in a `for` loop to step the motor. As the code runs, you'll see your attached stepper motor turn clockwise and then reverse and turn counterclockwise.

Arduino

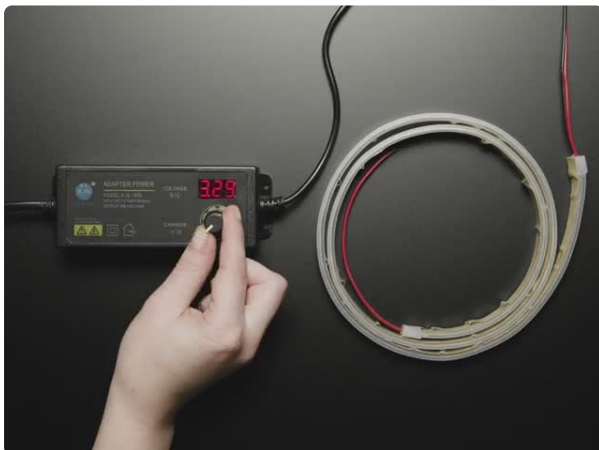
Using the A4988 breakout with Arduino involves wiring up the breakout with a stepper motor to your Arduino-compatible microcontroller and running the provided example code.



Mini Stepper Motor - 200 Steps - 20x30mm NEMA-8 Size

This tiny stepper motor is the same quality and step-size as the big NEMA-17's we stock, but so cute so it will work with compact CNC builds. This 4-wire bipolar stepper has...

<https://www.adafruit.com/product/4411>



Adjustable Power Supply with 2.1mm / 5.5mm DC - 3V to 12V at 5A

Put your Snap! cassette on and sing along to "I got the power!" with this super useful power supply adapter where...

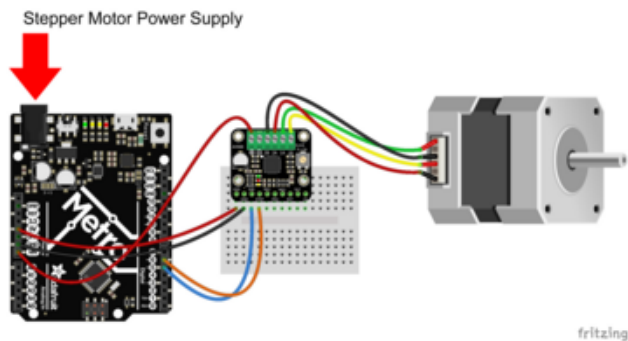
<https://www.adafruit.com/product/4880>

Wiring

Wire as shown for a **5V** board like an Uno. If you are using a **3V** board, like an Adafruit Feather, wire the board's 3V pin to the breakout VDD.

Here is an Adafruit Metro wired up to the breakout with a stepper motor. You'll need to connect the stepper motor power supply to the DC jack on the Metro.

Check your stepper motor wiring - your motor may have different wire colors or wire order.



Stepper motor power supply to Metro DC Jack

Breakout VDD to Metro 5V (red wire)

Breakout GND to Metro GND (black wire)

Breakout DIR to Metro pin 5 (blue wire)

Breakout STEP to Metro pin 6 (orange wire)

Breakout terminal block + to Metro VIN (red wire)

Breakout 1A to stepper motor coil 1 positive (green wire)

Breakout 1B to stepper motor coil 1 negative (yellow wire)

Breakout 2A to stepper motor coil 2 positive (red wire)

Breakout 2B to stepper motor coil 2 negative (black wire)

Example Code

```
// SPDX-FileCopyrightText: 2024 Liz Clark for Adafruit Industries
//
// SPDX-License-Identifier: MIT

const int DIR = 5;
const int STEP = 6;
const int microMode = 16; // microstep mode, default is 1/16 so 16; ex: 1/4 would be 4
// full rotation * microstep divider
const int steps = 200 * microMode;

void setup()
{
  // setup step and dir pins as outputs
  pinMode(STEP, OUTPUT);
  pinMode(DIR, OUTPUT);
}

void loop()
{
  // change direction every loop
  digitalWrite(DIR, !digitalRead(DIR));
  // toggle STEP to move
  for(int x = 0; x < steps; x++)
  {
    digitalWrite(STEP, HIGH);
    delay(2);
    digitalWrite(STEP, LOW);
    delay(2);
  }
  delay(1000); // 1 second delay
}
```

Upload the sketch to your board. You'll see your attached stepper motor turn slowly clockwise and then reverse and turn slowly counterclockwise.

Downloads

Files

- [A4988 Datasheet \(https://adafru.it/1abd\)](https://adafru.it/1abd)
- [EagleCAD PCB Files on GitHub \(https://adafru.it/1abe\)](https://adafru.it/1abe)
- [3D models on GitHub \(https://adafru.it/1ac9\)](https://adafru.it/1ac9)
- [Fritzing object in the Adafruit Fritzing Library \(https://adafru.it/1abf\)](https://adafru.it/1abf)

Schematic and Fab Print

