Quick Start Guide

Two phases hybrid 42 stepper motor

Introduction

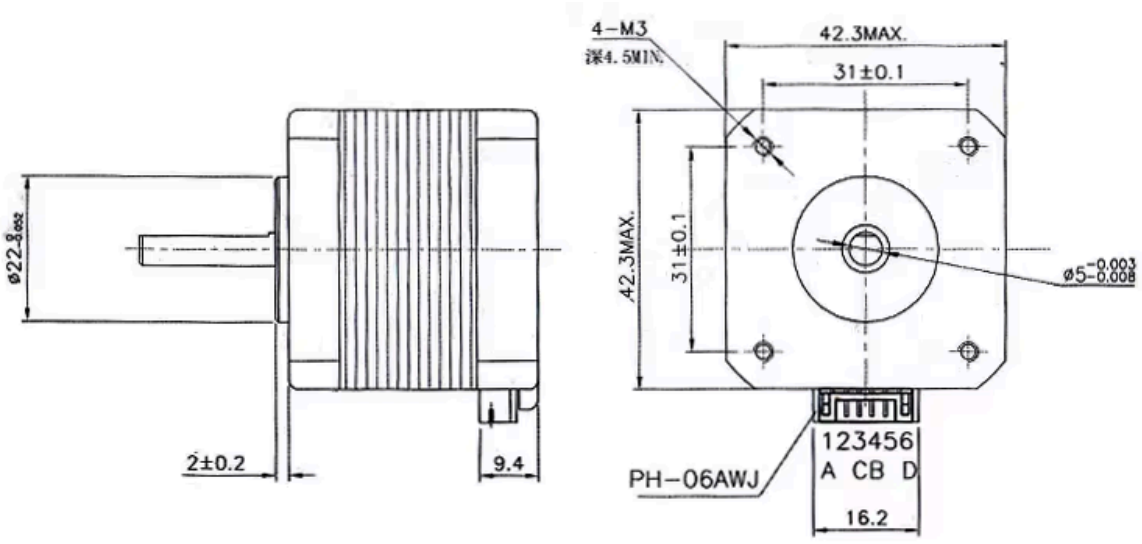
Thank you for choosing the two-phase hybrid 42 stepper motor!

This versatile and high-performance motor is designed to deliver precise, smooth, and reliable motion for a wide range of applications, from CNC machines to 3D printers. With its hybrid stepper motor design, it combines the best features of variable reluctance and permanent magnet stepper motors, offering higher torque, improved accuracy, and excellent efficiency.

In this guide, we'll provide you with all the essential information to maximize the performance of your two-phase hybrid 42 stepper motor. You'll learn about proper wiring and installation techniques, providing examples for interfacing it with a microcontroller.

Specifications

Reference	NEMA17
Current/Phase	0.4A
Holding Torque	28 Nm
Step Angle (degrees)	1.8°
Phases	2
Model Number	Height 34 mm
Type	Permanent Magnet
Voltage	12V
Frame size	42x42mm
Dimensions (LxWxH)	34x42x42mm



Functions

- **3D Printing Axis Control**: Drive the X, Y, and Z axes in 3D printers for precise positioning, enabling high-quality layer alignment and intricate designs.
- **CNC Machine Precision**: Execute accurate tool head movements in CNC machines for cutting, engraving, or drilling tasks, ensuring smooth, repeatable operations.
- **Robotic Joint Control**: Power robotic arms or actuators for precise angular motion, enhancing the performance of assembly lines, pick-and-place robots, and automated systems.
- **Industrial Conveyor Automation**: Manage conveyor belt motion for controlled item positioning and sorting in manufacturing and packaging applications.
- **Medical Device Positioning**: Enable exact positioning in medical diagnostic equipment and lab automation, improving sample handling and imaging accuracy.
- **Camera and Antenna Movement**: Provide smooth motion for camera gimbals, telescope mounts, and antenna trackers, ensuring stability and precision in professional photography and communication systems.

Overview

A Stepper motor is an electromechanical device that converts electrical pulses into discrete mechanical movements. The shaft of a stepper motor rotates in discrete steps as electrical command pulses are applied to the motor winding coils.

