ECE 1001 1002 Introduction to Robotics
Lab #9: DriveBot

Objectives

Drive a Bot around the bullpen; learn about motor control, and your robots

Introduction

You will need an Arduino robot which is able to drive around on the table by itself. Lab 9 builds from Lab 8, but you have programming to do now. You could start with the code from lab 8, but a bit more advanced version will be better, download it from Canvas.

Review the H-Bridge basics—you can skip this, read lab 8, or refer to as needed
The motor driver or motor controller is an "H-Bridge:"

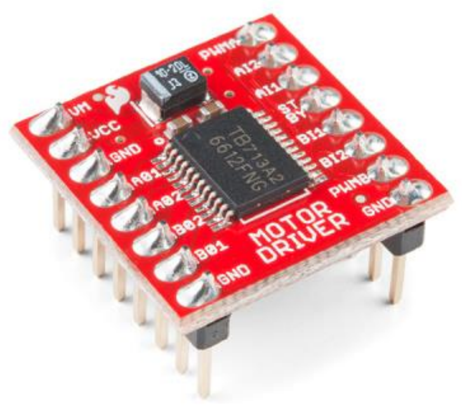


Figure 1 SparkFun online guide

Wire these pins every time you use the H-Bridge:
Pin VM on the motor controller → VIN on the Arduino (connects it directly to the battery power)
Pin VCC on the motor controller → 5V (the Arduino controls this 5V, the batter voltage is not controlled)
Pin GND on the motor controller → GND (just one of the GND pins needs to be connected to ground)
Pin AO1 on the motor controller → motor A red or black lead (you decide left or right, red or black)
Pin AO2 on the motor controller → motor A other wire
Pin BO1 on the motor controller → motor B red or black lead (you decide left or right, red or black)
Pin BO2 on the motor controller → motor B other wire

The H-Bridge can electronically switch the red/black wires leading to a motor to cause the motor to turn either forwards or backwards. The module in your kit contains two H-Bridges, one for motor A, and one for Motor B.

The H-Bridge must be turned on by setting the STBY pin to HIGH. It could be hardwired-on by connecting the STBY pin to 5V to keep it enabled at all times, but a better approach is to turn it on when it's needed, and turn it off when it's not needed.

You must configure the H-Bridge in hardware in a way that matches to the software, you do this by:

- Connecting I/O pins on the Arduino to the AI1, AI2, BI1, BI2 and STBY pins on the H-Bridge
- Declare to the Arduino which pins you connect to the H-Bridge
- Set each of those pins on the Arduino to act as an OUTPUT

Tell the Arduino how fast each motor should turn. Speed is controlled by "Pulse-Width-Modulation," which some of the digital pins are capable of (they have a tilde or "~" next to them on the board). These pins should be set to an OUTPUT, and connected between the Arduino and the motor controller. To turn them on and make them move, you need to send the PWM pins a speed between 0 and 255, where "0" means stopped, "255" is maximum. Speeds less than about 100 may not produce enough torque to turn the wheel, and speeds above 200 may be recklessly fast.

Then in the main loop () part of your program, you need to tell each wheel which direction to spin, and how fast to spin. The exact direction you need to tell your motors to spin can vary depending on how you wired it, which side of the bot each motor is on (is motor A left or right?) and which way the motor points. Fortunately, it's easy to switch the direction the motor turns.

The two "A" input pins on the H-Bridge must always be set to a complementary value:

- Tell the Arduino to set those pins either to a "HIGH" (5 Volts) or "LOW" (0 Volts) level.
- If AI1 is "LOW" then AI2 must be "HIGH" and vice-versa.
- Same for the BI1 and BI2 pins: if BI1 is "LOW" then BI2 must be "HIGH" and vice-versa.

If you want to change the direction of a motor, just switch the "HIGH" and "LOW." You control each wheel independently.

Remember the Arduino can do 10 million things per second, so if you turn the motors on and then turn them off, you'll never see anything because it happened so fast. If you want it to drive for a ways then stop, you need to add a delay command.

Be aware the speed will probably vary with battery charge; you may want to always have a fresh set on hand, but if you change them, you may need to re-adjust your speeds. Speed will also vary with the surface the bot is moving on, table, carpet, hard floor all are different. demonstrations are always on the table.

