ME 210 Winter 2025: Laboratory Assignment 2

Pre-Lab due at 11:59 pm on February 3, 2026.
Report Due at 11:59 pm on February 12, 2026.

Purpose:

This lab is intended to acquaint you with:

- Controlling a brushed DC motor
- Finding information in datasheets
- Phenomena encountered with DC motors
- Limitations of techniques implemented purely in software
- Connectorizing cables using Molex KK series connectors
- Pulse width modulation (PWM)
- **(Optional)** Controlling a stepper motor

Minimum Parts Required:

1 each 1N5245 15V Zener diode, 10kΩ or 20kΩ potentiometer

Each project team is also provided an additional bag of parts with:

Valuables: brushed DC motor, L298 module (these items to be returned on completion of lab)

Consumables stranded wire, Molex crimps & connectors

Combutibles None

Logistics:

This lab is to be done in groups of two (comprised of project team members). Each project team (two lab groups) will share a bag of valuables and consumables.

How your project team (of four) ***splits up cable-making duties*** is entirely up to you, however we do ask that the rest of the lab be done entirely independently within each group of 2; time-sharing of motors and control boards may be necessary within project teams (perfect project preparation!).

Because of the number of functional check-offs required as part of the lab, we are allowing ***peer check-offs*** by 2 peers not on your project team (keep a list of these checkoffs). Please include a copy of this sign-off list in your lab report.

Turn in *one copy* of your completed lab report on Gradescope, and make sure that *both* of your names appear on it!

To ease the grading task, please quote the question you are answering before your answer. This way, the grader can read the question you think you are answering, followed by your answer.

Part 0: Pre-Lab

Complete the following exercises *after* you have read through the lab assignment and readings, and *before* coming into the lab to complete it.

Assignment:

- ☐ **0.1)** Decide which Arduino ports and pins you will use to control the motors (all parts of the lab).
- ☐ **0.2)** Decide what, if any, initialization is required for each pin.

In the Report:

Include your answers to the exercises in the Pre-Lab.

Part 1: Pulse Width Modulation

Reading:

Textbook Chapter 8.4, *Pulse Width Modulation*

Assignment:

You are to design the software necessary for an Arduino to drive the supplied DC motor. Your motor drive should use PWM (pulse width modulation) with a minimum resolution of 1 part in 1024 (i.e. 10 bits). This control is to be implemented completely in software written by you (without the use of Arduino library functions that generate PWM signals, such as `analogWrite()`). The speed at which you run the motor should be determined by the setting of an external potentiometer, similar to the IR LED frequency in Lab 1. Rotation of the motor should stop when any key is pressed (sent to the Arduino in the serial monitor). When you have completed this task, you will be prepared to continue with the exercises in Part 2. You have already designed something very similar to this already in Lab 1 – feel free to reference it!

☐ **1.1)** Check that the Lab kit contains the necessary components for Parts 1 and 2 of the lab. The lab kit should contain the following items (only one set is needed per project team):

- 1× L298 H-bridge module Brushed DC motor
- 1× Brushed DC Motor

The provided brushed DC motor may or may not already have wires soldered on. If it is not already assembled, solder wire leads onto the motor, making sure to heat shrink the connection. Using a Molex connector is not required, but provides a convenient method for making connections to the motor. If you are interested in making Molex connectors, see a TA if you have more questions.

☐ **1.2)** Connect the two outer leads from the potentiometer to the Arduino's 5V & GND terminals, and the wiper of the potentiometer to your selected A/D pin on the Arduino. Verify that the potentiometer behaves appropriately.

☐ **1.3)** Write and upload the code to the spec outlined in the Assignment section above. Use your `TimerInterrupt` skills from Lab 1 to keep a consistent signal period! Verify that your output's duty cycle changes as expected.

☐ **1.4)** Using an oscilloscope, determine the frequency of the duty cycle waveform that you are generating. What is the frequency? take a picture of your generated waveform.

☐ **1.5)** Can you find a way to increase the operating frequency? What are the limitations of these methods?

☐ **1.6)** What happens to the upper frequency limit if we want more duty cycle resolution? Describe the trade-off between resolution and frequency of operation.