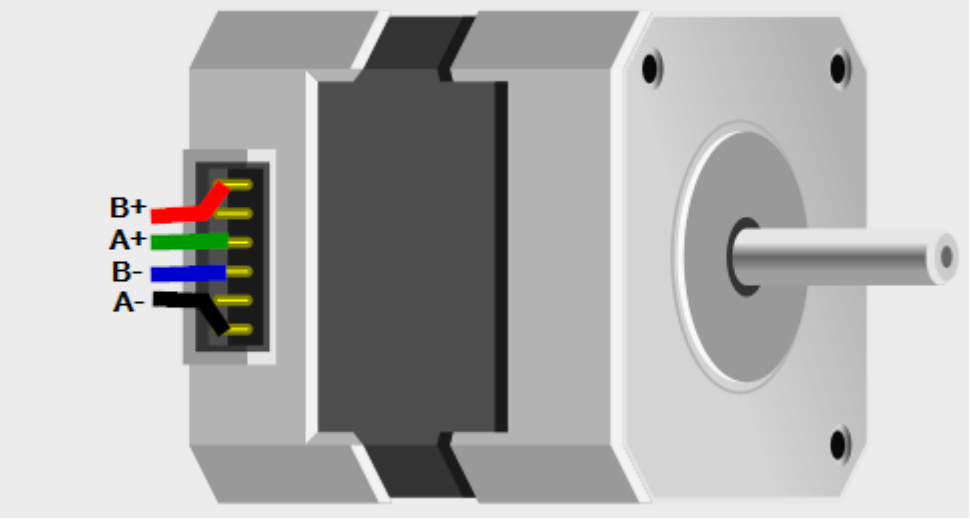
Pinout

Below is the pinout of a 6-wire stepper motor, highlighting two distinct coils:

Coil A (green wire, black wire)

Coil B (red wire, blue wire).

Each coil features two endpoints labeled as **A+**, **A-** (for Coil A) and **B+**, **B-** (for Coil B).



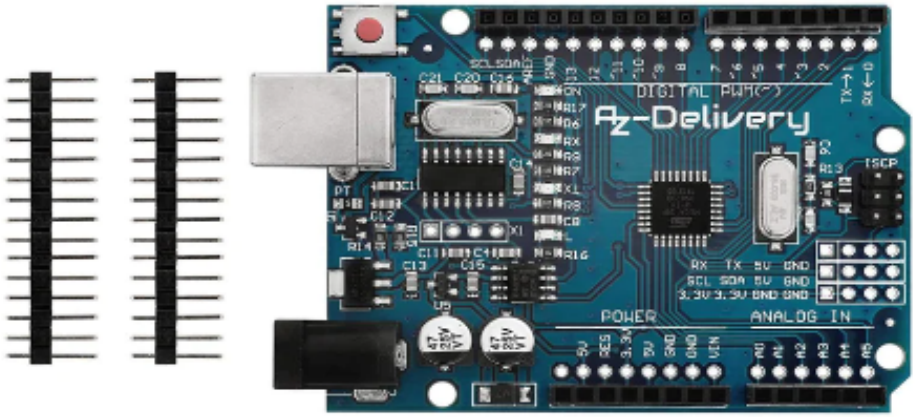
Control stepper motor with A4988

In this section, we will discuss how to control the NEMA17 stepper motor using a microcontroller and the A4988 driver. We will cover the wiring setup, configuring the A4988 current limit, and running test code to move the motor step by step.

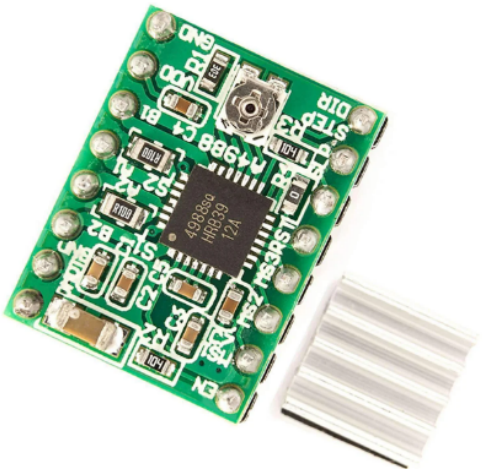
Setting up Arduino environment

Setup for Development environment, we need :

