

# RoboJackets Electrical Training Week 2 - Lab Guide

Varun Madabushi

October 1, 2018  
v1.0

## Contents

<b>1</b>	<b>Background</b>	<b>2</b>
<b>2</b>	<b>Materials</b>	<b>2</b>
<b>3</b>	<b>Transistors</b>	<b>2</b>
<b>4</b>	<b>H-Bridge Function</b>	<b>3</b>
<b>5</b>	<b>Programming</b>	<b>5</b>
5.1	Forwards and Backwards . . . . .	5
5.2	Speed Control . . . . .	5

# 1 Background

In the lecture, you learned some basics of different motors and how they are controlled. In this lab, you will implement a DC Motor control circuit using transistors to make an H-Bridge. As you know, an H-bridge is a circuit that allows you to control the direction of potential across a motor, thus allowing it to flow in different directions. With this, you can make your motor rotate in a clockwise or counter-clockwise direction.

Your first goal is to produce a circuit that accomplishes the following:

- Turns a motor on and off based on the position of a 2-position mechanical switch (as the input to your microcontroller)
- Makes the motor rotate forwards or backwards based on the position of another 2-position mechanical switch (also as an input to the microcontroller)

The extension to this lab involves using PWM to control the speed of the motor. PWM allows you to control the effective voltage by generating pulses of the source voltage of varying width.

Once your circuit from the first part is working successfully, update your program so it:

- Turns a motor at either a fast or slow rate based on the position of a 2-position mechanical switch
- Changes the direction of the motor based on the position of another 2-position mechanical switch

## 2 Materials

- x2 PNP Transistors ([BD788G](#))
- x2 NPN Transistors ([BD787G](#))
- x2 2-Position Mechanical Switches
- x1 Breadboard
- x1 5V DC Motor
- Assorted Wires

## 3 Transistors

In the H-Bridge Circuit, there are two sides: a high side and low side. The high side refers to any components that are plugged in between the voltage source and the load (in this case, the motor). The low side refers to components plugged in between the load (motor) and the ground. The transistors used in this lab are Bipolar Junction Transistors (BJT) which have three parts: a collector, an emitter, and a base. The base is (at risk of a gross oversimplification) the part of the transistor that you use to control the connection between the collector and emitter. For matching the pins of the transistors you are given, please refer to the datasheet in Section 2. Because of chemistry and construction and stuff (beyond the scope of this document), different types of transistors can switch different types of signals.

PNP Transistors:

- Can only switch high signals (on the high side of the circuit)
- Are closed when the signal going into the gate is low

NPN Transistors:

- Can only switch low signals
- Are closed when the gate signal is high

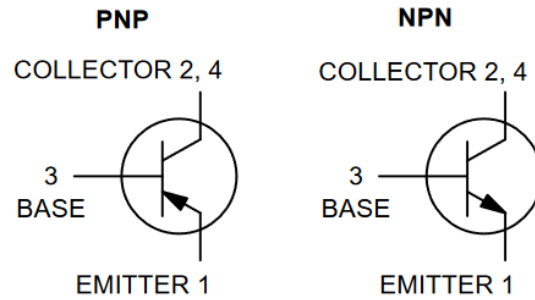


Figure 1: The two types of transistors and their circuit schematic symbols. Note the direction of the arrow, which is the only way to tell the difference of the two.

## 4 H-Bridge Function

An H-Bridge contains 4 transistors, which act like electronically controlled switches, opening or closing different sides of the bridge. This allows the motor to rotate forwards or backwards, depending on which pair of transistors is enabled. To turn off the motor, open all of the transistors (driving the high-side inputs high and low-side inputs low). The H-bridge in Figure 2 also includes diodes in the opposite direction of the current flow. These are called flyback diodes, and provide an alternate path for current in the event of back-EMF. For this lab, we will not be using these diodes for simplicity.

