CS466 Lab 7 – Motor Servo.
Final lab, Due Midnight, May 4

Notes:

- This lab requires the use of an H-Bridge motor Driver the H-Bridge was provided in your box-o-parts

Objective:

1. Learn what a positional servo is and how a how a software PID loop can be used to implement one.

Requirements:

2. Connect your Motor, TivaBoard, and H-Bridge electronics to match the provided wiring diagram.
3. Tune the PID servo parameters using the 'Manual Tuning Method' provided in https://en.wikipedia.org/wiki/PID_controller#Manual_tuning
4. Demonstrate your system seeking the positions given in the heartbeat task of the master.c file.
5. Answer the questions in the lab steps.

Prior To Lab:

- Take time to read and understand the requirements and what the system will do.
- Review the PID Controller in https://en.wikipedia.org/wiki/PID_controller
- Review the H-Bridge Operation in https://en.wikipedia.org/wiki/H-bridge
- Review the Servomechanism Wiki https://en.wikipedia.org/wiki/Servomechanism
- Review the L298N Board Info Document (L298N_Board_Info.pdf)
- Because we are short on time I have provided all the framework code for the lab. I do expect you to take the time to evaluate the code. There will probably be questions on the final regarding the motor and PWM code. Look first at the header files to see what each module exposes. Spend enough time to see what the motor.c and pwm.c do.
- I spent about 10 minutes tuning after I hooked the circuit up to resolve to this behavior for my example video: https://photos.app.goo.gl/avbZ1jfnkxD6yyGs8

How I would progress,

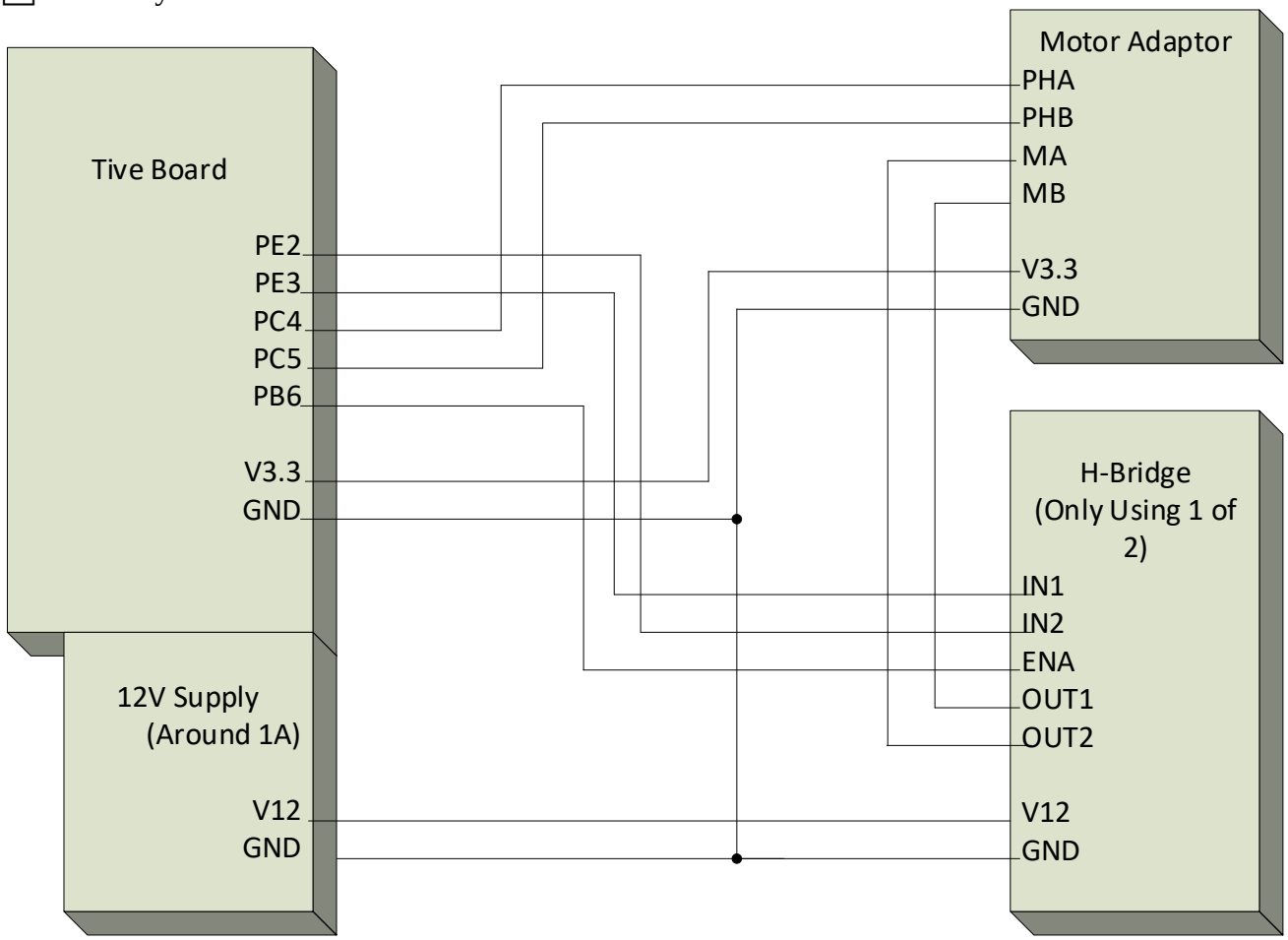
1. ☐ To abbreviate the lab, I have provided a working framework of code that only needs the PWM. I will update the class repo with a building target.

2. ☐ Take time to examine the software. master.c is the main program but the implementation is broken into a couple files. The motor.c file has the servo controller main loop

3. ☐ This lab is easier if the motor is secure and there is additional weight on the armature. I had a little stick attached to my motor that was balanced. Try to find something round and mark on it so that you can easily see the motor position.

4. ☐ This lab uses the Tiva PWM output to control torque to the motor and uses IN1 and IN2 to control the h-bridge direction. Combinations of IN1 and IN2 can be used on the motor to provide electric braking as well as a freewheel operation. In our lab the motor is always under tight control.

5. ☐ Connect your hardware as shown:



6. ☐ The PID terms are initially set to 0 and the motor will not have any drive capability as a result. I suggest starting with a Kp around 0.1 and increasing slowly from there…

7. ☐ Per the Wikipedia page adjust Ki and Kd in an effort to get the motor to operate close to mine.

8. ☐ Play with the motor some, Try to move it off its programmed setpoint. Does it try to correct?

9. ☐ Prepare a brief lab report and in it be sure to answer the following questions;

a) What is the purpose of using vTaskDelayUntil() as opposed to a simple vTaskDelay() in _pidPositionServo()
b) Write a sentence describing each of Kp, Ki, Kd, dt
c) What is deadband and why is it important
d) Take a look at what the define USE_MATH_ENCODER_ALGORITHYM enables. Contrast how the two quadrature interrupt routines work.
e) How does a single 12V supply turn the motor two directions.
f) The implementation is that of a positional servo. Another simple implementation is that of a velocity servo. Describe what changes would you make to convert this program to a velocity servo so that the 'pos' positions in heartbeat() would become target RPM's

10. ☐ Post your demonstration video somewhere and include the link in your report.

Be sure to writeup your lab according to the lab format handout.