-Mikrocontroller Board



-A4988 driver

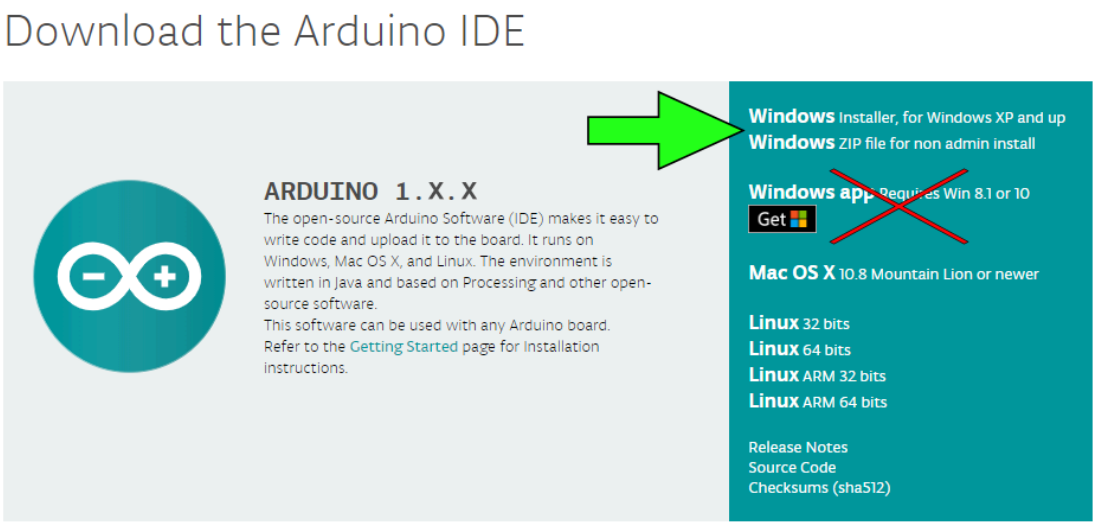


- Power supply 12V

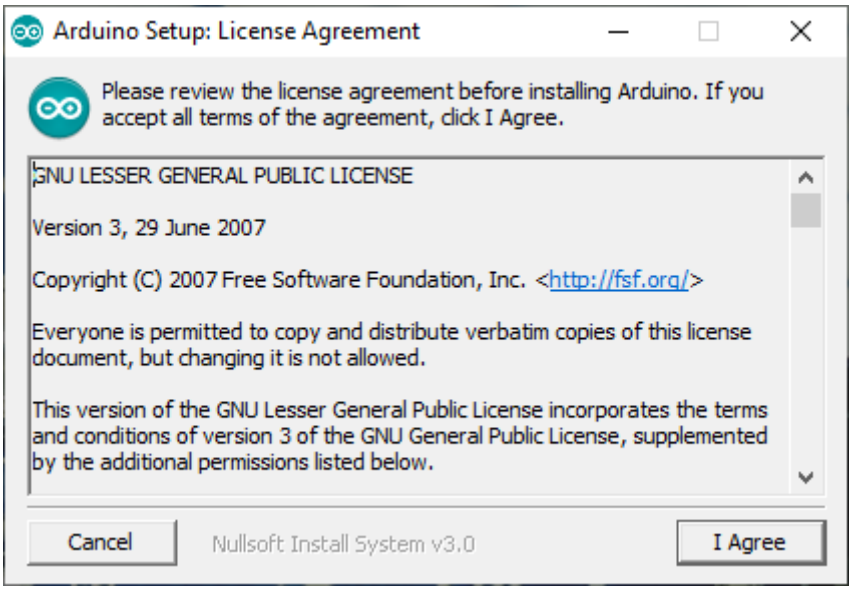
Installation of the Arduino IDE

You can download the free Arduino IDE development environment from the following link: https://www.arduino.cc/en/Main/Software

