ME218B

Laboratory Assignment Number 6 for ME 218b

Pre-Lab Due by 23:59 on January 20, 2026
Due by 17:00 on January 22, 2026

Goals:

In this lab you will explore the behavior of a DC motor being driven by Pulse Width Modulation (PWM). You will also gain experience writing an interrupt response routine to measure the speed of the running DC motor and report the speed of the motor.

Submitting your Report

All lab reports will be submitted electronically. Your report should be a single PDF file, based on the Word document template file provided on Canvas in the folder associated with this lab assignment. Any requested schematics, or graphs should be embedded into the document. This template file includes sections for everything that you need to submit for the report. Please be sure that your answers are in the correct sections of the template document. When calculations are requested, the calculations should be included as scans of your neatly handwritten calculations (use a physical scanner, or smartphone scanner app. Please do not submit a jpeg file, they can be difficult to read.). When including required schematics in your reports, be sure to use the procedure outlined in the Lab 0 assignment section Supplemental KiCad Info to copy from KiCad and paste into Word. Limit the size of the area being transferred so that the resulting images fit on the page in the report. When the report document is complete, print it to a PDF file, in color.

When copying your code into your report, please use the Highlight program to prepare an RTF file using the “bright” color theme. You can then open the RTF file in Word and copy and paste the code into your report and it will maintain the color highlighting. Look at the code that you paste into your report. If the indenting is messed up anywhere, it means that you had tab characters in your source files. If the indenting is messed up, stop, go back and fix it before running Highlight again and submitting your report.

You submit your report by uploading the report PDF file to Gradescope and identifying in your document where the answers to the various sections are located. In order to get your reports graded and the feedback back to you as soon as possible, it is essential (read that as required) that you use the template document as the starting point for your report and correctly associate the sections of the assignment in Gradescope to the correct pages in your report document. I have instructed the TAs not to search for your answers if they are not in the correct portion of the template document or if you have not properly associated the pages to the sections in the Gradescope assignment. If you submit your report after the due date, please inform us.

When labs require you to write code, the plain text .c files will be uploaded to Gradescope separately from the report so that they can be processed by MOSS (Measure Of Software Similarity) to insure that each of you has written the code for yourselves.

Part 0: Pre-Lab

Background:

The pre-lab should be completed after you have read through the entire lab assignment, but before you go into the lab to begin your work there. Completing the pre-lab, which can be done without any special tools or resources, will allow you to be most efficient with your time in the lab. We judge the pre-lab submission based on a reasonable effort in pre-lab document submitted.

Assignment:

- ☐ 0.1) Design the circuitry necessary to complete parts 1 & 2 of the lab assignment. Your circuitry should bi-directionally drive the DC motor provided at the lab bench (see Lab 6 Appendix A) using an SN754410 as the drive stage with a motor voltage of 5V, provide an analog voltage (0-3.3V) from the potentiometer in Part 1.1 to an analog input on the PIC32, provide for a digital input capable of input capture to read the encoder and include the drive circuitry for the bar-graph LED in Part 2. The 5V to drive the motor should come from the bench-top power supply, NOT from the breadboard power supply! Include this schematic in your report. The internal kick-back diodes in the SN754410 have proven to be problematic in prior years, so be sure to include external diodes in your design. Please confirm that the H-bridge specified can drive the motor (quote specifications/measurements to back that up)! You should make a measurement on the motor to confirm this.
- ☐ 0.2) Determine which pins on the PIC32 you need to hook up for the complete schematic from part 0.1. Prepare a table of the pin numbers and port connections that correspond to the pins identified in Part 0.1.
- ☐ 0.3) Design the event driven software that you will use for Part 1. Include pseudo code for program. Do not worry about getting the specifics of the PWM module ready for the pre-lab. We will cover the PWM subsystem in lecture on Thursday.
- ☐ 0.4) Design the software that you will use for Part 2. Include pseudo code for the program.

In the report:

Include the schematic diagram of the circuit you used for 0.1, the table of pin assignments from 0.2, pseudo-code from 0.3 & 0.4 and design calculations. Be sure to indicate on the schematic which bits of which ports of the PIC32 you used, and please include pin numbers.

