

Ardumoto Quickstart Guide

Gearmotor 100:1 (e.g. ROB-08910 [13])

- Micro Metal Gearmotor 30:1 (e.g. ROB-

by **Pete-O** | June 29, 2010 | *16 comments* **SparkFun** (14) ★ Beginner

- Mini Metal Gearmotor 100:1 (e.g. ROB-08912 [15])

Ardumoto Motor Driver Shield

Thanks for checking out the Ardumoto motor driver shield for Arduino! Combined with an Arduino controller board, the Ardumoto makes a fantastic controller platform for small scale robotics or RC. This page will help to get you going quickly.

to hook up our motors and wire strippers to strip the wire:

[17]

[19]

[20]



What is it?

[18]

The Ardumoto is a board that has an L298. Why do you need a driver board? Well, motors connected straight from your Arduino output pins will not work. You need to control a whole bunch of current (good

for driving small DC motors. The Ardumoto lets you control a motor without needing a lot of pins on your Arduino.



What else do I need?

There's a good chance you've already got it complete (or you're a noob, which is totally fine), but you'll need a 5V Arduino board:

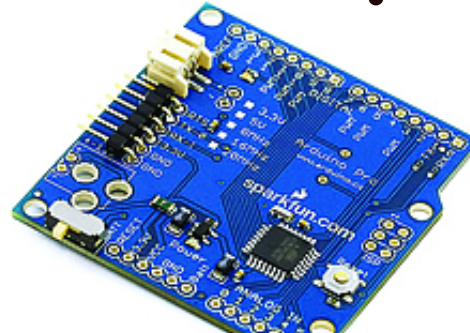
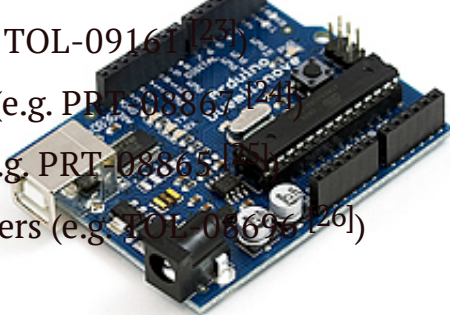


For the sake of being consistent, you'll need an Arduino Uno or Arduino Pro Mini.

[21]

Soldering iron (e.g. TOL-00085 [22])

- Solder (e.g. TOL-09161 [23])
- Black wire (e.g. PRT-08865 [24])
- Red wire (e.g. PRT-08865 [24])
- Wire strippers (e.g. TOL-08696 [26])



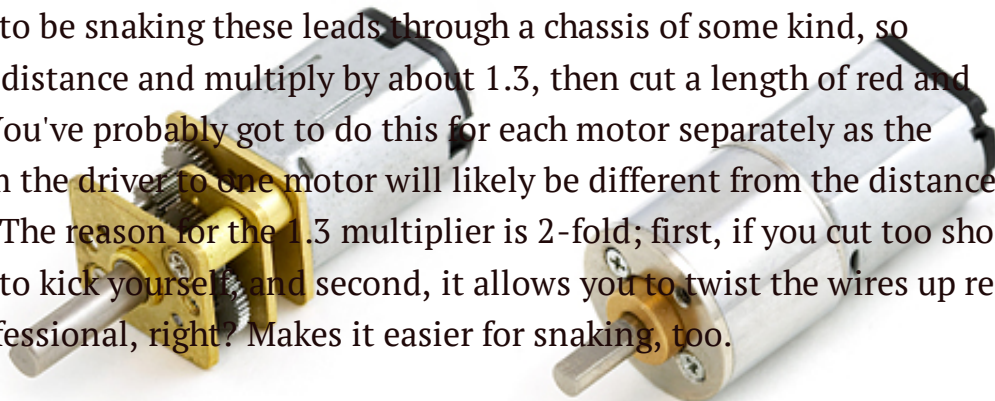
You'll also need a computer to do the programming... no, I'm not going to list computer sources. We'll assume that if you're here you've most likely gone through the basic Arduino tutorials and have programmed one once or twice before this. If you haven't, check out our beginning Arduino tutorial [27].