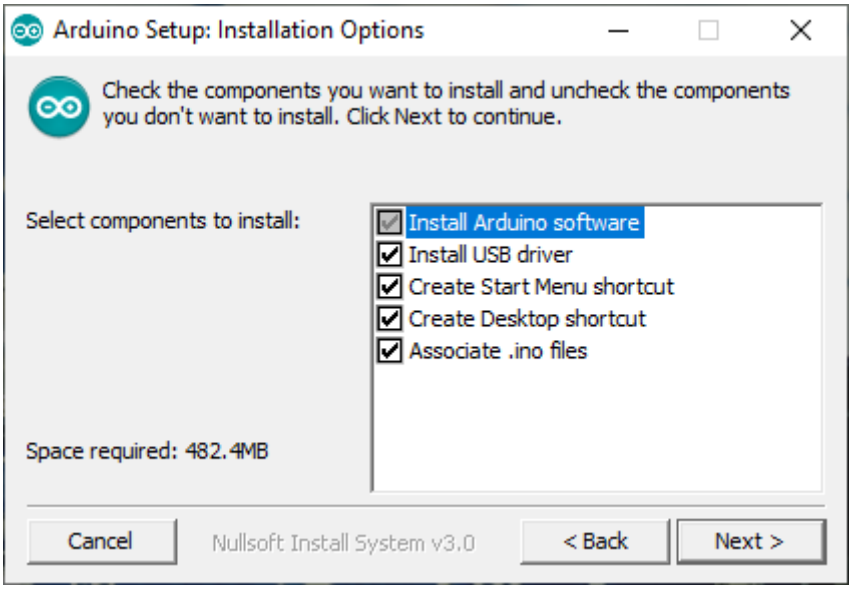
Windows users should definitely use one of the first two download options for the Arduino IDE. The "Windows App" version from the Windows Store will cause connection problems especially when using third party board definitions.



After starting the Arduino IDE installation file "arduino-1.X.X-windows.exe" the license conditions of the software must be read and accepted:



In the next step, different options can be selected for installation.

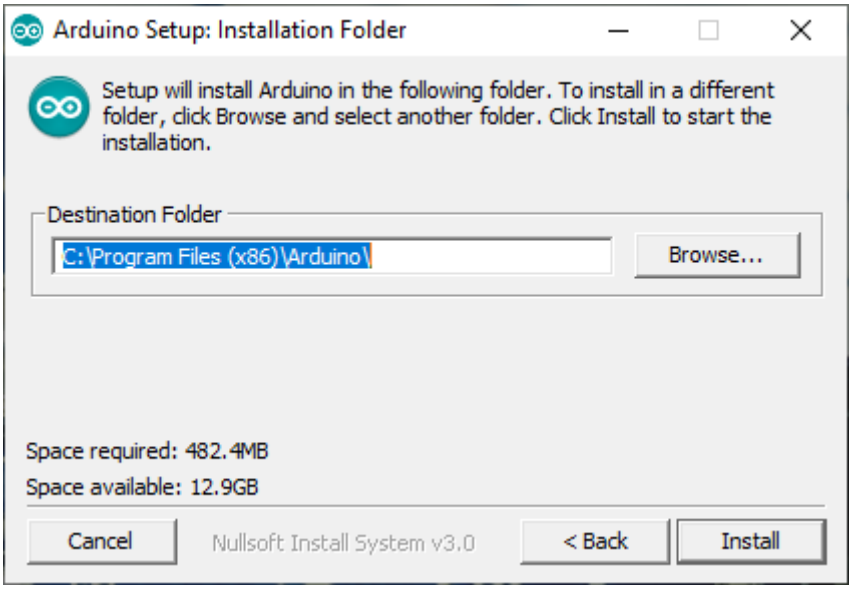


A brief overview of the different options follows, with a brief explanation of each option:

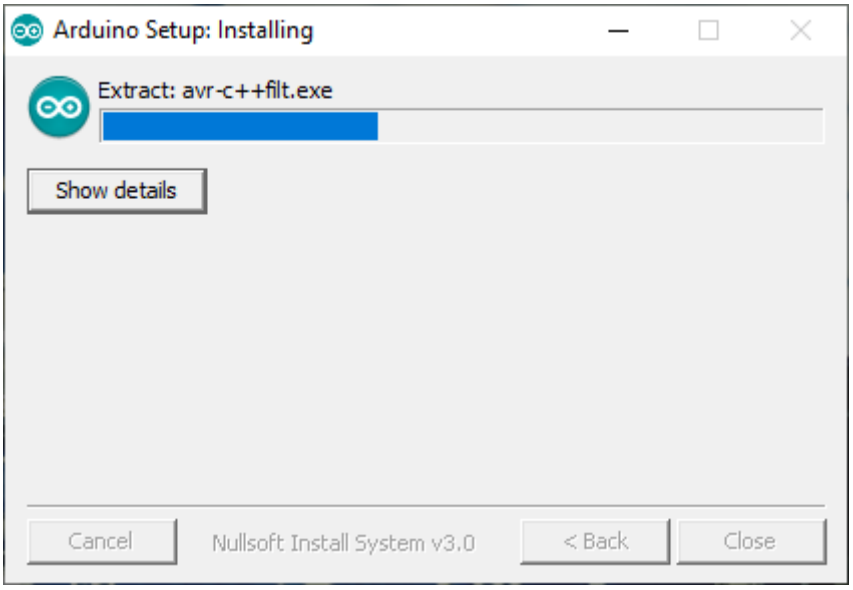
Option	Erklärung
Install Arduino Software	Installs the Arduino IDE - This option can not be deselected
Install USB Driver	Installs USB drivers for various other microcontrollers. These are not required to use the software with the D1 mini, but we strongly recommend installing them if you also use other microcontrollers
Create Start Menu shortcut	Creates a shortcut in the Windows Start menu (optional)
Create Desktop shortcut	Creates a shortcut on the workstation (Optional)