Part 1: Driving a DC Motor Using PWM

Reading:

Sections 12, 14 & 15 in PIC32 Data Sheet (Canvas), Section 14 in PIC32 FRM Timers (skip 14.5) (Canvas), Section 16.3.3 in PIC32 FRM Output Compare(Canvas), COK: Chapter 8, section 8.3 & Chapter 23

Components Required:

PIC32MX170F256B, Bench top Power Board, 1 ea. L293NE (or SN754410), 4 ea. 1N4935(or 1N4936) diodes, 1 ea. 10K potentiometer and assorted other components of your choosing. See a TA for any parts you need that are not in your parts kit.

Assignment:

You are to design the necessary circuitry and software to connect the PIC32 to drive the supplied DC motor using the L293NE (or SN754410) as the power stage and PWM using the hardware PWM capability of the PIC32. Shoot for a drive frequency of about 200Hz. Do not use the PWM library. The speed at which you run the motor should be determined by the setting of an external potentiometer which your software reads using the A/D converter in the PIC32. Your program should periodically (10 Hz, you may use the framework timer library for this) read the voltage set by the potentiometer and set the motor speed directly proportional to the voltage.

- ☐ 1.1) Implement your DC motor driver and potentiometer design from Part 0.1 and an Event Driven software design to read the A/D converter connected to the potentiometer (powered from the 3.3V supply on the breadboard power supply as in Lab 5) and adjust the speed from Part 0.3. The code to read the A/D should be implemented in a service within the framework. Hook it up to the motor and get the motor to run with the speed controlled by the potentiometer.
- ☐ 1.2) Demonstrate your working software to a coach/TA/Ed and get signed off on the check-out sheet.

In the report:

Include a Highlighted version of the source code to the program. Be sure to be signed-off by a coach or TA This quarter, we will be using Gradescope for your code submissions as well. Look for a Gradescope assignment entitled Lab 6 Code and submit the raw .C files (not Highlighted). Note, you will complete 1 code submission (including the files for all parts) when you submit, so save this step to the last. Please make sure that your files are named such that we can tell which

files go with which parts of the lab.

Time Spent:

Preparing outside of the lab_____

In the lab working this part_____

Part 2: Measuring Motor Speed

Reading:

COK Chapter 8, section 8.3, Section 11 (all subsections) in PIC32 Data Sheet.

Components Required:

PIC32MX170F256B, bench top Power Distribution Board, 1 ea. L293NE (or SN754410), 4 ea. 1N4935/6 diodes, 1 ea. 10K potentiometer, 1 ea. LTA-1000HR 10 element bar graph array & 74ACT244 (from ME218a Lab Kit) and assorted other components of your choosing.

Assignment:

In this part of the lab, you will use a combination of interrupt and non-interrupt based routines to measure and display the speed of a DC motor.

- ❑ 2.1) Using your code from Part 1 as a base, add an Input Capture interrupt response routine to measure the period of the encoder signal. Hook up a single channel of the encoder output to the appropriate input line of the PIC32. The 5V supply for the encoder and display should come from the breadboard power supply. Be sure to map the input capture to a 5V tolerant pin on the PIC32!

- ❑ 2.2) Add code to the mainline (non-interrupt-driven) code that will write a scaled version of the current value of the encoder count to 8 of the 10 LEDs (implement an 8-segment bar graph with the length of the bar proportional to the value of the interval between encoder edges). You will need to use a 74ACT244 (or equivalent, e.g. 'VHCT, 'HCT, etc.) to drive the LEDs. Your bar graph should show 1 bar lit at max speed and progressively more bars as the speed drops. Scale it so that it reaches 8 bars just above the slowest possible speed. You should only update the display if there has been a new encoder edge since the last update.
  Compile and download this program. Run the program. When this is working, get a coach/TA/Ed to sign off on it

- ❑ 2.3) Add a few instructions to your code from Part 2.2 to raise an I/O line when you begin the calculation of the scaled value and lower the line after you write the scaled value to the LEDs. How long did the calculation take? Hint: use a 'scope or logic analyzer to measure the high time of the pulse.