- Arduino Duemilanove (e.g. DEV-00666)
- Arduino Pro 328 - 5V/16MHz (e.g. DEV-09219)

Sweet. Now how do I make it work?

For the purpose of this tutorial, we'll assume you've got an Arduino Duemilanove to work with. You'll also need some motors to drive. These should probably be motors that draw no more than 1-2 amps when stalled (when it's under power, but you're holding it still). If you're still shopping for those, we have a few smaller motors to choose from:

First, estimate how long you think you need the motor leads to be. We'll assume you're going to be snaking these leads through a chassis of some kind, so estimate the distance and multiply by about 1.3, then cut a length of red and black wire. You've probably got to do this for each motor separately as the distance from the driver to one motor will likely be different from the distance to the other. The reason for the 1.3 multiplier is 2-fold; first, if you cut too short you're going to kick yourself, and second, it allows you to twist the wires up real nice and professional, right? Makes it easier for snaking, too.



These little motors I've used for an example can be a pain to solder. Not because of solder/iron/heat issues, but because these little suckers slide all over the place as soon as they see you coming with an iron in your hand. Our Third Hand vise

- Micro Metal Gearmotor 100:1 (e.g. ROB-08910)
 - Micro Metal Gearmotor 50:1 (e.g. ROB-08911)
- hold them steady while you solder. The 8911 (Micro Metal Gearmotor 50:1) motors listed above actually have round gears, so they'll be even easier to solder.

have round cases, so they'll be even more wiley.

- Mini Metal Gearmotor 24:1 (e.g. [ROB-08913](#))

[30]

For my purposes, I'm going to use a couple of the 30:1 micros, the second one down in the list above.

V and wire strippers to strip



[31]

If you want to look really pro, use some heat shrink ([PRT-08831](#) ^[32]) on your soldered connections.

[33]