Associate .ino files	Creates a filename extension for files with the extension .ino and links it to the Arduino IDE

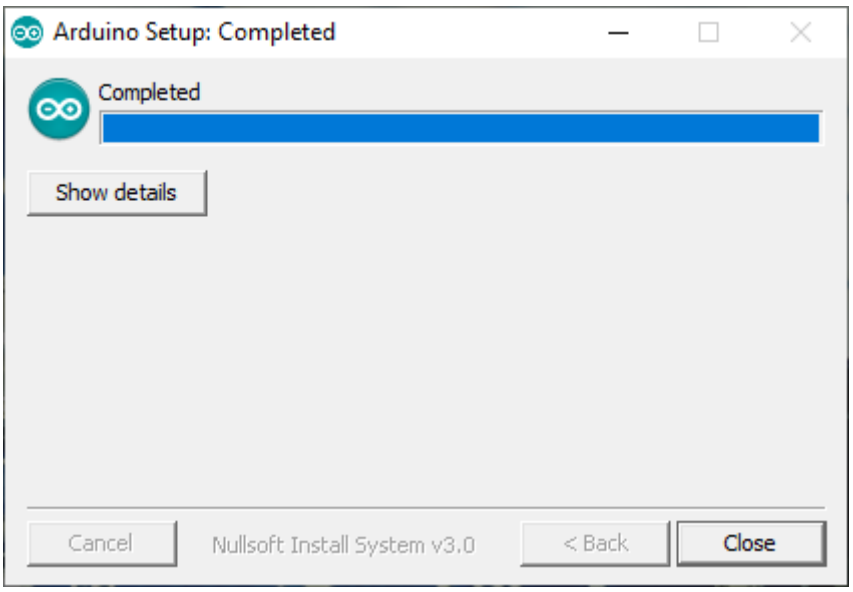
Finally, the destination folder must be specified. The installation requires about 500MB of free disk space.



Click "Install" to start the installation.



After successful installation, the installation program can be closed via the "Close" button:



Set current limit

To set the current limit for an A4988 driver with a rated current of 0.4A and a sense resistor of 0.1Ω, follow these steps:

1/ Calculate Vref

$$Vref = Ilimit \times 8 \times Rsense$$

Target I limit: Use **70–80% of 0.4A** to avoid overheating.

$$Ilimit = 0.4 \times 0.7 \; to \; 0.4 \times 0.8 = 0.28A \; to \; 0.32A$$

Calculate $Vref$:

For $limit = 0.28A$

$$Vref = 0.28 \times 8 \times 0.1 = 0.224V$$

For $limit = 0.32A$

$$Vref = 0.32 \times 8 \times 0.1 = 0.256V$$

So, set $Vref$ **between 0.224V and 0.256V**.

