05 - H-Bridge Motor Control

So, you want to control a motor with your Arduino? If you want to control both speed and direction, you need an H-Bridge. This is a nifty integrated circuit (IC) that will allow us to control this feature.

SKILL REQUIREMENTS

ELECTRICAL PROTOTYPING	■□□□□
ROBOTICS	□□□□□
SOLDERING	□□□□□
PROGRAMMING	■□□□□
DIY	□□□□□

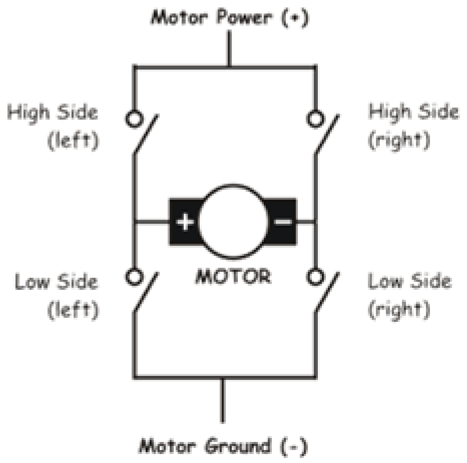
MATERIALS LIST

- Computer
- Arduino UNO
- USB Cable
- Motor
- Breadboard
- H-Bridge (SN754410)

STEP 1: Wiring / Background

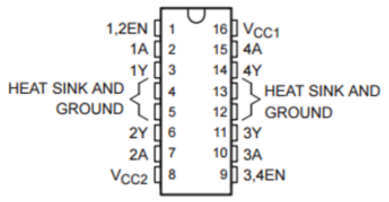
An H-Bridge is basically a set of 4 switches that can be controlled by an Arduino. Using a couple simple digital OUTPUT signals, you can open or close these switches. By selectively choosing which switch you close, you can control the direction the motor spins.

For example, if you close the switches on the (left) High Side and the (right) Low Side, the motor spins ***clockwise***. If you close the (right) High Side and the (left) Low Side, you change the direction of current through the motor and the motor spins ***counter-clockwise***. Remember that the Arduino can only output about 40 mA? The H-Bridge chip allows you to the small signal / low-current output of the Arduino pins to drive a motor. Using the SN754410 IC, you can drive up to 1 A of current through a motor in any direction.



STEP 2: The SN754410 IC

The SN754410 is a dual H-Bridge integrated circuit (IC). This means that inside the little black chip are actually two sets of H-Bridges. This is perfect for spinning / controlling up to two motors. Let's look at the pins on the IC. Here's what all of the pins do.



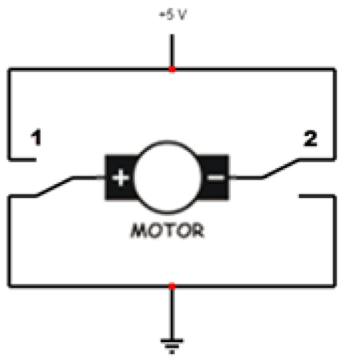
FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

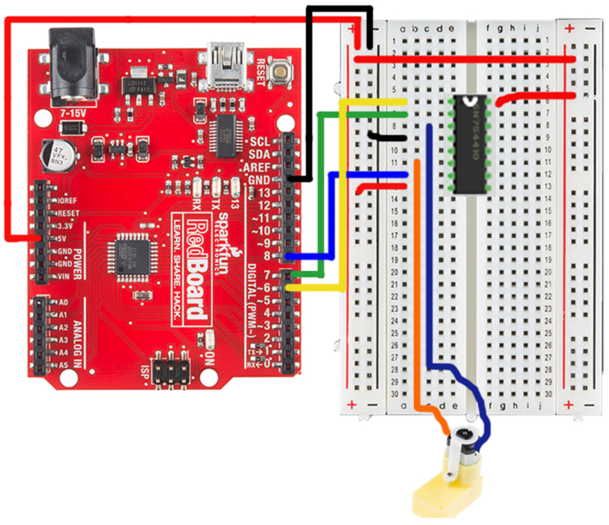
H = high-level, L = low-level
X = irrelevant
Z = high-impedance (off)
† In the thermal shutdown mode, the output is in a high-impedance state regardless of the input levels.

Vcc1	5V: This is power for the chip and logic.
HEAT SINK AND GROUND	GND: Connect any of the 4 pins to gnd.
Vcc2	5V: This is motor power. You can connect this up to any voltage up to 36V. For now, we're just going to use 5V.
1,2EN and 3,4EN	These are the enable pins for the first H-Bridge (1,2) and the second H-Bridge (3,4). Using a PWM output on Arduino, you can pulse the enable pin and control the speed of the motor.
1A, 2A, 3A, 4A	Inputs to control the switches. 1A controls the output on 1Y, 2A controls the output on 2Y, etc...
1Y, 2Y, 3Y, 4Y	Outputs to the motor. These should be paired together 1,2 and 3,4 for each motor.

In the original drawing of an H-Bridge, if you closed both the (left) High Side and (left) Low Side, you would have a direct short circuit from Power to ground. This would be bad! In actuality, the H-Bridge really looks more like this. The two controls (1A or 2A) switch the output to either HIGH level or a LOW level. You can't set one side to both HIGH and LOW at the same time.