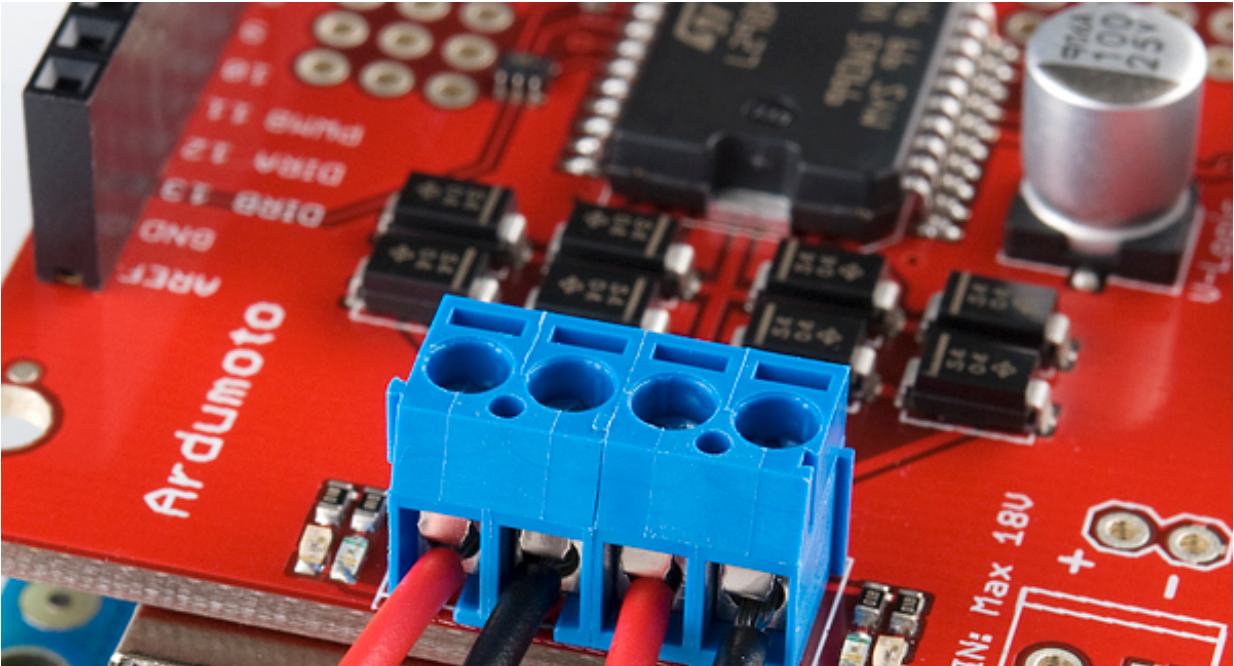


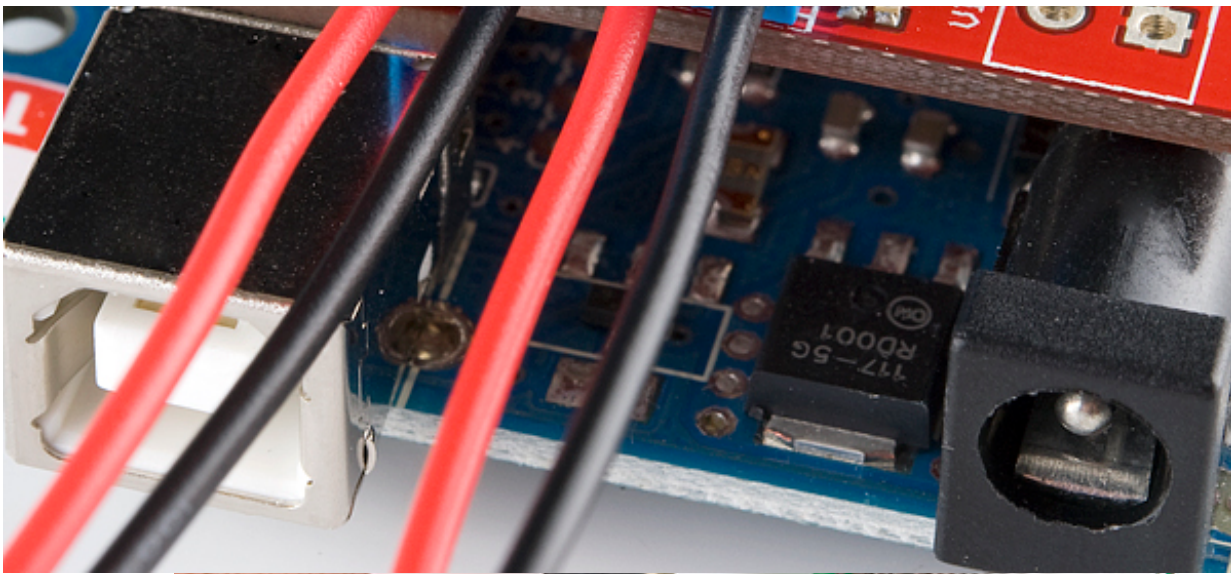
Y
a
p

computer sources. We'll
utorials and have
ing [Arduino tutorial](#).

Sweet. Now how do I make it work?

If you haven't already done so, now would be a good time to [assemble the Ardumoto](#). Once you've don
When you've got both motors set up with leads, you can mount them as you like,
that...
and route the wiring how you like to the place where you're Ardumoto/Arduino
First, estimate how long you think you need the motor leads to be. We'll assume you're going to be
Is going to live (those steps are going to be specific to the user's application, so
snaking these leads through a chassis of some kind, so estimate the distance and multiply by about 1.3,
we'll gloss over that a little bit and focus on the meat of what we're here to do).
then cut a length of red and black wire. You've probably got to do this for each motor separately as the
When you've got that worked out, you can attach the motors to the
distance from the driver to one motor will likely be different from the distance to the other. The reason f
the 1.5 multiplier is 2-fold, first, if you cut too short you're going to kick yourself, and second, it allows
Strip off about 1/8" (5/16" if the wires, solid, right? Makes it easier for making and
Ardumoto/Arduino pair at the screw terminals with a small Phillips screwdriver.
tighten them down (one of the motors to lines 1-2, and the other motor to lines
These little motors I've used for an example can be a pain to solder. Not because of solder/iron/heat issu
3-4)
but because these little suckers slide all over the place as soon as they see you coming with an iron in yc
hand. Our Third Hand vise ([TOL-09317](#)) could be useful here, but I normally do something low-tech lik
h³⁶ it down with the spool of solder. The Mini-motors listed above actually have round cases, so they'l
be even more wiley.