2/ Prepare the A4988

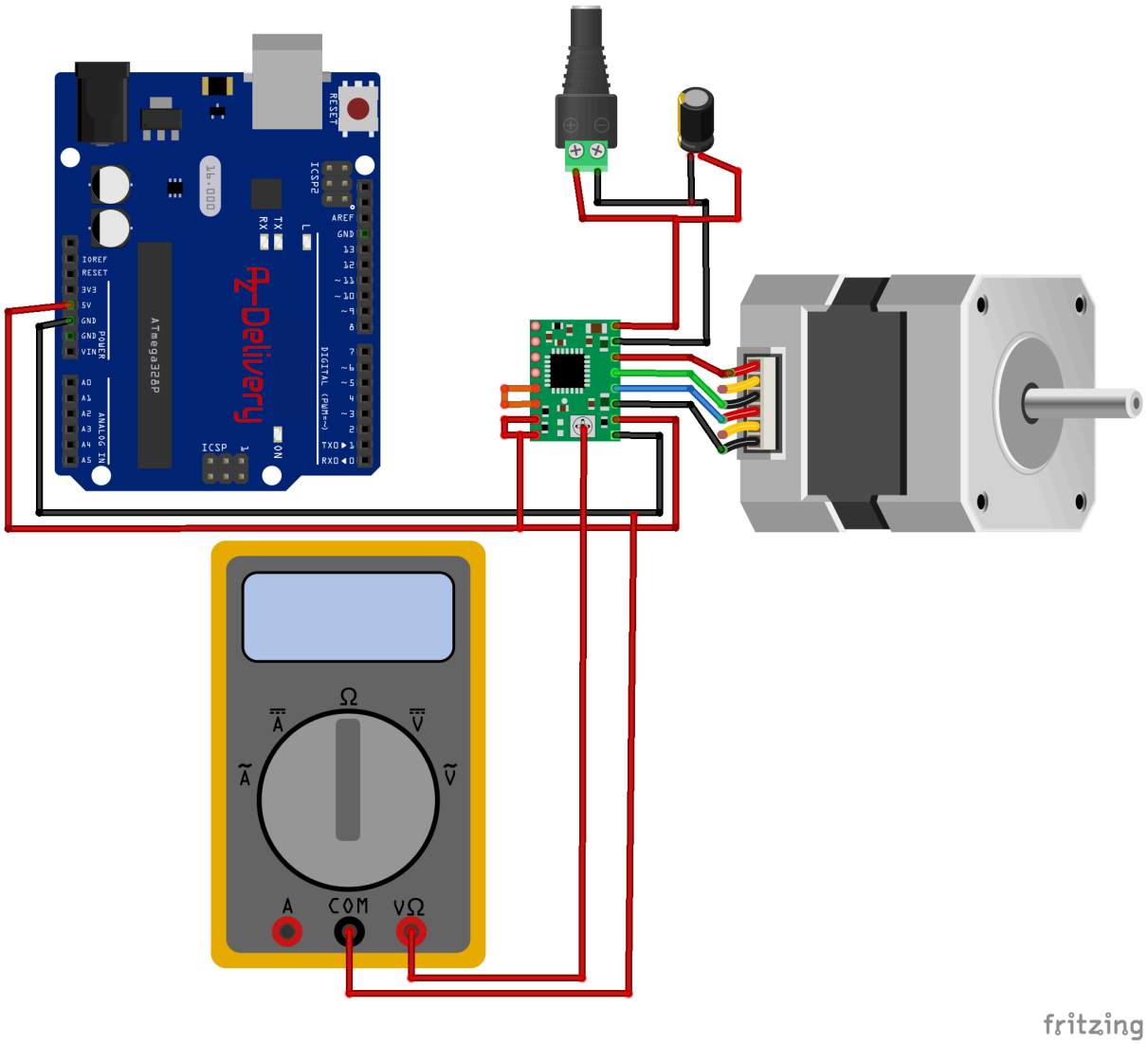
1. Power Off the System:
   Disconnect the motor and the power supply to avoid accidental damage.
2. Connect a Multimeter:
   Set the multimeter to voltage mode (DC).
   Place the multimeter's positive probe on the potentiometer's test point (or the small metal top of the potentiometer).
   Place the negative probe on the GND pin of the driver.
3. Power On the Driver:
   The Arduino provides 5V logic power to the A4988.
   The 12V supply powers the motor circuit (though the motor is disconnected at this point).

3/ Adjust $Vref$:

1. Use a small screwdriver to adjust the **potentiometer** on the A4988:
   - **Turn clockwise** to increase $Vref$.
   - **Turn counterclockwise** to decrease $Vref$.
2. Watch the multimeter as you adjust and stop when $Vref$ is between **0.224V and 0.256V**.