- ❑ 2.4) Add code to a service that will write the speed of the output wheel, as an integer RPM, to the TeraTerm window. Update (re-write) this speed at a rate of 10 times per second (you may use the framework timer library for this).
  Compile and download this program. Run the program. When this is working, get a coach/TA/Ed to sign off on it.

- ❑ 2.5) Add a few instructions to your code from Part 2.4 to raise an I/O line when you begin the calculation of the RPM and lower the line after you complete the calculation. How long did the calculation take?

- ❑ 2.6) Add a few instructions to your code from Part 2.4 to raise an I/O line immediately before writing the RPM and lower it immediately after writing the RPM. How long did it take to write the RPM to the screen?

In the report:

In addition to the answers from 2.3, 2.5 & 2.6, include a Highlighted version of the source code to the programs that you wrote for Part 2. Be sure to be signed-off by a coach or TA This quarter, we will be using Gradescope for your code submissions as well. Look for a Gradescope assignment entitled Lab 6 Code and submit the raw .C files (not Highlighted). Note, you will complete 1 code submission (including the files for all parts) when you submit, so save the submission of raw C code to the last step. Please make sure that your files are named such that we can tell which files go with which parts of the lab.

Time Spent:

Preparing outside of the lab_____

In the lab working this part_____

Part 3: Mapping a DC Motor

Reading:

Section 20 (all subsections) in PIC32 Data Sheet, Section 11 (all subsections) in PIC32 Data Sheet.

Components Required:

PIC32MX170F256B, bench-top Power Board, 1 ea. L293NE (or SN754410), 4 ea. 1N4935(6) diodes, 1 ea. 10K potentiometer and assorted other components of your choosing.

Assignment:

In this part of the lab, you will use the lab setup to explore the relationship between duty cycle and motor speed.

- ❑ 3.1) Using your code from Part 2 as a base, adjust the PWM frequency to approximately 250Hz. Step through drive duty cycles from 0 to 100% in 10% increments and record the speed of the motor at each duty cycle. Be sure to wait until the motor speed stabilizes before recording the speed. For regions where there is a large change in speed between two adjacent 10% duty cycle points, go back and add 2 more duty cycle points between them. Create a plot of Speed (Y) vs Duty Cycle (X)
- ❑ 3.2) Repeat Part 3.1 with the PWM Frequency set to approximately 500Hz. (add to the same graph)
- ❑ 3.3) Repeat Part 3.1 with the PWM Frequency set to approximately 1000Hz. (add to the same graph)
- ❑ 3.4) Repeat Part 3.1 with the PWM Frequency set to approximately 2000Hz. (add to the same graph)
- ❑ 3.5) Repeat Part 3.1 with the PWM Frequency set to approximately 10000Hz. (add to the same graph)

In the report:

Include a separate table for of duty cycle and corresponding motor speed for each of the sub-parts. Include a single plot of RPM (Y) as a function of Duty Cycle (X) using the data from each of the sub-parts (label the lines with the corresponding frequency).

Time Spent:

Preparing outside of the lab_____

In the lab working this part_____

Time Summary

Please add the time that you spent preparing the report to the times that you logged in each section and enter the data from the table into the Time Summary for Lab 6 assignment on Canvas.

While this information is being gathered solely to produce statistical information to help improve the lab assignments, it is a required part of the lab assignment.

Appendix A

Motor Stand Connections

Stepper Ph B	1	20	NC	
Stepper Ph B	2	19	NC	
Stepper Ph A	3	18	NC	
Stepper Ph A	4	17	NC	Ribbon Cable
DC	5	16	NC	
DC	6	15	NC	
Encoder A	7	14	NC	
Encoder B	8	13	NC	
Encoder +5	9	12	NC	
Encoder Gnd	10	11	NC	

The encoder on the large motors gives 512 pulses per revolution on each channel, 90 degrees out of phase from one another. There is a 5.9:1 gearbox between the motor and the output wheel.

For the small motors, the encoder gives 3 pulses per revolution on each channel, 90 degrees out of phase from one another. The small motor has a 298:1 gearbox whose output drives the wheel.