

ME 430 Exam 2, Fall 2019 - 2020

Name _____

Class periods _____

You may use only:

- Any paper notes (including course handouts) you brought to the exam, or electronic notes residing on your local (C:) hard drive.
- The course website, including any code from the website. This is the only approved use of the internet for this exam.
- Code written by you or by your lab partner(s).
- Pencil/pen and a calculator (optional).
- The green demo board and its accessories.
- Moodle for code submission.

Anything not specifically allowed is prohibited. In particular, you may not use code written by someone outside your lab group unless it came from the course website.

If your code for one of the problems works properly, you should get it checked off. There are points associated with the check off itself. You have 3 total tries for each checkoff.

The only code in the programs should be the code which is necessary to accomplish the task—points will be deducted if there is extra stuff that we need to sort through. At the end of the test submit your .c files to Moodle. You do not need to submit LCD Module.c.

Problem	Points	Check off
1a	/ 14	/ 3
1b	/ 13	/ 3
1c	/ 25	/ 3
1d	/ 16	/ 3
2	/ 17	/ 3
	/ 85	/ 15
Total	/100	

For all checkoffs, for an individual part or for an entire problem, please use the MPLAB X “Make and Program Device” button to **PROGRAM** your board then remove the PICkit3. The program will continue to run on its own. It will speed checkoffs for everyone if your program is simply ready to go and it lets you move on to the next part while waiting for a checkoff.

Problem 1 – LCD and ADC

Start this problem from “**template with interrupts.c**”, but rename it to “**Lastname_Problem1.c**”. This problem will use the LCD display so perform all necessary steps to use “LCD Module.h” and “LCD Module.c”.

The goal of this problem is to display the ADC reading of RA2 (the horizontal joystick) to the LCD and light some LEDs to follow the joystick movement left and right. To help you solve this problem you can work it using the parts below. You can get a part checked off when you complete it if you like or just check off the entire problem with the final part.

Part A. Initial LCD display

In this part display a message on the LCD screen like this...

J	O	Y	S	T	I	C	K								
R	A	2			=		1	2	3	4					

When you get this part working, you can check it off or continue to the next part.

If you choose to checkoff now, **program** your green board, remove the PICkit3, and call your instructor / TA over to get this part checked off.

Part B. Display RA2 readings

Now program the PIC to read the output of RA2 (the horizontal joystick) and print the value to the LCD. The program can use the while(1) loop to continuously update the LCD. Note, your variable reading RA2 should be an **int**.

J	O	Y	S	T	I	C	K								
R	A	2			=		1	0	2	3					

The numbers should vary between 0 and 1023. Be sure to follow the exact format shown above. For this problem it is acceptable to display leading zeros or not display leading zeros.

When you get this part working you can check it off or continue working on the next part.

Problem 1 continued.**Part C. Timer 0 interrupts**

Using Timer 0, a 4 MHz clock, and a 1:256 prescaler, setup a timer interrupt that will fire every **0.75 seconds**. You are required to use Timer 0 as a **low** priority interrupt. Add the line of code to set the **priority** explicitly even if that is the default. In addition to writing the code, complete the blank below.

Clock = 4 **MHz**

Timer 0 Prescaler = **1:256**

Parameter passed into WriteTimer0 = _____

When the Timer 0 interrupt fires, update the LCD display using the ADC reading.

Make sure you REMOVE (or comment out) you code from the while(1) loop that was displaying the ADC readings in the prior part. Have **only the Timer 0 interrupt update the LCD**.

When you get this part working you can check it off or continue working on the next part.

To avoid guessing at the value **many** times you only get 3 attempts to get the value above correct.

Instructor area	<input type="checkbox"/> Attempt 1	<input type="checkbox"/> Attempt 2	<input type="checkbox"/> Attempt 3
------------------------	------------------------------------	------------------------------------	------------------------------------

Problem 1 continued.**Part D. LEDs**

In addition to printing the analog output to the LCD, light LEDs using the following format:

RA2 ADC Reading	Lights on
0 - 100	RC0 and RC2
101 - 450	RC1 and RC3
451 - 550	RC1, RC3, RC7, RC5
551 - 900	RC7 and RC5
901- 1023	RC6 and RC4

When complete the LEDs will appear to follow the joystick movement left and right. For example, when the joystick is all the way left, the left side LEDs RC0 and RC2 are on.

Update the LEDs inside the **Timer 0** interrupt so they only update once every timer overflow. You should be able to observe a short lag between the joystick movement and the LEDs.

Since this is the final part of this problem, call your instructor / TA over to get this part checked off. When you demo this part make sure you **PROGRAM** the PIC (don't use the debugger).

Problem 2 – PWM

Start this problem from “**template.c**”, but rename it to “**Lastname_Problem2.c**”.

Using a 2 MHz clock set PWM channel 2 to flash the LED on RC1 at 600 Hz. Use the **analog input on channel 1** to determine the duty cycle for **PWM channel 2**. Show that you can dim the LED on RC1 using the RA1 potentiometer. Record you chosen settings below:

Clock = 2 MHz

Timer 2 Prescaler = _____

TimerTicks = _____

When you demo this part (or any part) make sure you **program** the PIC.

Once you complete the exam OR when time expires submit your .c files (only) to Moodle. You do not need to submit LCD Module.c.

To avoid guessing at the values **many** times you only get 3 attempts to get the values above correct.

Instructor area	<input type="checkbox"/> Attempt 1	<input type="checkbox"/> Attempt 2	<input type="checkbox"/> Attempt 3
------------------------	------------------------------------	------------------------------------	------------------------------------