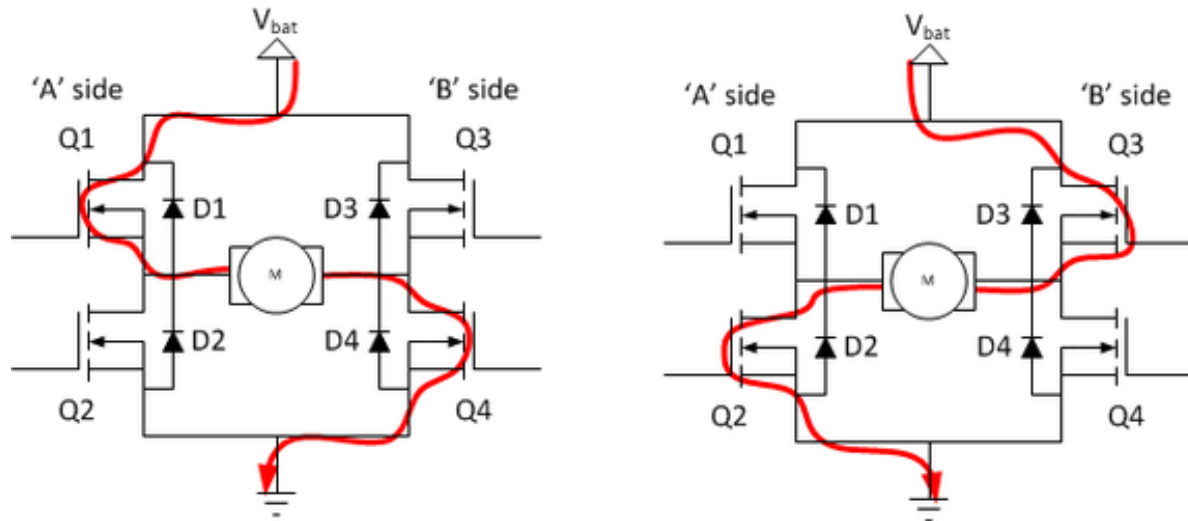


Figure 2: A general H-Bridge diagram showing two possible current paths. Note that this diagram uses MOSFET instead of BJT, so ignore the transistor and diode symbols for now.

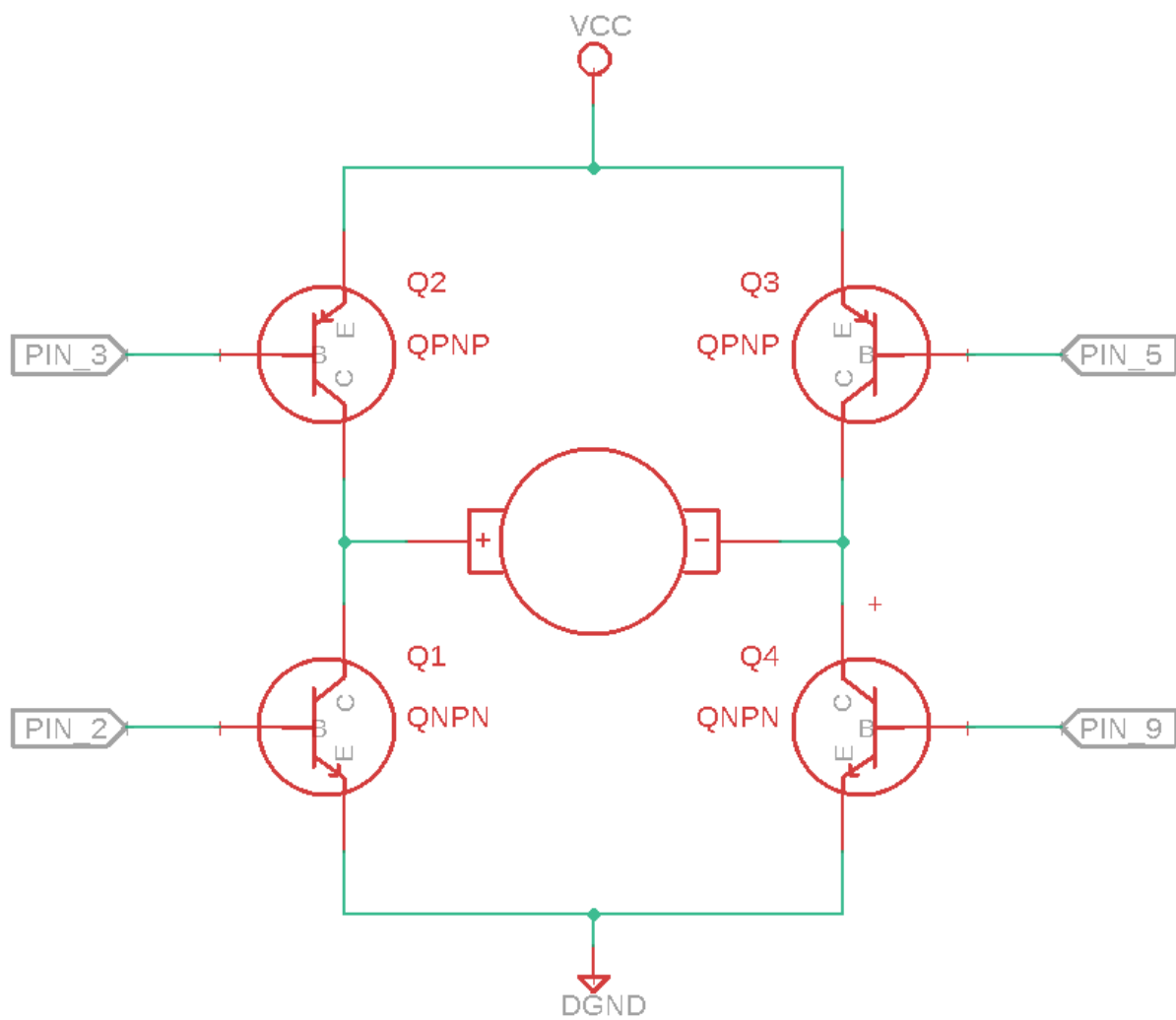


Figure 3: An example hookup guide to assist you in assigning pins. Note the usage of PNP transistors on the high side and NPN transistors on the low side.

