Arduino and Stepper Motor Configurations

Learn how to control a variety of stepper motors using unipolar / bipolar circuits with Arduino.

Author • Arduino Last revision • 24/02/2024

Stepper motors, due to their unique design, can be controlled to a high degree of accuracy without any feedback mechanisms. The shaft of a stepper, mounted with a series of magnets, is controlled by a series of electromagnetic coils that are charged positively and negatively in a specific sequence, precisely moving it forward or backward in small "steps".

There are two types of steppers, Unipolars and Bipolars, and it is very important to know

example code will control both kinds of motors. See the **unipolar** and **bipolar** motor schematics for information on how to wire up your motor.
The stepper is controlled by with digital pins 8, 9, 10, and 11 for either unipolar or bipolar motors. The Arduino board will connect to a **U2004 Darlington Array** if you're using a unipolar stepper or a **SN754410NE H-Bridge** if you have a bipolar motor.

Hardware Required

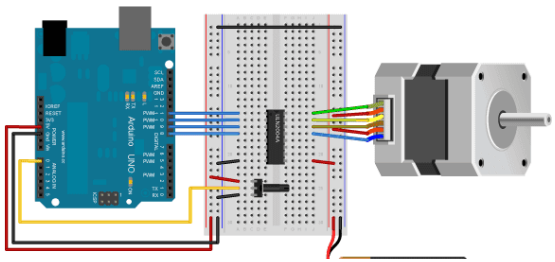
- Arduino Board
- stepper motor
- U2004 Darlington Array (if using a unipolar stepper)
- SN754410ne H-Bridge (if using a bipolar stepper)
- power supply appropriate for your particular stepper
- hook-up wires

Circuit

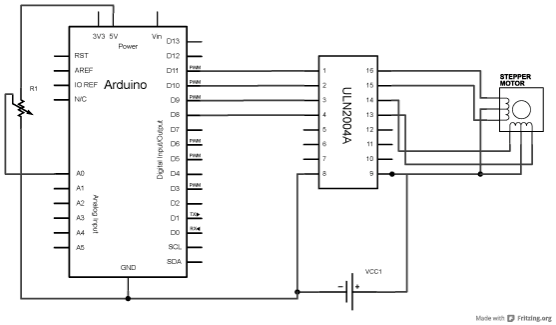
Below you'll find circuits for both unipolar and bipolar steppers. In either case, it is best to power your stepper motors from an external supply, as they draw too much to be powered directly from your Arduino board.

> Note: Both circuits below are four wire configurations. Two wire configurations will not work with the code provided.

Unipolar Stepper Circuit and schematic

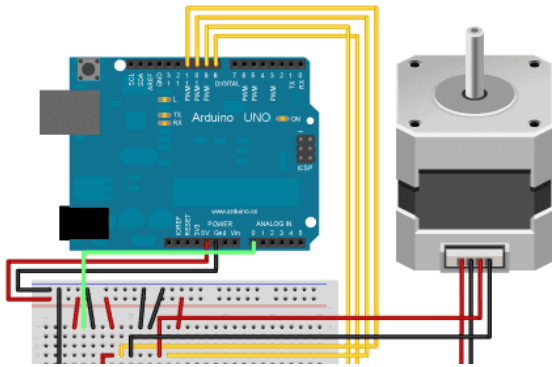


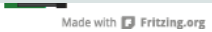
Unipolar Motor Knob Circuit. Image made using Fritzing.



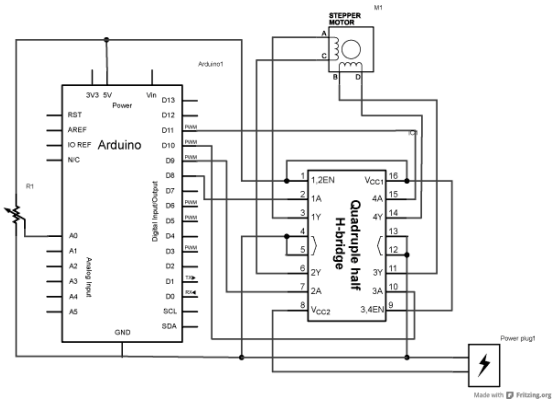
Unipolar Motor Knob Schematic. Image made using Fritzing.

Bipolar Stepper Circuit and Schematic





Bipolar Motor Knob Circuit. Image made using Fritzing.



Bipolar Motor Knob Schematic. Image made using Fritzing.

Examples

MotorKnob

A stepper motor follows the turns of a potentiometer (or other sensor) on analog

StepperOneRevolution

The motor should revolve one revolution in one direction, then one revolution in the other direction.

```
// for your motor

// initialize the stepper libr
Stepper myStepper(stepsPerRevo
```

StepperOneStepAtATime

The motor will step one step at a time, very slowly. You can use this to test that you've got the four wires of your stepper wired to the correctpins. If wired correctly, all steps should be in the same direction.

COPY

```
// for your motor

// initialize the stepper libr
Stepper myStepper(stepsPerRevo

int stepCount = 0;         //

void setup() {
  // initialize the serial por
  Serial.begin(9600);
}

void loop() {
  // step one step:
  myStepper.step(1);
```

StepperSpeedControl

The motor will rotate in a clockwise direction. The higher the potentiometer value, the faster the motor speed. Because setSpeed() sets the delay between steps, you may notice the motor is less responsive to changes in the sensor value at low speeds.

```
const int stepsPerRevolution =
// for your motor


// initialize the stepper libr
Stepper myStepper(stepsPerRevo

int stepCount = 0;  // number

void setup() {
  // nothing to do inside the
}

void loop() {
  // read the sensor value:
  int sensorReading = analogRe
  // map it to a range from 0
  int motorSpeed = map(sensorR
  // set the motor speed:
  if (motorSpeed > 0) {
    myStepper.setSpeed(motorSp
    // step 1/100 of a revolut
    myStepper.step(stepsPerRev
  }
}
```

The content on docs.arduino.cc is facilitated through a public GitHub repository. You can read more on how to contribute in the contribution policy.

Help Center

Ask the Arduino Forum

Discover Arduino Discord