ME102 Lab 4: Stepper motor lab

1| Prelab

Use the internet to become a little more familiar with stepper motors.

We will be using a NEMA-17 stepper motor from Adafruit. Here is the datasheet.

1) Is this a bipolar or unipolar motor?
2) What is the max current per phase? What is the max Voltage?
3) What is the coil resistance? Pretend you didn't have the wiring diagram: how would you figure out which colors are paired?

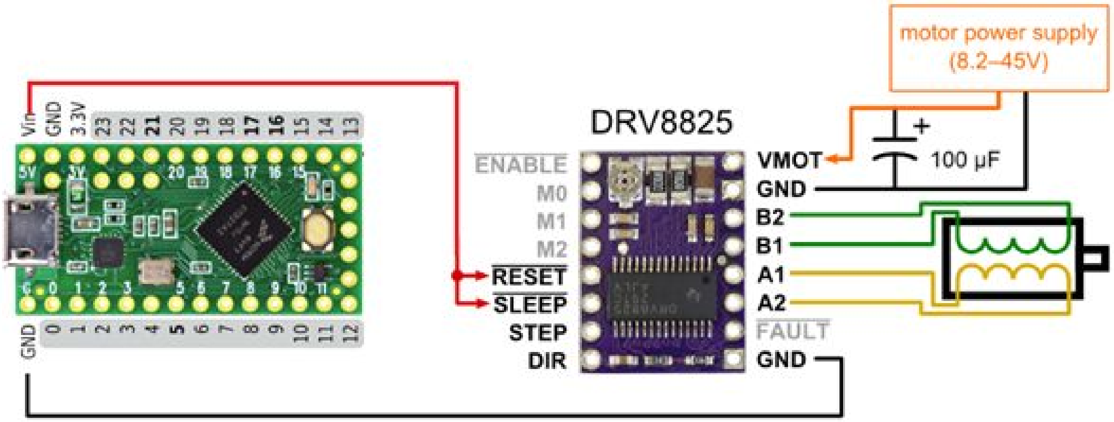
We will be using a stepper motor driver that is a rip-off of Pololu's DRV8825 driver. Scan Pololu's documentation. One significant difference is that the test point via cannot be used to set the current limit.

1) What is the supply voltage range?
2) Look over the pinout diagram.
3) Read about how to set the current limit.
4) Read about microstepping.

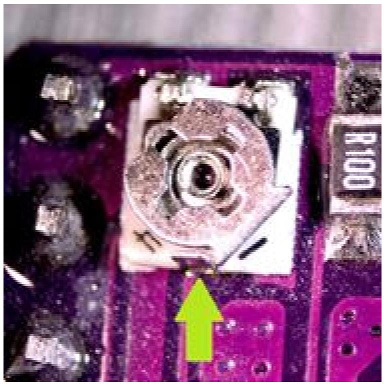
Procedure

2| Setting the current limit

For this lab we will use the DRV8825 stepper driver. First we will learn about setting the current limit. Assemble this circuit.