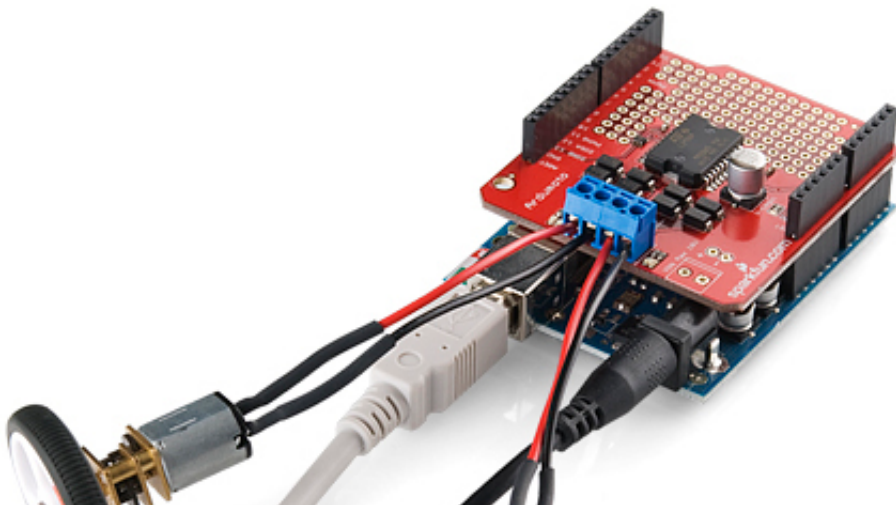


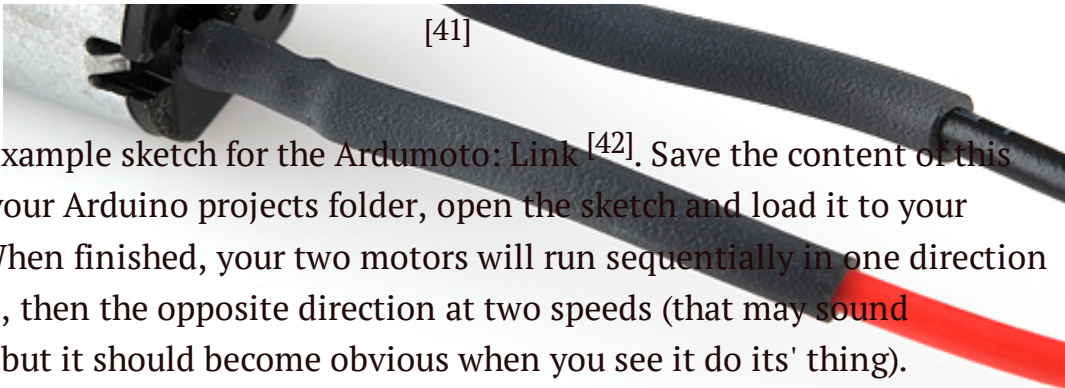


[36]

Getting close now. The Ardumoto takes power from its host Arduino board from Vin to drive the motors and +5V (regulated) to drive the logic on the L298 driver chip. This means two things to you: 1) you can't drive the Ardumoto using the USB connection from an Arduino Duemilanove (only the logic will power up), and 2) you might want to use a power source that's capable of sourcing a couple of amps (depending on your motor choice). We've got a few really good LiPo battery packs that would work really awesome, like PRT-09704 ^[37] (7.4V, 1000mAh) and PRT-09703 ^[38] (7.4V, 2200mAh), but for our demonstration we'll just use a 9V, 650mA wall-wart power supply (TOL-00298 ^[39]) that will just plug into the Duemilanove's barrel jack (we have small motors). Lastly, plug in your USB to your computer and fire up your Arduino development suite.

[40]





[41]

Here's an example sketch for the Ardumoto: [Link](#) ^[42]. Save the content of this zip file to your Arduino projects folder, open the sketch and load it to your Arduino. When finished, your two motors will run sequentially in one direction at 2 speeds, then the opposite direction at two speeds (that may sound confusing, but it should become obvious when you see it do its' thing).

