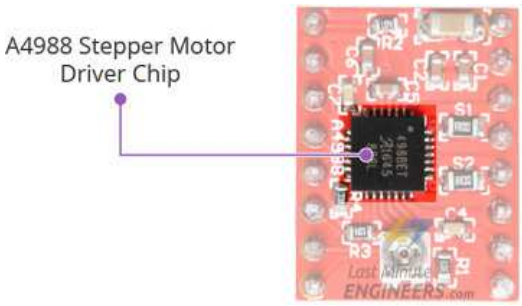
A4988 Stepper Motor Driver Chip

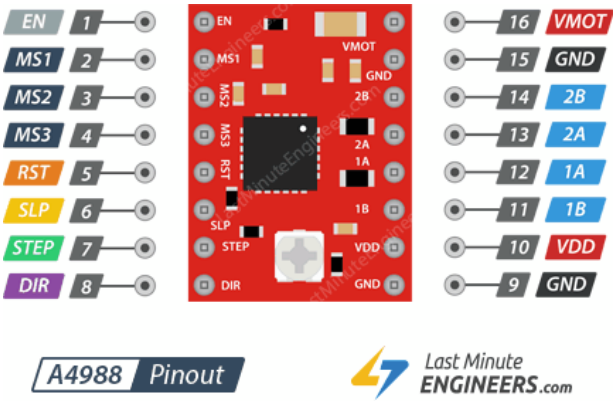
At the heart of the module is a Microstepping Driver from Allegro – A4988. It's small in stature (only 0.8″ × 0.6″) but still packs a punch.

The A4988 stepper motor driver has output drive capacity of up to 35 V and ±2A and lets you control one bipolar stepper motor at up to 2A output current per coil like NEMA 17.

The driver has built-in translator for easy operation. This reduces the number of control pins to just 2, one for controlling the steps and other for controlling spinning direction.



The driver offers 5 different step resolutions viz. full-step, half-step, quarter-step, eighth-step, and sixteenth-step.



A4988 Motor Driver Pinout

The A4988 driver has total 16 pins that interface it to the outside world. The connections are as follows:

Let's familiarize ourselves with all the pins one by one.

Power Connection Pins

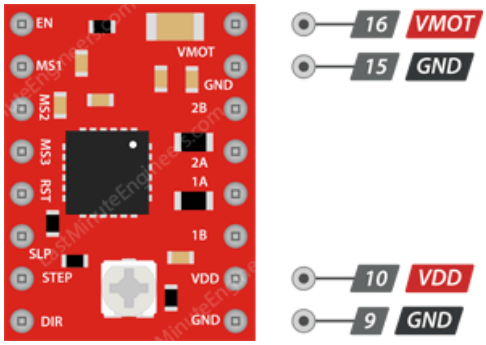
The A4988 actually requires two power supply connections.

VDD & GND is used for driving the internal logic circuitry which can be 3V to 5.5V.

Whereas,

VMOT & GND supplies power for the motor which can be 8V to 35V.

According to datasheet, the motor supply requires appropriate decoupling capacitor close to the board, capable of sustaining 4A.



WARNING

This driver has low-ESR ceramic capacitors on board, which makes it vulnerable to voltage spikes. In some cases, these spikes can exceed the 35V (maximum voltage rating of A4988), potentially permanently damaging the board and even the motor.

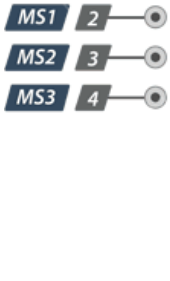
One way to protect the driver from such spikes is to put a large 100µF (at least 47µF) electrolytic capacitor across motor power supply pins.

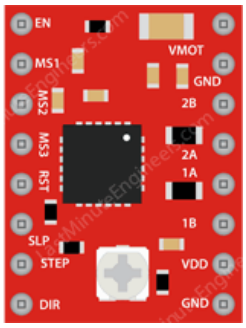
Microstep Selection Pins

The A4988 driver allows microstepping by allowing intermediate step locations. This is achieved by energizing the coils with intermediate current levels.

For example, if you choose to drive NEMA 17 having 1.8° or 200 steps per revolution in quarter-step mode, the motor will give 800 microsteps per revolution.

The A4988 driver has three step size(resolution) selector inputs viz. *MS1, MS2, MS3*. By setting appropriate logic levels to these pins we can set the motors to one of the five step resolutions.





