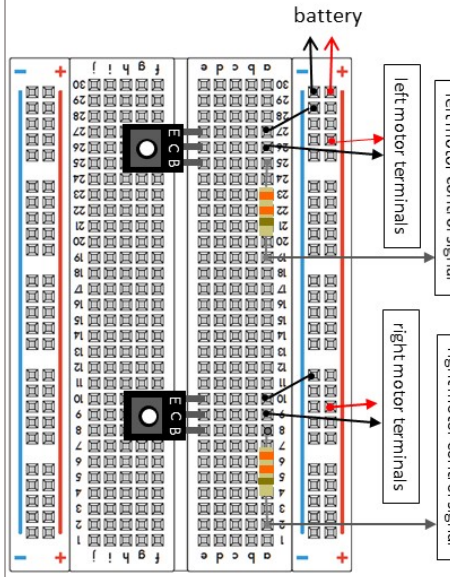
Name/NetID:	Partner:

Module: H-Bridge – Going Backwards

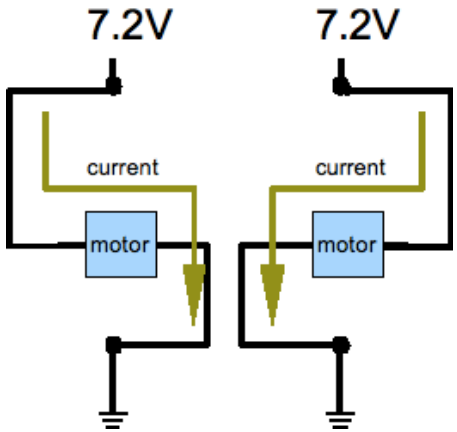
Introduction

What happens if you want to go backwards sometimes – not just forward? In your motor drive circuit you have each DC motor connected to a transistor so that the motors can be turned on and off by a voltage supplied by the Arduino or other external circuitry. The transistor allows the motor to be driven by a battery capable of providing enough power while being controlled by small-signal devices.

The figure to the right shows a cirucuit mounted on a proto-board with the ***bare essentials*** needed to drive two DC motors. The power is delivered to the motors using the unregulated 7.2 or 9V supplied directly from the battery, or if an Arduino board is used the voltage available on the V_{in} Pin. A transistor acting as a switch is connected to the motors at the collector. The state of the transistor is controlled by a signal supplied from the Arduino board or sensor output which is powered by a regulated 5V.



A single transistor can only connect the DC motor so the current flows in only one direction. To have a single motor run in both directions an H-bridge (why it is called this will become clear in a moment) is commonly used. The H-bridge uses digital signals to control the ON/OFF state of the motor and the FORWARD/BACKWARD state as shown in Figure 1. The unusual shape of the connection will help you see how the H-bridge works.



Notes:

***Figure 1**: A simple setup showing 2 motors each connected to a 7.2V battery.*

The next figure was drawn to illustrate how an H-bridge works. Four switches have been inserted between the terminals of the 7.2V battery and ground with one of the motors connected in the center. There are 16 possible ways to open and close the four switches, only two of which are of interest. One of the interesting configurations is shown to the right where switch 1 and switch 4 are closed. In this configuration the current is flowing through the motor in one direction. If switch 1 and 4 are open but switch 2 and switch 3 are closed instead you can see that the current will flow through the opposite direction. Of the other 14 states we will only say that some of them are completely useless and some are downright destructive. This is what an H-bridge does – it is a device where the switches are transistors and internal logic makes certain that only the two desirable states are ever accessed.

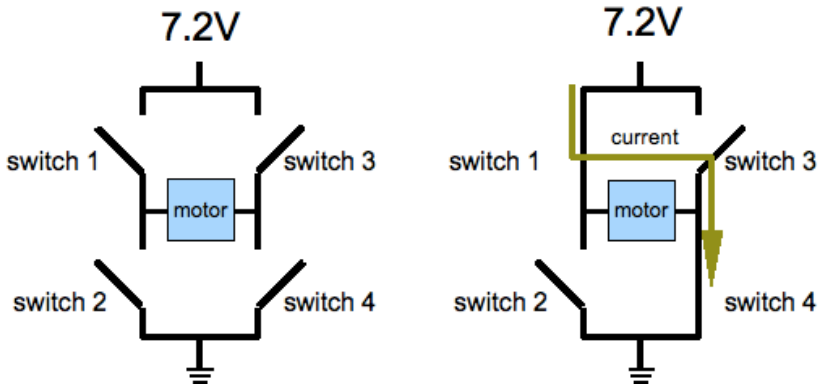


***Figure 2**: A simple setup showing 2 motors each connected to a 7.2V battery.*

Notes:

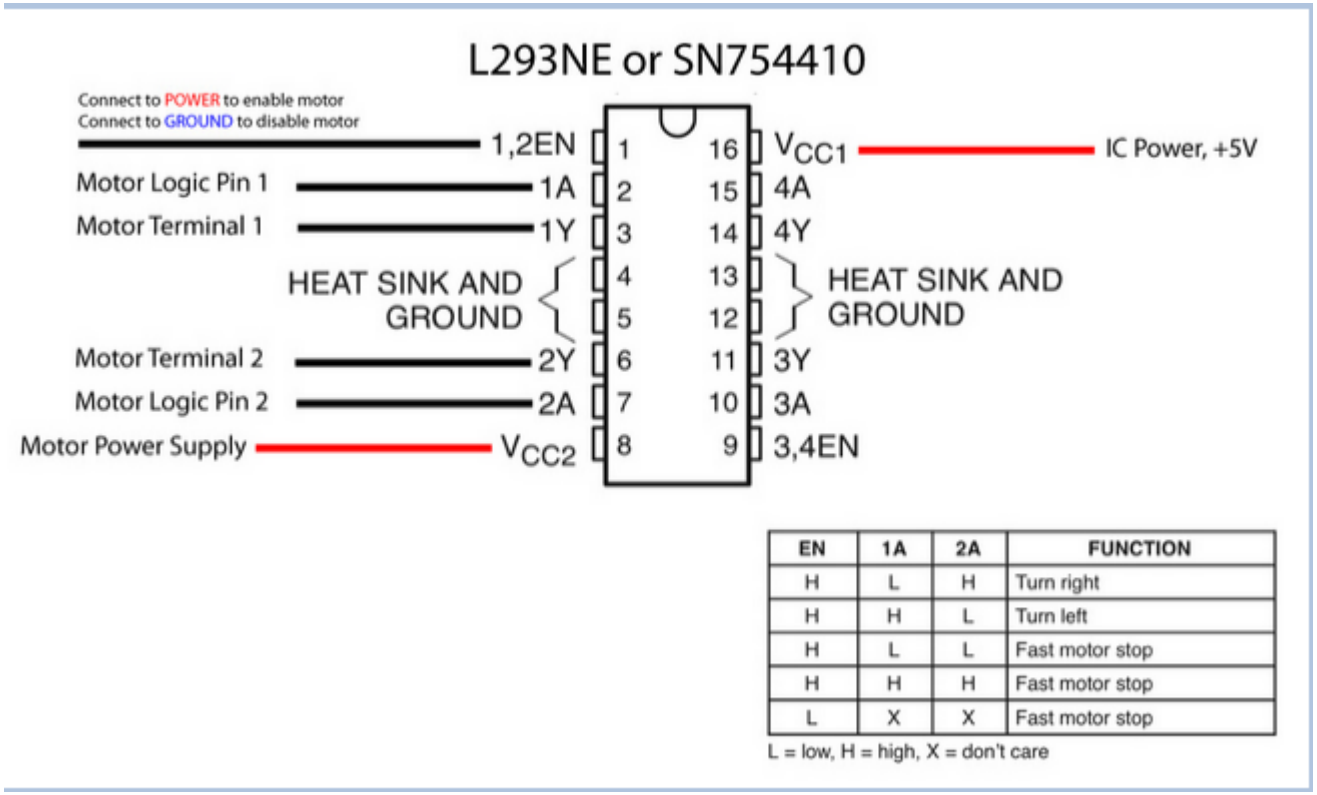
Question 1: The table below enumerates all 16 possible combinations – only 2 are useful. *Fill in the circuit behavior column.* One is done as an example.