Explanation of the Sketch (Sketchplanation ?)

The example sketch has been designed to be as simple as possible. Painfully so, in fact. Here - I'll prove it. First, we identify what pins will get what signals.

```
//int pwm_a = 10; //PWM control for motor outputs 1 and 2 is on digital pin 10
int pwm_a = 3; //PWM control for motor outputs 1 and 2 is on digital pin 3
int pwm_b = 11; //PWM control for motor outputs 3 and 4 is on digital pin 11
int dir_a = 12; //dir control for motor outputs 1 and 2 is on digital pin 12
int dir_b = 13; //dir control for motor outputs 3 and 4 is on digital pin 13
```

the wiring has to be specified here to do the screw into each line 3-4).

```
void setup()
{
  pinMode(pwm_a, OUTPUT); //Set control pins to be outputs
  pinMode(pwm_b, OUTPUT);
  pinMode(dir_a, OUTPUT);
  pinMode(dir_b, OUTPUT);

  analogWrite(pwm_a, 100);
  //set both motors to run at (100/255 = 39)% duty cycle (slow)
  analogWrite(pwm_b, 100);
```

```

}

void loop()
{
  digitalWrite(dir_a, LOW); //Set motor direction, 1 low, 2 high
  digitalWrite(dir_b, LOW); //Set motor direction, 3 high, 4 low

  delay(1000);

  analogWrite(pwm_a, 255);
  //set both motors to run at 100% duty cycle (fast)
  analogWrite(pwm_b, 255);

  delay(1000);

  digitalWrite(dir_a, HIGH); //Reverse motor direction, 1 high, 2 low
  digitalWrite(dir_b, HIGH); //Reverse motor direction, 3 low, 4 high

  delay(1000);

  analogWrite(pwm_a, 100);
  //set both motors to run at (100/255 = 39)% duty cycle
  analogWrite(pwm_b, 100);

  delay(1000);
}

```

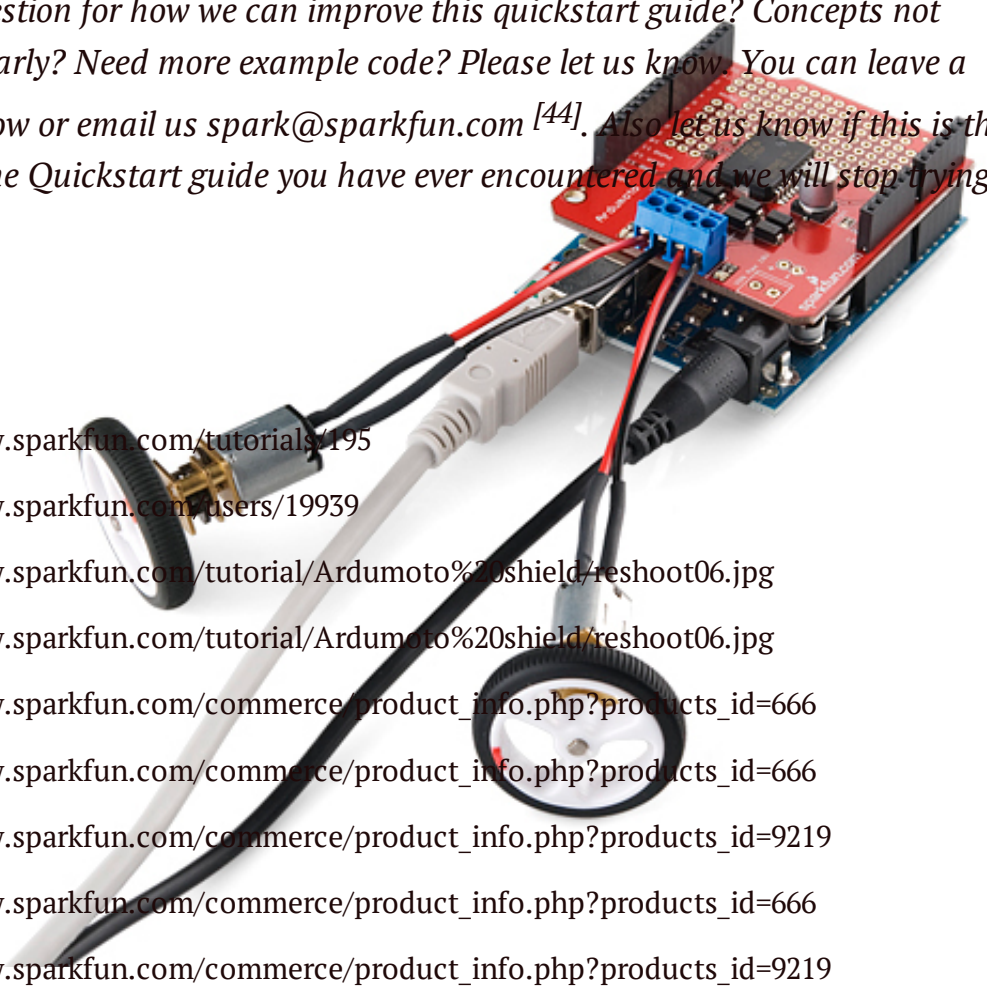
e the motor
 1) you ca
 c will powe
 s (dependir
 on your motor choice). We've got a few really good LiPo battery packs that would work really awesome
 like [PRT-09704](#) (7.4V, 1000mAh) and [PRT-09703](#) (7.4V, 2200mAh), but for our demonstration we'll
 use a 9V, 650mA wall-wart power supply ([TOL-00298](#)) that will just plug into the Duemilanove's barrel
 jack (we have small motors). Lastly, plug in your USB to your computer and fire up your Arduino
 development suite.