Microstep selection pin logic levels & resolutions			
MS1	MS2	MS3	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	Quarter step
High	High	Low	Eighth step
High	High	High	Sixteenth step

These three microstep selection pins are pulled LOW by internal pull-down resistors, so if we leave them disconnected, the motor will operate in full step mode.

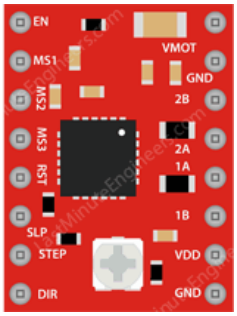
Control Input Pins

The A4988 has two control inputs viz. STEP and DIR.

STEP input controls the mirosteps of the motor. Each HIGH pulse sent to this pin steps the motor by number of microsteps set by Microstep Selection Pins. The faster the pulses, the faster the motor will rotate.

DIR input controls the spinning direction of the motor. Pulling it HIGH drives the motor clockwise and pulling it LOW drives the motor counterclockwise.

If you just want the motor to rotate in a single direction, you can tie DIR directly to VCC or GND accordingly.



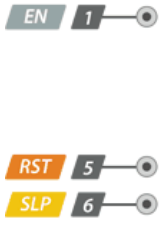
NOTE

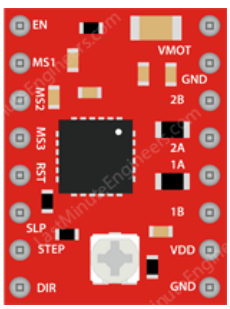
The STEP and DIR pins are not pulled to any particular voltage internally, so you should not leave them floating in your application.

Pins For Controlling Power States

The A4988 has three different inputs for controlling its power states viz. EN, RST, and SLP.

EN Pin is active low input, when pulled LOW(logic 0) the A4988 driver is enabled. By default this pin is pulled low so the driver is always enabled, unless you pull it HIGH.





RST Pin is active low input. Meaning, pulling this pin LOW puts the driver in sleep mode, minimizing the power consumption. You can invoke this especially when the motor is not in use to conserve power.

SLP is also an active low input. When pulled LOW, all STEP inputs are ignored, until you pull it HIGH. It also resets the driver by setting the internal translator to a predefined Home state. Home state is basically the initial position from where the motor starts and it's different depending upon the microstep resolution.

TIP

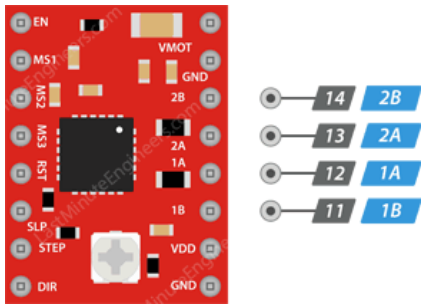
The RST pin is floating. If you are not using the pin, you can connect it to the adjacent SLP/SLEEP pin to bring it high and enable the driver.

Output Pins

The A4988 motor driver's output channels are broken out to the edge of the module with *1A, 1B, 2A, 2B* pins.

You can connect any bipolar stepper motor having voltages between 8V to 35 V to these pins.

Each output pin on the module can deliver up to 2A to the motor. However, the amount of current supplied to the motor depends on system's power supply, cooling system & current limiting setting.



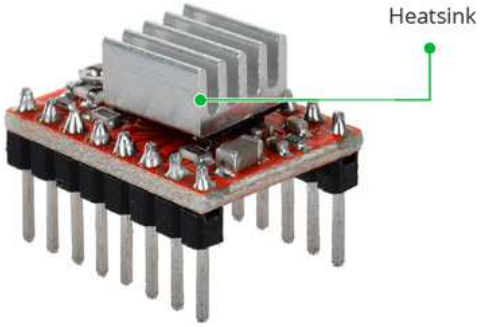
Cooling System – Heatsink

Excessive power dissipation of the A4988 driver IC results in the rise of temperature that can go beyond the capacity of IC, probably damaging itself.

Even if the A4988 driver IC has a maximum current rating of 2A per coil, the chip can only supply approximately 1A per coil without getting overheated.

For achieving more than 1A per coil, a heat sink or other cooling method is required.

The A4988 driver usually comes with a heatsink. It is advisable to install it before you use the driver.