Switch 1	***Switch 2***	***Switch 3***	***Switch 4***	***Circuit Behavior***
open	open	open	open	
open	open	open	closed	
open	open	closed	open	
open	open	closed	closed	
open	closed	open	open	
open	closed	open	closed	
open	closed	closed	open	
open	closed	closed	closed	Motor ponders it's untimely demise.
closed	open	open	open	
closed	open	open	closed	
closed	open	closed	open	
closed	open	closed	closed	
closed	closed	open	open	
closed	closed	open	closed	
closed	closed	closed	open	
closed	closed	closed	closed	

Notes:

Wiring the H-Bridge

In your kits you should have an H-bridge chip SN754410 – each H-bridge chip can drive two motors so you only need 1 chip. The H-bridge is designed to replace transistor and biasing resistors that you have been using as the motor drive circuitry not only adding additional functionality by allowing the motors to spin in either direction but also taking less real estate on your protoboard.



EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

The figure above shows the pinouts for the H-bridge chip.

POWER - The first thing to notice is that there are two pins that provide power, one labeled V_{cc1} (pin 16) and the other V_{cc2} (pin 8). All of the control circuitry on the chip that is responsible for switching the direction of current flow allowing the motor to run

Notes:

in both directions is fabricated using a technology that needs V_{cc1} =5V to work correctly. The second is the voltage used to drive the motors – so far you have been using 7.2V or 9V batteries to power the motors provided directly or via the V_{in} pin on the Arduino board. In the middle of the chip there are 4 pins (pins 4, 5, 12, 13) dedicated to ground. All four must be connected together and to the negative terminals of BOTH batteries/power sources.

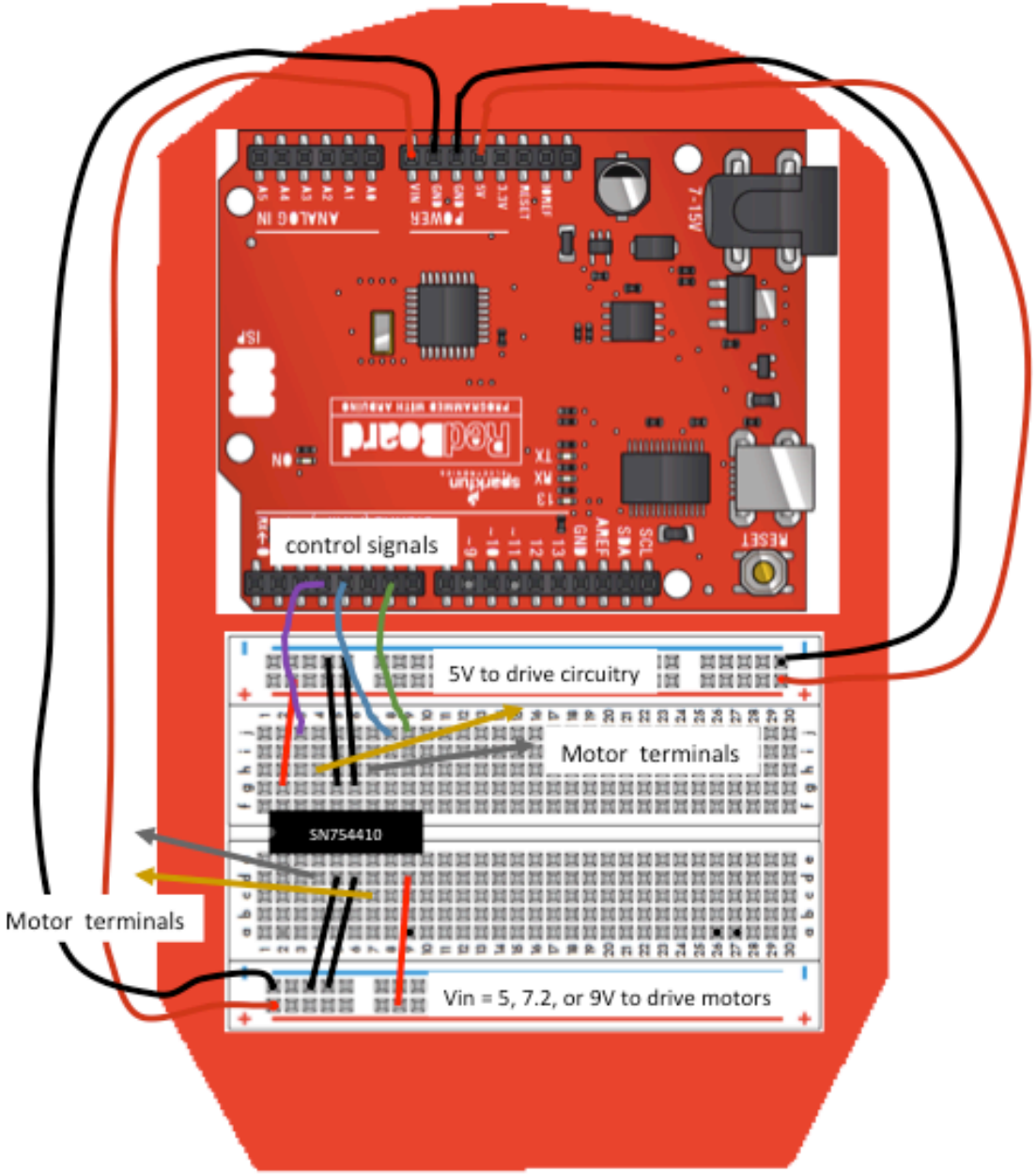
ENABLE - As you can see there are two H-bridges so one chip can control 2 motors. Each side has ENABLE pins (pins 1,9). These pins are used for more sophisticated designs where multiple devices are sharing common control signals. Your application will not need this flexibility so these pins are always set HIGH. Or are they… The control signal can be used to control the speed of the motor by enabling and disabling the circuitry. So you do not have to gain the ability to go backwards at the expense of being able to control the speed of the motor using Pulse Width Modulation.