☐ **1.7)** Can your solution achieve a full range of operation? (That is: what happens if you try to get 0% or 100% duty cycle?) Why or why not? If it can't, don't try to fix it just explain why it won't work.

☐ **1.8)** For a given duty cycle, what are the sources of error in the approach you have chosen? (i.e., if you could set the potentiometer perfectly to ask for 50% duty cycle do you get it? ...theoretically, that is, as you probably will not be able to measure the error.)

☐ **1.9)** With respect to motor control, why would you want a high PWM frequency? What are the tradeoffs of this?

☐ **1.10)** Set the frequency of your output to approximately 490 Hz for the remainder of the lab.

In the Report:

Include a listing of the source code used to generate the PWM signal, your answers to the questions in parts 1.4 to 1.9, and the oscilloscope waveform capture requested for part 1.4.

Part 2: Interfacing to a DC Motor and H-Bridge

Reading:

Canvas	ME210 L298 Dual 1A H-Bridge Module documentation
Online	Arduino `analogWrite()` and PWM documentation: docs.arduino.cc/learn/microcontrollers/analogoutput

Assignment:

You are to use the L298 H-Bridge to drive the motor. You will also add a little complexity: you will include a provision to change the direction of rotation of the motor by pressing any key (and sending this to the Arduino using the serial monitor). For this section of the lab, you will be using the Arduino standard library PWM functions (`analogWrite()`). Make sure to read through the documentation to ensure that you are connected to PWM-enabled pins, and that your arguments to `analogWrite()` are in the correct range.

- ☐ **2.1)** You will need alligator clips or a power connector (large Molex) to supply power to the H-bridge on the L298 Module. Connect power and ground to the `Vin` connector, with the + pin on the left and - on the right. Be sure to supply at least 8V to the power connector, as the L298 will not operate with motor supply voltages lower than 7.5V when its logic supply is at 5V.
- ☐ **2.2)** You will need to provide the enable (5V) and half-bridge inputs to the chip from the Arduino. You can use female-female jumper leads or assemble a Molex KK connector for this.
- ☐ **2.3)** Demonstrate your working motor and code to two peers to get checked off for Part 2. You must show that your motor gradually changes speed as you adjust the potentiometer. Also, you must show that your motor reverses direction if a key is pressed.

In the Report:

Include the schematic of the circuit you used. Be sure to indicate on the schematic which pins on the Arduino you used. Include a listing of the source code used to drive the DC motor using the L298. Ensure you are checked off for part 2.3.

Hints on working through this assignment

This can be a challenging laboratory assignment. For this lab in particular, it is often very helpful to begin by quickly reading through all the materials provided: the lab handout, the hardware documentation and the software library documentation. Knowledge of what information is provided, and where to find it, will be very helpful as you work your way through the parts of the lab. It is not essential that you understand everything you're reading on the first pass, but it is critical to understand what's there and where to look when questions arise.

Once you understand what's there, it's also helpful to spend some time becoming familiar with the function of the software modules and the Arduino. Write some simple test code to send out pulses with the Arduino's PWM functions. Once you understand first-hand how the software and hardware work, you'll be in a good position to perform the specific tasks requested in each part of this lab assignment. This also prepares you to design your software (i.e. writing pseudo-code) and hardware.

And this next part is especially important in ME210: **If you have spent more than one hour on any single task (after coming in prepared), then something is wrong! STOP and ask a TA, instructor, or your neighbor to take a look at what you are doing.** Often, a fresh pair of eyes will spot simple problems that you've missed, and this can save you a great deal of time.

General Debugging Tips:

- Make sure you properly call `pinMode(pin, mode)` for each of the I/O pins that you use.
- For the DS3658, the ground of the control pins should be connected to the ground of the 7.5V power supply.
- When you are reading in from the Serial input, `Serial.available()` will only tell you whether or not there is anything in the Serial buffer. **It will NOT clear the Serial buffer for you!** To do so, you can call `Serial.read()` repeatedly until `Serial.available()` returns false. Failure to do so can result in always reading `Serial.available() == TRUE`, or reading multiple inputs right after each other (depending on your line ending settings in your Serial monitor), which may cancel each other out.
- Ensure that the pins chosen for Encoder A and Encoder B are interrupt-enabled.