Connection Diagram

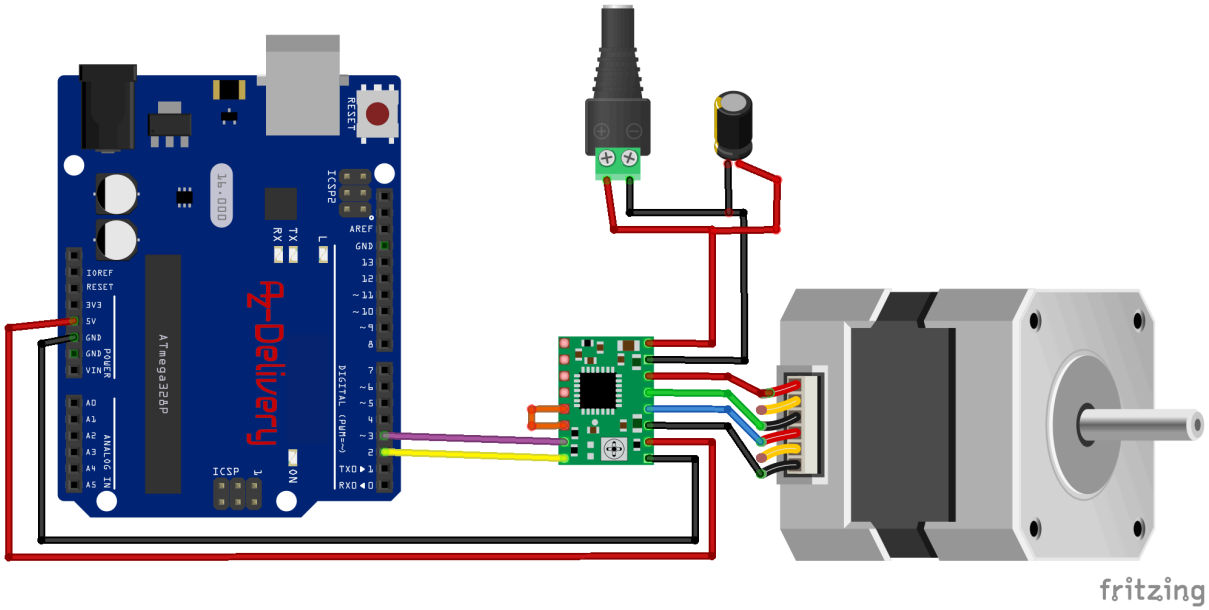
Here’s the schematic you need to follow in order to set the good current limit:



Stepper Motor With Arduino & A4988 Driver

Now let’s connect the motor to the microcontroller. Start by wiring VDD and GND (next to VDD) to the 5V and GND pins. Connect DIR and STEP to digital pins 2 and 3, respectively. Attach the stepper motor to the 1A, 1B, 2A, and 2B pins, ensuring proper coil alignment. To enable the driver, link the RST pin to the SLP pin. Finally, connect the motor power supply to VMOT and GND, and

add a 100µF capacitor across these pins to prevent voltage spikes. Avoid connecting or disconnecting the stepper motor while the driver is powered to prevent damage.



Pinout connection

A4988 Pin	Description	Motor control
VMOT	Motor power supply	Connect to 12V power supply positive (+).
GND	Ground for motor power supply	Connect to 12V power supply ground (-).
VDD	Logic power input	Connect to 5V from Arduino.
GND	Ground for logic power	Connect to Arduino GND.
DIR	Motor direction control input	Connect to Arduino pin 2 (or any GPIO pin).
STEP	Step pulse control input	Connect to Arduino pin 3 (or any GPIO pin).

MS1, MS2, MS3	Microstepping mode selection	Connect to Arduino GPIO pins or to GND/5V for fixed configuration.
1A, 1B, 2A, 2B	Stepper motor coil outputs	Connect to the corresponding motor wires.