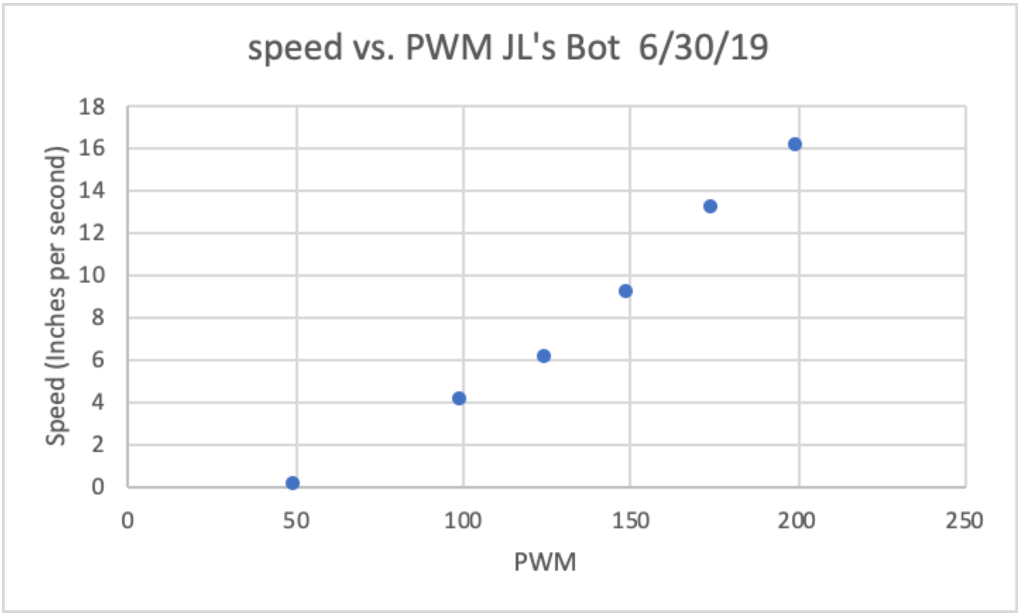
Be careful, changes to your hardware can make a difference—if the motors come off, they need to be attached in just the same way, with the same location. A little twist or shift makes a difference. If the wheel rubs sometimes, adjust it. Don't let wires hang off or snag. Start the bot in the same location each time, with the same orientation.

Requirements

You must write code, modifying the starter file as needed. There is a turn-in location in canvas to upload all of the code you write.

1) Make the bot go only forward, measure the speed of your robot. Each motor uses a Pulse Width Modulation (PWM) value between 0 and +255, with max speed at +255 and no movement at 0. Running your robot near max speed will probably cause troubles. To start, find what PWM value is needed to move the bot at a speed of about 5 inches per second.

Timing how long it takes your bot to go a measured distance will an accurate way to find the speed. Measured distance on the table, time how long it takes to go that distance with a stopwatch for a given PWM setting. Try at least 50, 100, 125, 150, 175, 200. Build a graph showing speed vs. PWM setting, this will be useful in other labs. It could look like this (hand-drawn is OK, if it's neat); your plot is needed for sign-off



2) Make your bot go across the table in a straight line. You need to fine tune the speed of one of the wheels individually to cause the bot to go straight. If it turns left, make the right motor a little faster or left motor a little slower.

Demonstrate your bot driving from the starting gate into the finish gate on the other side of the table. The bot must stop within the finish gate—if it hits the wall or doesn't enter the gate, you'll have to adjust your program.

3) Driving straight will only get you across the table, now make it drive backwards to the starting area, again in a straight line.

Demonstrate a bot driving across the table to the gates, stopping for 0.5 seconds, and then reversing to the starting point.

4) Even more useful is if you can turn. There are several ways to turn, a simple but useful turn is a "pivot-turn" where only one-wheel moves,

Make a 90 degree one wheel turning pivot-turn to the right. Drive forward roughly 10 inches, then turn right and continue for about 10 more inches. Once you master that, add a left turn.

Demonstrate your bot doing these in order:

- Drive straight about 36 inches
- Turn right about 90 degrees
- Drive straight about 36 inches
- Turn right about 90 degrees
- Drive straight about 36 inches
- Turn left about -90 degrees
- Drive straight about 36 inches
- Stop (and do not repeat)

The bot can move any speed you want, although not too fast is better. Pausing a short time before and after turning may improve the accuracy of the turn.

5) Repeat the right-left turns of task 4, but this time demonstrate with a two-wheel turn—one wheel turns forward, the other wheel turns backwards.

6) Make your bot go from one corner of the table to the opposite in a smooth arc, demonstrate this. It must start and stop within a box.

No Report—turn the signoff sheet by the due date and submit the code from requirements 2--6 to the Canvas Lab3 code submission portal