COSC 4P78 – Lab 3

Before we begin:
Before we get started, we'll first sit through a very brief demonstration highlighting some of the key differences between different switching mechanisms. In particular, we'll compare a typical logic-friendly MOSFET against a simple electromechanical relay, and discuss the capabilities and limitations of each.

We should also probably briefly discuss what an h-bridge is.

Preconditions:
Of course, you need to have completed the previous two weeks to be able to continue with this week. i.e. you need a working microcontroller, including the serial cable.

And now, onto the lab!
For this lab, you'll simply be assembling your robot platform, and making it ready for use. This will include soldering any necessary connections within the base itself, adding a motor driver, and getting the robot to move. Since this will be our first time adding motion, it would be best to investigate the fun of *differential drive*.

Assembling the robot:
There isn't really very much to explain here. The kits are pretty easy to figure out. However, there are a few tips worth considering:

- Be ***exceptionally*** careful with the motor terminals! Those things are fragile!
  - It's especially easy to tug on a wire, not realizing how much torque you can generate
  - Remember to use the *stranded* wires for this
- Consider adding ceramic capacitors to the terminals of your motors
  - Some of the motors might have a flexible wrap around them covering the terminals. If yours is like this, just skip this step; it isn't immensely important
- It doesn't really matter “which wire is which”, because you'll be able to swap them in the logic
- You should ***strongly*** consider attaching your wheel encoders now. We aren't using them this lab, but you'll need to partially disassemble the robot if you want to add them later

Beyond that, it should all go swimmingly. Certainly ensure that everything feels solid, but don't strip anything by trying to over-tighten.

The motor driver:
You've been provided with a Pololu motor driver. This acts as an **H-Bridge**. Technically, you don't need to worry about using it as one, but it's nice to have the option. Similarly, you should consider using the PWM control pins.
There's info here: https://www.pololu.com/file/0J86/TB6612FNG.pdf
http://www.robotshop.com/ca/en/pololu-dual-dc-motor-driver-1a-4-5v-3-5v-tb6612fng.html

Okay, so what do we need to know?

- If you have extra space, you *can* install it onto your microcontroller board, but I'm working on the assumption that you're using one of the included mini-breadboards
- **BE INCREDIBLY CAREFUL** which pins you're using. Don't get the left/right mixed-up when you flip the chip over!
  - How important is that? I used *Comic Sans*. Ugh. I feel dirty.

- **Common is common**
  - Connect all of the ground pins for the chip, but also ensure that both the robot platform's ground and the microcontroller's ground are all electrically connected
- If you have room, consider adding a *decoupling capacitor* (or *bypass capacitor*) to the board between the robot's power supply and ground
  - A 47uF electrolytic should work well
- If you want to be super-fancy, also add a bypass capacitor between the motor driver's VCC and ground (on the mini-breadboard)
  - 22uF electrolytic is probably plenty (even 11 should work fine)
- Definitely ensure that you understand how the chip works before you start adding voltage
  - Note that, if *all* you want is to pick variations of going forwards (including some forms of turning), you could get away with only two logic inputs

This should be clear from above, but just to be absolutely certain:
THE 9V BATTERY POWERS THE MICROCONTROLLER. IT (AND, BY EXTENSION, THE MICROCONTROLLER) DOES NOT POWER THE MOTORS. EVER. FOR LOTS OF REASONS. I'M SUPER-CEREAL HERE.

If you kill your motor driver:
Don't.

Yeah, but if you do:
I'm pretty sure none of the labs will actually *require* an h-bridge. If you do kill your driver (or, for some reason, simply prefer to not use it), we should have enough spare MOSFETs for you to wire up a basic on/off control. I'd suggest a 10K pull-down on the gate, and use one of the simple logic-level transistors to control the MOSFET. Power the transistor off the microcontroller, and use the ground of the motor as the drain of the MOSFET (with the MOSFET's source connected to the actual ground).

Task for demonstration:
Make the robot move.

- Ideally, make both wheels turn at different rates, using PWM. However, the only requirements for the lab are simply that you assemble the robot and make it move