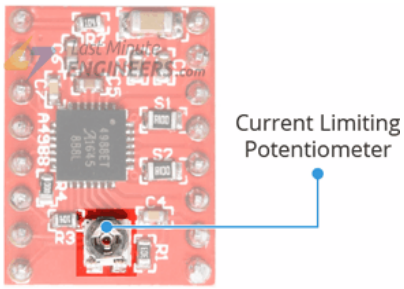


Current limiting

Before using the motor, there's a small adjustment that we need to make. We need to limit the maximum amount of current flowing through the stepper coils and prevent it from exceeding the motor's rated current.

There's a small trimmer potentiometer on the A4988 driver that can be used to set the current limit. In order to set the current limit, you need to follow below steps.

Take a look at the datasheet for your stepper motor. Note down it's rated current. In our case we are using NEMA 17 200steps/rev, 12V 350mA.

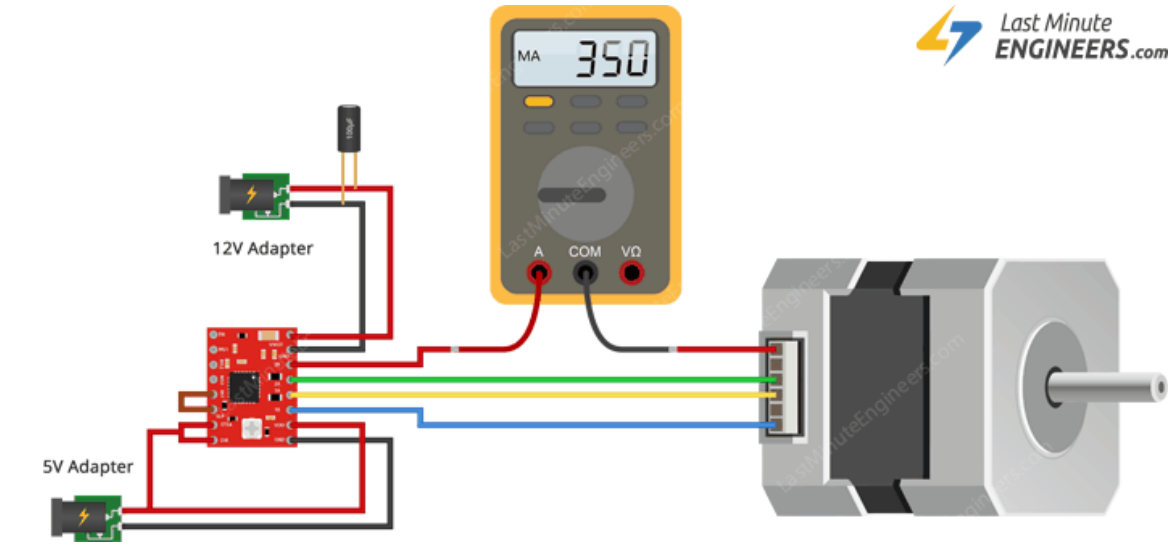


Put the driver into full-step mode by leaving the three microstep selection pins disconnected.

Hold the motor at a fixed position by not clocking the STEP input. Do not leave the STEP input floating, connect it to logic power supply(5V)

Place the ammeter in series with one of the coils on your stepper motor and measure the actual current flowing.

Take a small screwdriver and adjust the current limit potentiometer until you reach rated current.



Measuring Coil Current & Setting Current Limit for A4988 with Multimeter

NOTE

You will need to perform this adjustment again if you ever change the logic voltage(VDD)

Wiring A4988 stepper motor driver with Arduino UNO

Now that we know everything about the driver, we will connect it to our Arduino.

Connections are fairly simple. Start by connecting VDD and GND(next to VDD) to the 5V and ground pins on the Arduino. DIR and STEP input pins are connected to #2 & #3 digital output pins on Arduino respectively.