## 5 Programming

### 5.1 Forwards and Backwards

When programming your H-Bridge Circuit, you first want to set up your pins. Initialize an input pin for each of the mechanical switches, and an output pin for each transistor. The general outline of your code will be as follows:

1. Define all of your pin numbers.
2. Create three methods: one for going backwards, one for going forwards, and one for not moving.
3. Within your main loop, use `if()` statements to read the values of your inputs, and select which method to run.

**Warning:** Turning on the incorrect pair of transistors (i.e. accidentally closing Q2 and Q1 from Figure 3) will cause a short between power and ground. This low-resistance path to ground causes high current flow and can cause your transistors to heat up or get damaged. Make sure to double-check your pins before powering up your circuit. In addition, do a quick hot check by touching your finger briefly to each transistor. During normal operation the transistors will get warm, but if its hot enough to cause pain or discomfort there is a high chance that a problem exists in the wiring or code.

### 5.2 Speed Control

Pulse Width Modulation (PWM) is a technique used to control the effective voltage being fed to a motor (reference Figure 4). This is accomplished by sending pulses of varying widths to the motor. For the second part of the lab, you will re-purpose your enable switch into a speed selection switch to go between two speeds of the motor (30% and 70%).

In order to control the speed of the motor, you will be pulsing between states on the H-bridge. This is as simple as calling your forward and stationary methods with a given timing between.

To get the motor to turn smoothly, the switching frequency must be high (and the switching period must be low). A good period for the motors used in this lab (as per empirical testing done by the instructors) is 50 ms. Feel free to experiment with values of this period and find what works or what does not.

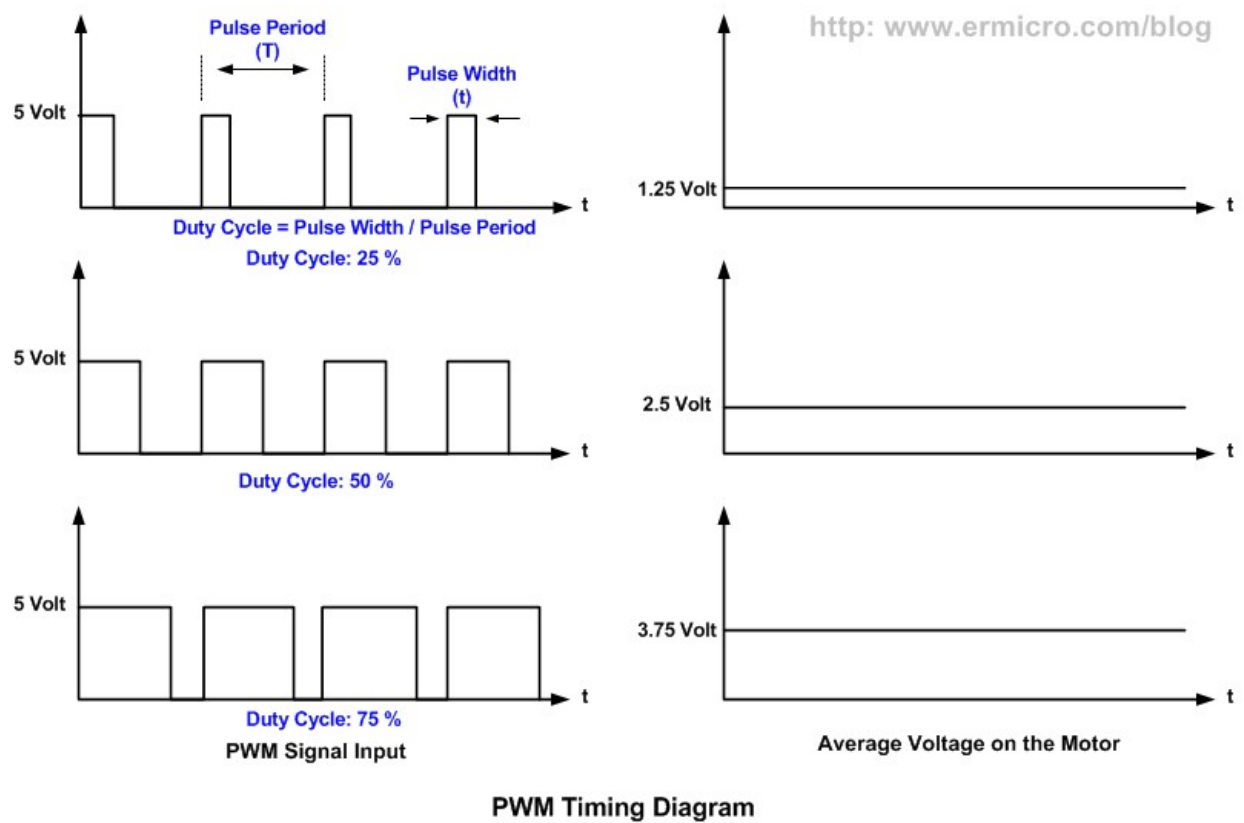


Figure 4: An explanation of the characteristics of a PWM signal. From [ermicroblog](http://www.ermicro.com/blog).