corresponds to the motor strapped between outputs 1 and 2, and dir_b
 corresponds to the motor strapped between outputs 3 and 4). The delay
 functions are in milliseconds and are there to provide a little wait-time so the
 user can more easily see the operation of the Ardumoto. The analogWrite
 function is a PWM (Pulse Width Modulation ^[43]) function, and it's this value
 that sets the speed of the motors (pwm_a goes with dir_a and outputs 1 and 2,
 pwm_b goes with dir_b and outputs 3 and 4). This function will accept values
 from 0 (motor's stopped) to 255 (full throttle).

And...that's it! You're off and running (into walls, chairs, cats...) Please drop us a

line and let us know about your projects!

Have a suggestion for how we can improve this quickstart guide? Concepts not explained clearly? Need more example code? Please let us know. You can leave a comment below or email us spark@sparkfun.com ^[44]. Also let us know if this is the most awesome Quickstart guide you have ever encountered and we will stop trying to improve it.

- 
1. <http://www.sparkfun.com/tutorials/195>
 2. <http://www.sparkfun.com/users/19939>
 3. <http://www.sparkfun.com/tutorial/Ardumoto%20shield/reshoot06.jpg>
 4. <http://www.sparkfun.com/tutorial/Ardumoto%20shield/reshoot06.jpg>
 5. http://www.sparkfun.com/commerce/product_info.php?products_id=666
 6. http://www.sparkfun.com/commerce/product_info.php?products_id=666
 7. http://www.sparkfun.com/commerce/product_info.php?products_id=9219
 8. http://www.sparkfun.com/commerce/product_info.php?products_id=666
 9. http://www.sparkfun.com/commerce/product_info.php?products_id=9219

Here's an example sketch for the Ardumoto. <http://www.sparkfun.com/tutorials/195>. Save the contents of this zip file to your Arduino project folder, open the sketch and load it to your Arduino. When finished, your two motors will run sequential in one direction at 2 speeds, then the opposite direction at two speeds (that may sound confusing, but it should become obvious when you see it do its thing).

http://www.sparkfun.com/commerce/product_info.php?products_id=8911

Explanation of the Sketch (Sketchplanation?)

http://www.sparkfun.com/commerce/product_info.php?products_id=8912

The example sketch has been designed to be as simple as possible. Painfully so, in fact. Here - I'll prove First, we identify what pins will get what signals.