Motor Terminal Connection – the pins 1Y and 2Y (pins 3 and 6) are the two pins that are connected to the terminals of the motor. Pins 3Y and 4Y (pins 11 and 14) are the same connections to the second H-bridge. Be sure to connect the two motors with the same polarity – if pin 1Y is connected to the positive terminal of one of the motors connect pin 3Y to the positive terminal of the other.

Control Signals – pins 1A and 2A (pins 2 and 7) are the pins used to control the direction of spin of one motor. The pins 4A and 3A are the same signals for the second H-bridge. It is these signals that we are most interested in. The table at the bottom specifies the behavior of the for different digital voltage levels – H implies that the voltage at the pin is HIGH or 5V, L implies that the voltage at the pin is LOW or 0V, and an X means that is doesn't matter. The designation *turn left*, and *turn right* are arbitrary descriptions of the two direction the motor spins.

Notes:

This figure illustrates how you interface the H-bridge circuit to the Arduino. Note: Only one set of 3 control signal connections are shown for simplicity.



Notes:

Controlling the Motors

This is a very nice chip to use for this application because it can carry enough current to drive the motors included in your kits. Now that you have connected your motor(s) to the H-bridge you will find that running the motor forward or backwards is very simple if you follow a few rules. To illustrate let's look at the necessary Arduino code to have the motors turn both in the same direction.

```
const int Motor1Direction1=3;
const int Motor1Direction2=4;
const int Motor1Speed=6;

const int Motor2Direction1=12;
const int Motor2Direction2=11;
const int Motor2Speed=9;

// the setup function runs once when you press reset or power the board
void setup() {

  pinMode(Motor1Direction1, OUTPUT);
  pinMode(Motor1Direction2, OUTPUT);
  pinMode(Motor1Speed,OUTPUT);

  pinMode(Motor2Direction1, OUTPUT);
  pinMode(Motor2Direction2, OUTPUT);
  pinMode(Motor2Speed,OUTPUT);

}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(Motor1Direction1, HIGH);   // Motor 1 controls
  digitalWrite(Motor1Direction2, LOW);
  analogWrite(Motor1Speed,110);

  digitalWrite(Motor2Direction1, LOW);   // Motor 2 controls
  digitalWrite(Motor2Direction2, HIGH);
  analogWrite(Motor2Speed,90);
}
```

Notes:

Setup loop –

The constants Motor1Direction1, Motor1Direction2, and Motor1Speed are the control signals that set the direction and speed of which ever motor you name motor 1. These signals are connected to the Arduino digital I/O pins 3,4, and 6, all of which are designated as OUTPUT pins – same for the other motor.

Loop loop -

In this section by setting the control signals you can set the direction of each motor and its speed. The program above causes my car to move forward in a straight line, but my motors run do not run at the same speed so to keep going straight one of the motors is directed to move faster. Also depending on how your motors are wired and how you connected the control signals you may need to play around with the MotorDirection signals. Remember – the two control signals responsible for setting the direction of the motor spin must be set to different values – if they are either both HIGH or both LOW the motors stops – which could also be useful.

Copy the code into a new sketch, download it to the Arduino (RedBoard) and try it out.