STEP 3: Wiring



Notice that the H-Bridge chip has a small notch on one end. Place this on the breadboard so that it is up or away from you. You first need to supply power for the IC. Connect 5V to **pin 16** (upper right corner) and GND to any of the GROUND pins. We chose to use the 4th pin from the upper-left corner.

Finally, the Motor power is on a separate pin (V_{CC2}). This is the pin on the lower-left corner of the IC. For this project, connect this to 5V. For higher power applications, you might connect your Arduino Board up to an external power source like a 12V battery, and you could connect the V_{CC2} pin to VIN on the Arduino board.

To control each motor, you need a total of three OUTPUT signals from the Arduino. Two OUTPUT signals will be used to control / set the direction of the motor, and the third will be used to control the speed.

Wire up the two pins from the Arduino to control direction to 1A (pin 2) and 2A (pin 7) on the H-Bridge. You are going to use pins 7 & 8 on the Arduino Board for this. Finally, connect the 1.2EN (pin 1) on the H-Bridge to pin 6 of the Arduino. Pin 6 on the Arduino has a ~ meaning that it has PWM capability, and it is going to be used for your speed control.

Finally, connect the motor to the 1Y (pin 3) and 2Y (pin 6) outputs of the H-Bridge.

STEP 4: Example Code

To spin the motor, we need to set one of the direction pin OUTPUTs to HIGH and the other direction pin OUTPUT to LOW. This simple example spins the motor clockwise for 2 seconds and then spins the motor counter-clockwise for 2 seconds. It loops this continuously over and over as an example.

```
// simple h-bridge test code.
void setup()
{
   pinMode(6, OUTPUT); // PWM speed
control
   pinMode(7, OUTPUT); // 1A
   pinMode(8, OUTPUT); // 2A
}

void loop()
{
// set a motorSpeed variable.
   int motorSpeed = 100;
```

```
   // spin motor clockwise
   digitalWrite(7, LOW);
   digitalWrite(8, HIGH);
   analogWrite(6, motorSpeed);

   // wait for 2 seconds
   delay(2000);

   // spin motor counter-clockwise
   digitalWrite(7, HIGH);
   digitalWrite(8, LOW);
   analogWrite(6, motorSpeed);
   delay(2000);
}
```

STEP 5: Braking
What about stopping? Well, the easy way is to just set the motorSpeed to 0. All motors, however,will continue to spin for just a bit longer because of inertia. Setting the motorSpeed to 0 will cause the motor to coast to a stop. If we need the motor to stop abruptly or immediately, we need to apply a *brake* to the motor. We can do this by setting 1A and 2A to LOW. This shorts the motor.

Let's see what happens when you short the motor. Working with a partner, take out one of the motors and give the motor a good hard spin with your fingers. Feel how it spins? Now, have your partner connect the two wires of the motor together. Now, spin the motor again. Does it feel different? This is because the electricity generated by the motor is actually working against the motor itself! This is a great example of using the motors to brake.

STEP 5: Serial Motor Control
Similar to the example we used with the servo motor, here is a quick example for using the Serial Monitor to control the speed of a single motor. Open up and upload this example sketch to your board: codebender.cc/sketch:**137383**

After uploading this example, open up the Serial Monitor. Type in a speed from 0 to 255. This example only drives the motor in one direction. Can you modify the code so that if the number is positive, it drives the motor clockwise, and if it's negative, it drives the motor in the opposite direction?

(Need a hint? Check out our example at: codebender.cc/sketch:**137437**)

TAKING IT FURTHER - Wiring up a second motor

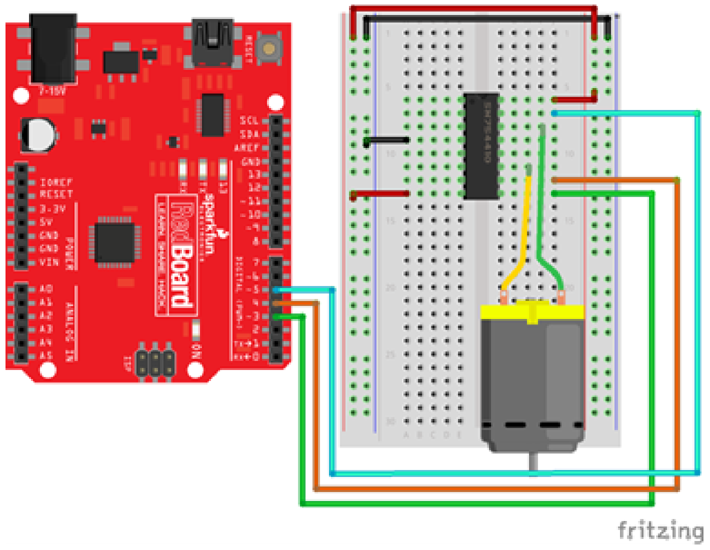
Note: We only show the use of the right side of the H-Bridge in this diagram for clarity.

- Using pins 3, 4, & 5 on the Arduino, connect these up to the 3,4EN, 3A and 4A, pins on the H-Bridge

- Finally, connect up your second motor to the 3Y and 4Y pins on the H-Bridge.

- Adapt the example code from above. Make sure you initialize the pins 3, 4, and 5 as OUTPUTs using the pinMode() function in your setup().

- Why would you want two motors? How about building a little robot? Find a box, platform, or a piece of cardboard. Tape the motors to this, add some wheels, and see what kinds of dance routines you can program your robot to do!



Visit us at: learn.sparkfun.com

DRAFT- UNRELEASED