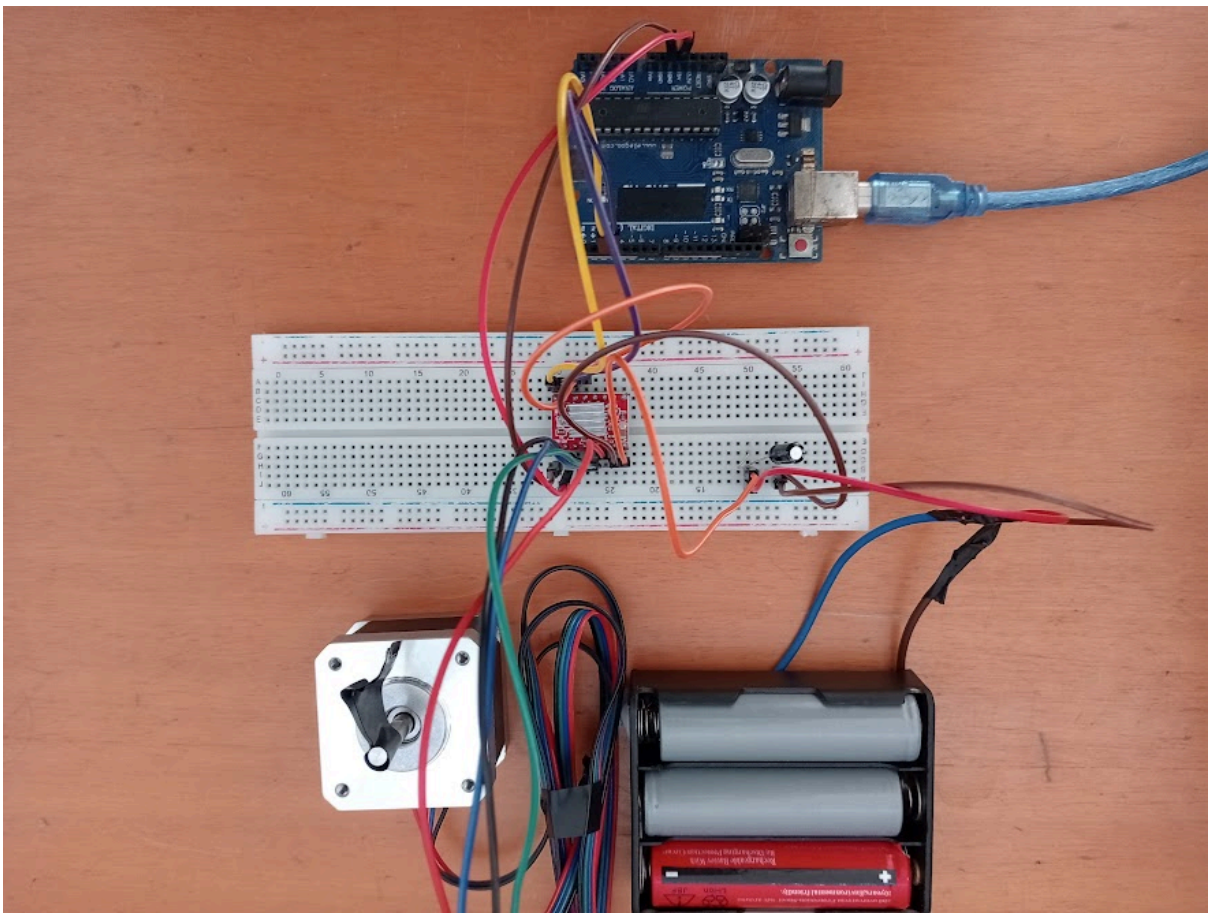
Code Exemple

```
// Define the pins for direction and step signals and set motor steps per
revolution
const int dirPin = 2;       // Pin to control motor direction
const int stepPin = 3;      // Pin to send step pulses
const int stepsPerRevolution = 200; // Steps needed for one full revolution
of the motor

void setup()
{
    // Configure the direction and step pins as outputs
    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);
}

void loop()
{
    // Set the motor to rotate in a clockwise direction
    digitalWrite(dirPin, HIGH);

    // Rotate the motor at a slow speed for one revolution
    for (int x = 0; x < stepsPerRevolution; x++)
    {
        digitalWrite(stepPin, HIGH);  // Generate a step pulse
        delayMicroseconds(2000);     // Delay to control speed (slower)
        digitalWrite(stepPin, LOW);  // End the step pulse
        delayMicroseconds(2000);     // Delay between steps
    }
    delay(1000); // Pause for one second
```

```
    // Set the motor to rotate in a counterclockwise direction
    digitalWrite(dirPin, LOW);

    // Rotate the motor at a faster speed for one revolution
    for (int x = 0; x < stepsPerRevolution; x++)
    {
        digitalWrite(stepPin, HIGH);  // Generate a step pulse
        delayMicroseconds(1000);      // Delay to control speed (faster)
        digitalWrite(stepPin, LOW);   // End the step pulse
        delayMicroseconds(1000);      // Delay between steps
    }
    delay(1000); // Pause for one second
}
```

Demonstration

Before connecting the motor power supply to the module, upload the source code to the microcontroller using the Arduino IDE. Once the upload is complete, connect the motor power supply and observe the stepper motor's movement.



You've done it, you can now use your module for your projects :)

Now it is time to learn and make the Projects on your own. You can do that with the help of many example scripts and other tutorials, which you can find on the internet.
If you are looking for the high quality microelectronics and accessories, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

https://az-delivery.de

Have Fun!

Impressum

https://az-delivery.de/pages/about-us