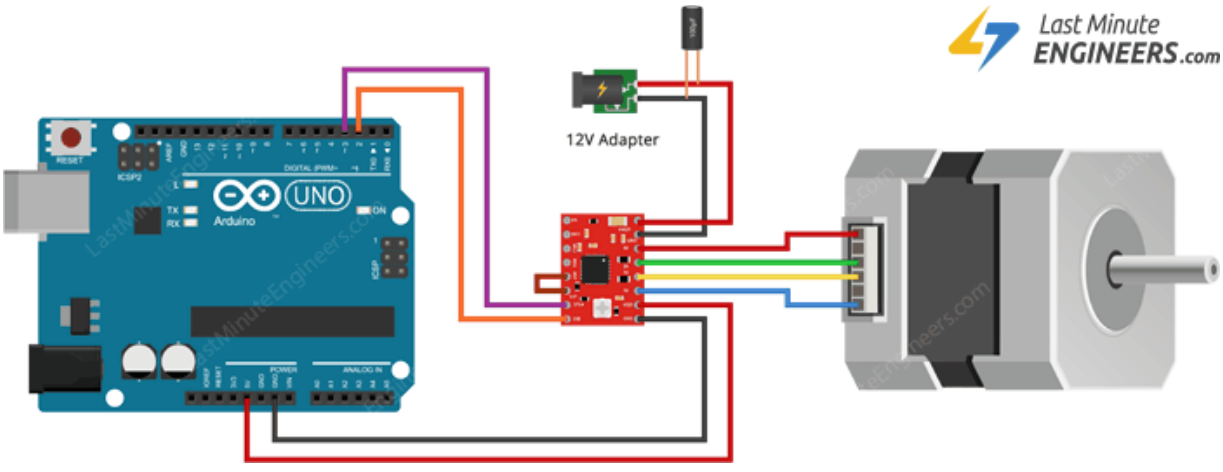
connect the stepper motor to the 2B, 2A, 1A & 1B pins. Actually A4988 is conveniently laid out to match the 4-pin connector on several bipolar motors so, that shouldn't be a problem.

WARNING

Connecting or disconnecting a stepper motor while the driver is powered can destroy the driver.

Next, Connect RST pin to the adjacent SLP/SLEEP pin to keep the driver enabled. Also keep the microstep selection pins disconnected to operate the motor in full step mode.

Finally, connect the motor power supply to the VMOT and GND pins. Remember to put a large 100µF decoupling electrolytic capacitor across motor power supply pins, close to the board.



Wiring Nema 17 Stepper Motor to A4988 driver & Arduino

Arduino Code – Controlling a Stepper Motor

The following sketch will give you complete understanding on how to control speed and spinning direction of a bipolar stepper motor with A4988 stepper motor driver and can serve as the basis for more practical experiments and projects.

```
// Define pin connections & motor's steps per revolution
const int dirPin = 2;
const int stepPin = 3;
const int stepsPerRevolution = 200;

void setup()
{
	// Declare pins as Outputs
	pinMode(stepPin, OUTPUT);
	pinMode(dirPin, OUTPUT);
}
void loop()
{
	// Set motor direction clockwise
	digitalWrite(dirPin, HIGH);

	// Spin motor slowly
	for(int x = 0; x < stepsPerRevolution; x++)
	{
		digitalWrite(stepPin, HIGH);
		delayMicroseconds(2000);
		digitalWrite(stepPin, LOW);
		delayMicroseconds(2000);
	}
	delay(1000); // Wait a second

	// Set motor direction counterclockwise
	digitalWrite(dirPin, LOW);

	// Spin motor quickly
	for(int x = 0; x < stepsPerRevolution; x++)
	{
		digitalWrite(stepPin, HIGH);
		delayMicroseconds(1000);
		digitalWrite(stepPin, LOW);
		delayMicroseconds(1000);
	}
	delay(1000); // Wait a second
}
```

Code Explanation:

The arduino code is pretty straightforward. It doesn't require any libraries to get it working.

The sketch starts with defining Arduino pins to which A4988's STEP & DIR pins are connected. We also define `stepsPerRevolution`. Set this to match your stepper motor specifications.

In setup section of code, all the motor control pins are declared as digital OUTPUT.

In loop section we spin the motor clockwise slowly and then spin it counterclockwise quickly at an interval of a second.

Control Spinning Direction: To control the spinning direction of a motor we set the DIR pin either HIGH or LOW. A HIGH input spins the motor clockwise and a LOW will spin it counterclockwise.

Control Speed: The speed of a motor is determined by the frequency of the pulses we send to the STEP pin. The higher the pulses, the faster the motor runs. A pulses is nothing but pulling the output HIGH, waiting a bit then pulling it LOW and waiting again. By changing the delay between two pulses, you change the frequency of those pulses and hence the speed of a motor.