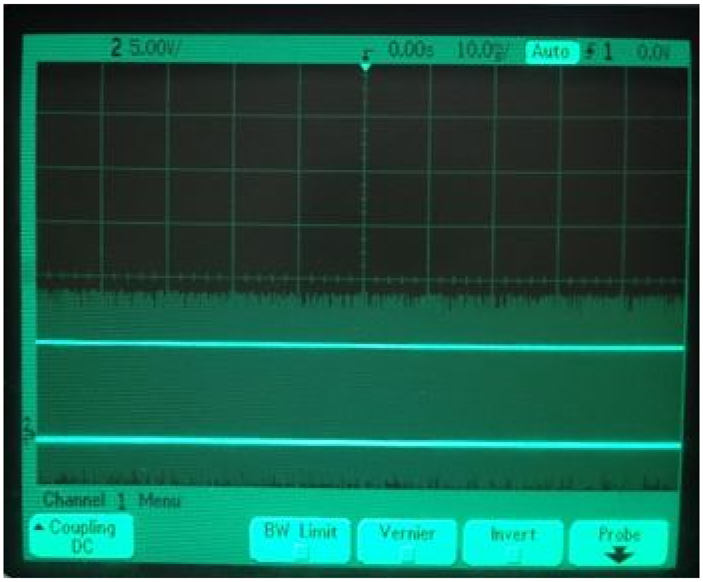


1) There is a potentiometer on the board that controls the current limit. **Please do not use too much force when turning it; it is extremely delicate.** Last semester people broke 24 of them. Observe that there is a tiny sheet metal tab that functions as an end-of-travel stop. When it’s bent over there is no barrier between the minimum and maximum current settings.



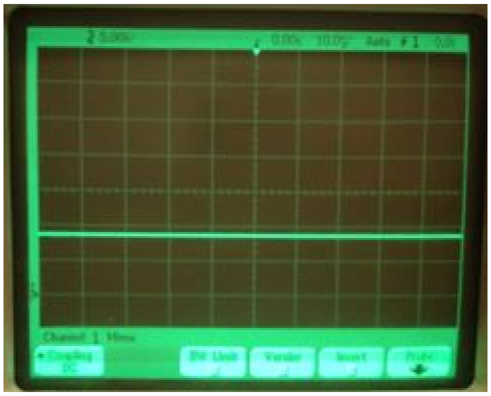
Turn the potentiometer to the fully clockwise position.

2) Power the Vmot pin from the 20V output of the HP power supply and set it to 9V. Connect the Sleep and Reset pins to 5V from the Teensy, or 5V from the other output of the HP PS.
3) At this point the motor will be enabled but not moving. Use the DMM to monitor the voltage on one of the stepper coils (should be close to 0V). Slowly turn the potentiometer counterclockwise until the voltage on the coil is between 2-5V. Connect a scope probe to the side of the coil that has positive voltage. At this setting the driver is limiting the current delivered to the coil with PWM. You should see something like this:



Note that the lower bar is ground and the higher bar is the supply voltage (almost).

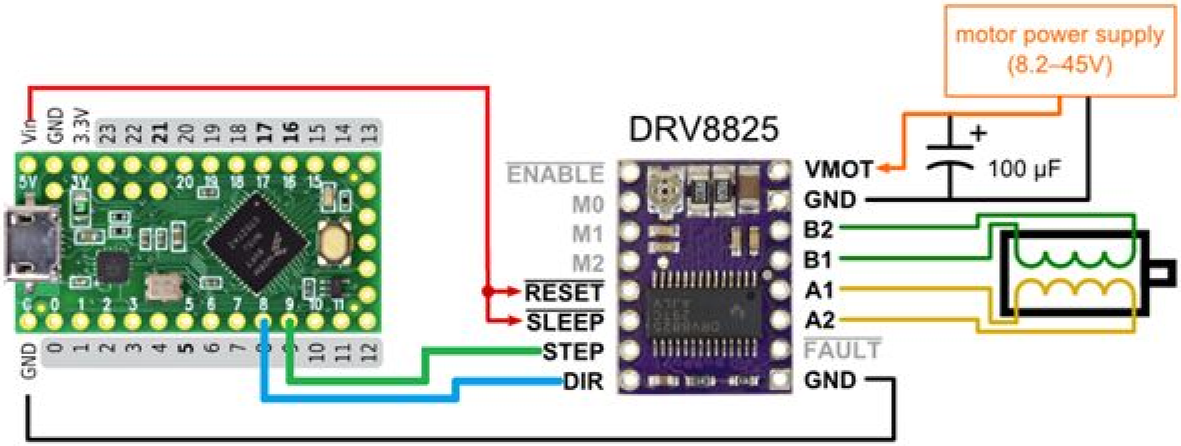
4) Turn the potentiometer counterclockwise until the switching seen on the signal goes away and you are left with a solid DC voltage. The readings on the DMM and the scope should be the same. In this state the driver is not current limiting and there is no PWM on the signal.



