EE31 Junior Design Spring 2019 Project Design Phase 02B
Pre-Studio

Assignment Date and Due Date: check Assignment/Due Date Listing Pre-studio assignments are to be recorded in your Bound Lab Notebook and reviewed by the instructor at the start of class before starting your design work.

If you are not sure what to do on this assignment, ask your instructor or TAs.
Pre-studio questions: Motion control

Objective: Design and construct a breadboard electrical circuit platform that is able to demonstration motion control with two motors using software from the Arduino microcontroller for forward, reverse, turn left, turn right, turn 180°, turn an arbitrary number of degrees, and vary the speed using the L293D H-Bridge integrated circuit.

Think: 1. What problem are you trying to solve?

Design a drive system that can drive two motors both forwards in backwards

2. What does success look like for this design problem? How do you know when you are done? What are the specifications on the deliverable? The tolerances?

Fixture that can drive in a straight line at 1.5 inches/second for 5 inches without deviating more that 10 degrees from the straight line. We know when we are done once we have

3. Define pulse width modulation (PWM).

Modifying the duty cycle of a periodic digital signal.

4. Who is the customer? What is the customer requirement that impacts motion control? In the last phase there was only one motor, now there are two motors. How does your design specification change from last time? What customer requirement does this reference?

The customer is the course staff. The customer requirement impacts motion control is the requirement to drive in a straight line for 5 inches at 1.5 inches per second without devaiting more 10 degrees from the straight. The design specificatoin change from last time is that we need twice as much current, two control signals and callibration between the two motors. They all reference straight line requirement.

5. What are the pin assignments for the L293D H-Bridge integrated circuit?

- PIN 1 = ENABLE1
- PIN 2 = IN1
- PIN 3 = OUT1
- PIN 4 = GND
- PIN 5 = GND
- PIN 6 = OUT2

- PIN 7 = IN2
- PIN 8 = v_s
- PIN 9 = ENABLE2
- PIN 10 = IN3
- PIN 11 = OUT3
- PIN 12 = GND
- PIN 13 = GND
- PIN 14 = OUT4
- PIN 15 = IN4
- PIN 16 = V_{ss}

6. What are the voltage and current limits of this H-Bridge?

Parameter	Value	Unit
Supply Voltage	36	V
Logic Supply Voltage	36	V
hline Input Voltage	7	V
Enable Peak Output Current	1.2	V

1. You know how a CMOS inverter is constructed from MOSFETs: How is the buffer shown on the L293D schematic constructed? How is this similar to your discrete FET controller last week?

2. Draw out the internal circuit schematic of the H-Bridge with an applied voltage of 5V such that two motors are controlled. What voltages are applied to each of the pins to make two motors move forward? Reverse? Rotate? How is this related to the truth table for the H-Bridge? How would you modify these modes to include speed control?

3. What is the estimated current draw? Does this change your battery considerations or power subsystem?

4. Sketch how the Arduino connects to the H-Bridge to turn the two motors on and off.

5. How fast can the Arduino be switched from one motor to the other?

6. In lab you may observe that the Arduino occasionally resets when the motor turns on. Speculate on why this happens. How is this prevented?

7. The bot is moving in a straight line. One of the sensors detects a problem requiring the bot to turn left. Write out the pseudo-code that make the bot change direction.

8. What states are required to provide all desired motion control for the bot? Which pins need to be defined in the setup() function? Which pins are global?

9. The loop() function runs forever. How do you make the Arduino do forward, reverse, turn left, turn right as required by the current 'state' of the bot (i.e., input from the bot's sensors)?

10. How does the bot turn 60°? (Hint: what is the specified shaft speed and wheel diameter?) What is the error tolerance on turning a desired angle? Plan:
11. Design, sketch and analyze a schematic of the circuit that you will construct during class to meet the posted specification.
12. What parts will you need to build a motor control circuit with an H-Bridge?
13. Design in pseudo-code the Arduino program that you will implement during class to meet the posted specification.
14. Write out the pseudo-code for moving in a straight line?
15. Write out the pseudo-code for turning left?

Do:

1. Are you following your task list? Who is responsible for what? When will it be completed? What and how did you get a commitment? Do you believe the task will be complete by the promised delivery date? Do you trust your teammate to follow through on their commitment? Why or why not? If not, what are you going to do about it?
2. What is the fixture required to get the design constructed? What makes a fixture so important for this phase? What is the critical functionality of the design fixture?

Test: 1. What is your test plan?

2. What equipment is required?
3. How do you test the speed and direction of the bot?
4. For what reasons do you need a special test bed?

Reflect: 1. What was the most unexpected failure? How did it happen?

2. What did you learn?
3. Why myth in your head was dispelled?

Task Questions: 1. How well did you complete the team design tasks you took responsibility for from the last design phase? What did you do well? What unexpected problems did you have? How did you overcome them? What did you learn?

2. What part of the team design tasks for this phase are you planning to take responsibility for this assignment?
3. What part of the team building and testing tasks for this phase are you planning to take responsibility for this assignment?
4. What will you need to learn to complete the team tasks for this phase for which you have taken responsibility?

Word to the Wise: The best design approach is to assembly four mosFET ICs as a discrete H-bridge to understand how an H-bridge works and how it controls the motors – forward, backward, and turns – and at various speeds. When you have completed a discrete H-bridge AND have software on the Arduino working for motion control, THEN AND ONLY THEN duplicate the circuit with the H-bridge IC and keeping the discrete bridge intact to make your debugging efficient. Make sure the two circuits work the same. If they do not, understand the changes you need to make to make your design work. Typical issues are bypass caps and diodes needed versus non-needed, voltage levels on the PWM, ground loop, power source not wired correctly or at the incorrect voltage, voltage not being stable enough, and a host of other items. This is the reason you want to start with a discrete Hbridge as you will understand what is happening, determine how to fix the problems, and when it happens with the Hbridge IC when built into your bot, you understand the flaky behavior, recognize it, and know how to fix it.

You can do anything you want, as it is your design and your design approach. However, there is a reason this section is in a large bold font. What might that reason be?