http://www.sparkfun.com/commerce/product_info.php?products_id=8911

```
//int pwm_a = 10; //PWM control for motor outputs 1 and 2 is on digital pin 10
int pwm_a = 3; //PWM control for motor outputs 1 and 2 is on digital pin 3
int pwm_b = 11; //PWM control for motor outputs 3 and 4 is on digital pin 11
int dir_a = 12; //dir control for motor outputs 1 and 2 is on digital pin 12
int dir_b = 13; //dir control for motor outputs 3 and 4 is on digital pin 13
```

http://www.sparkfun.com/commerce/product_info.php?products_id=8867

```
void setup()
{
  pinMode(pwm_a, OUTPUT); //Set control pins to be outputs
  pinMode(pwm_b, OUTPUT);
  pinMode(dir_a, OUTPUT);
  pinMode(dir_b, OUTPUT);
  analogWrite(pwm_a, 100);
  //set both motors to run at (100/255 = 39)% duty cycle (slow)
```

```

    analogWrite(pwm_b, 100);
23. http://www.sparkfun.com/commerce/product_info.php?products_id=9161%20
}
24. http://www.sparkfun.com/commerce/product_info.php?products_id=8867
void loop()
25. http://www.sparkfun.com/commerce/product_info.php?products_id=8865
    digitalWrite(dir_a, LOW); //Set motor direction, 1 low, 2 high
26. http://www.sparkfun.com/commerce/product_info.php?products_id=8694
    digitalWrite(dir_b, LOW); //Set motor direction, 3 low, 4 high
27. http://www.sparkfun.com/commerce/tutorial_info.php?tutorials_id=145
    delay(1000);
28. http://www.sparkfun.com/commerce/tutorial_info.php?tutorials_id=183
    //set both motors to run at 100% duty cycle (fast)
29. http://www.sparkfun.com/commerce/product_info.php?products_id=9317
    analogWrite(pwm_b, 255);
30. http://www.sparkfun.com/tutorial/Arduimoto-Quickstart/This%20Guide-03.jpg
    delay(1000);
31. http://www.sparkfun.com/tutorial/Arduimoto-Quickstart/This%20Guide-03.jpg
    digitalWrite(dir_a, HIGH); //Reverse motor direction, 1 high, 2 low
    digitalWrite(dir_b, HIGH); //Reverse motor direction, 3 low, 4 high
32. http://www.sparkfun.com/commerce/product_info.php?products_id=8831
    delay(1000);
33. http://www.sparkfun.com/tutorial/Arduimoto-Quickstart/This%20Guide-04.jpg
    analogWrite(pwm_a, 100);
34. http://www.sparkfun.com/tutorial/Arduimoto-Quickstart/This%20Guide-04.jpg
    //set both motors to run at (100/255 = 39)% duty cycle
    analogWrite(pwm_b, 100);
35. http://www.sparkfun.com/tutorial/Arduimoto%20shield/reshoot08.jpg
    delay(1000);
}
36. http://www.sparkfun.com/tutorial/Arduimoto%20shield/reshoot08.jpg
37. http://www.sparkfun.com/commerce/product_info.php?products_id=9704

```

The function of each line is likely self explanatory, but the way the jobs get done may not be if you're new to Arduino. The digitalWrite function sets a pin either high or low, and is used here for controlling the direction of the motors (dir_a corresponds to the motor strapped between outputs 1 and 2, and dir_b corresponds to the motor strapped between outputs 3 and 4). The delay functions are in milliseconds and are there to provide a little wait-time so the user can more easily see the operation of the Arduimoto. The analogWrite function is a PWM (Pulse Width Modulation) function, and it's this value that sets the speed of the motors (pwm_a goes with dir_a and outputs 1 and 2, pwm_b goes with dir_b and outputs 3 and 4). This function will accept values from 0 (motor's stopped) to 255 (full throttle).

43. http://en.wikipedia.org/wiki/Pulse-width_modulation
 And...that's it! You're off and running (into walls, chairs, cats...) Please drop us a line and let us know about your projects!

Have a suggestion for how we can improve this quickstart guide? Concepts not explained clearly? Need more example code? Please let us know. You can leave a comment below or email us spark@sparkfun.com. Also let us know if this is the most awesome Quickstart guide you have ever encountered and we will stop trying to improve it.

SparkFun Electronics ® | [Boulder, Colorado](#) | [Customer Service](#)