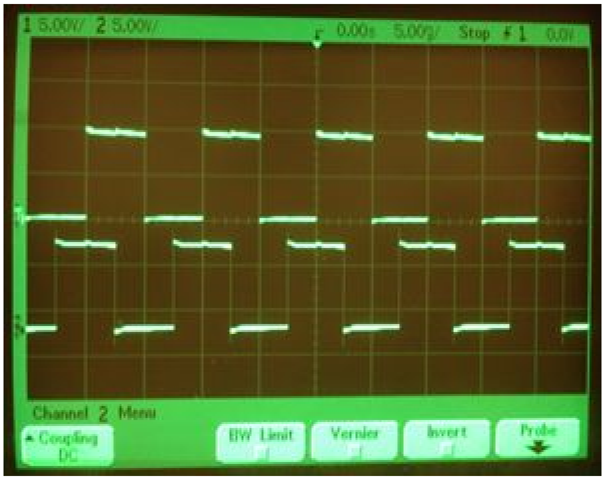
5) Increase the voltage on the power supply to around 12V while watching the coil voltage on the scope and meter. Turn the pot to set the coil voltage to around 10V, or until the power supply starts current limiting (yellow LED). At higher settings the motor will become quite toasty. You can reduce the value if the motor becomes uncomfortable to handle.
6) Disconnect the multimeter from the coil and now measure the voltage on the potentiometer dial itself (connect one lead to ground and touch the other lead to the top of the pot). The voltage should be approximately half the current displayed on the power supply. As described on the Pololu datasheet, this is one way to set the current limit on the driver. Do the voltage and current readings make sense given the coil resistance of this motor?

3| Stepper Commutation

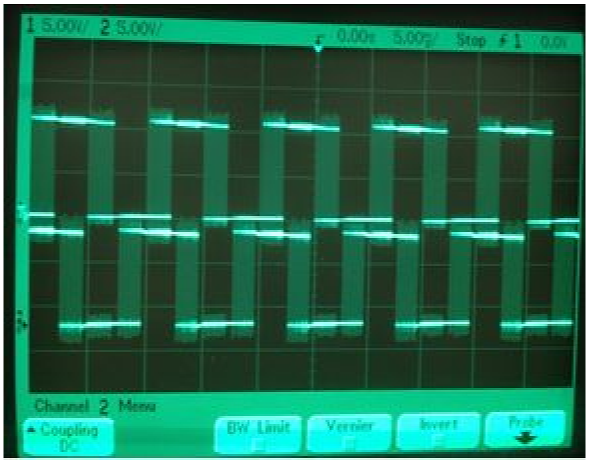
7) Now we are going to move the stepper with code. Connect pins 8 and 9 on the Teensy to PWM and STEP on the driver.



8) We will use stepper library written by Laurentiu Badea (laurb9). It has custom options for the DRV8825. I've loaded it onto the lab computers but if you are using your laptop you'll have to download it first. Go to Sketch > Library > Manage Libraries and search for DRV8825.
9) Load Examples > StepperDriver > BasicStepperDriver. It's all the way at the bottom under Examples from Custom Libraries. Compile and run. You should have a moving motor.
10) Lower the motor power supply to 9V, or until the driver is not current limiting. Connect two probes to a lead on each coil and observe the signals on the scope (ie, A1 & B1). You should see something like this:



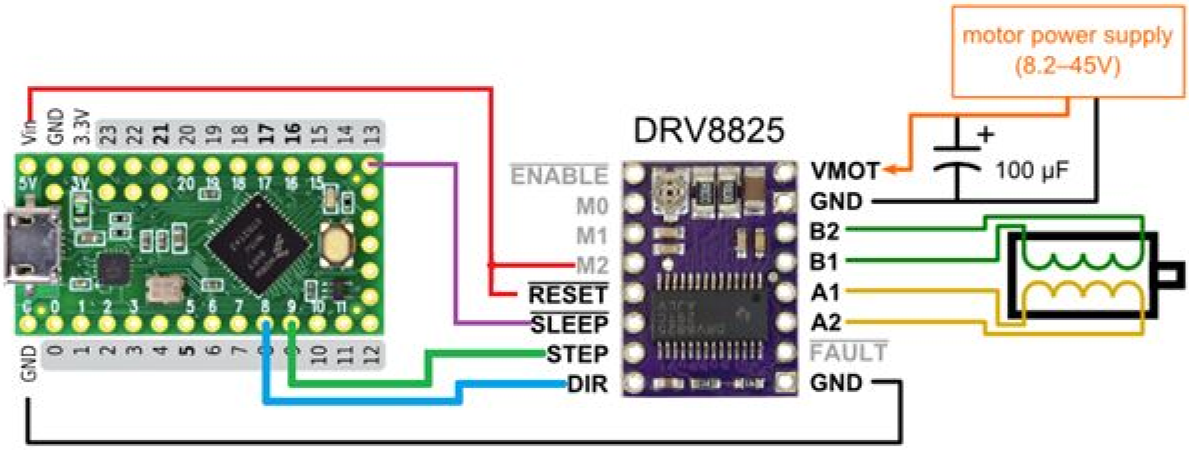
11) Raise the motor power supply back to 12V. You should see PWM on the signals.



12) In the #define are at the top, change RPM to 200, compile, and upload. Yup, faster. Increase the RPM in increments of 100 until the motor stops working. Bonus question: Why did the motor stop moving correctly?
13) Try lightly resisting the motion of the motor with your hands while reducing the current limit. See how the torque is proportional to the current.

4| Code with microstepping, acceleration, and sleep

14) Disconnect SLEEP from 5V and instead connect it to pin 13 of the Teensy. Connect M2 to 5V.



15) I modified the code to use microstepping, acceleration, and sleep functions. Compile and upload this code.

```
/* Simple demo, should work with any driver board
 * Connect STEP, DIR as indicated
 * Copyright (C)2015-2017 Laurentiu Badea
 * This file may be redistributed under the terms of the MIT license.
 * A copy of this license has been included with this distribution in the file LICENSE. */

#include <Arduino.h>
#include "BasicStepperDriver.h"

#define MOTOR_STEPS 200       // Motor steps per revolution
#define RPM 220
#define MOTOR_ACCEL 2000
#define MOTOR_DECEL 1000
// Since microstepping is set externally, make sure this matches the selected mode
// If it doesn't, the motor will move at a different RPM than chosen
// 1=full step, 2=half step etc.
#define MICROSTEPS 16

// All the wires needed for full functionality
#define DIR 8
#define STEP 9
//Uncomment line to use enable/disable functionality
#define SLEEP 13

 #include "DRV8825.h"
 #define MODE0 10
 #define MODE1 11
 #define MODE2 12
 DRV8825 stepper(MOTOR_STEPS, DIR, STEP, SLEEP, MODE0, MODE1, MODE2);

void setup() {
	stepper.begin(RPM, MICROSTEPS);
	// if using enable/disable on ENABLE pin (active LOW) instead of SLEEP uncomment next line
	// stepper.setEnableActiveState(LOW);

	stepper.setSpeedProfile(stepper.LINEAR_SPEED, MOTOR_ACCEL, MOTOR_DECEL);
}

void loop() {
	stepper.enable();    // energize coils - the motor will hold position

	stepper.rotate(720);
	delay(2000);

	stepper.move(-2*MOTOR_STEPS*MICROSTEPS);

	stepper.disable();    // pause and allow the motor to be moved by hand
	delay(2000);
}
```

16) Notice stepper.disable(); in the loop. When sleep is not enabled (LED on Teensy is off), try turning the shaft of the motor with your hand.
17) In the #define area, modify RPM, MOTOR_ACCEL, and MOTOR_DECEL and see what happens. Reduce the travel by changing values in stepper.rotate() and stepper.move() and see what happens.