

ME 430 Exam 2, Winter 2019-2020, All Sections

Name _____

Section _____

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.

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.

Problem	Points	Check off
1a	/ 8	/2
1b	/ 24	/2
1c	/ 24	/2
2a	/ 12	/2
2b	/ 22	/2
	/ 90	/10
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 PICKit. 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 Interrupts

Start this problem from “**template with interrupts.c**”, but rename it to “**Lastname_Exam2_P1.c**”. This problem will use the LCD display so perform all necessary steps to use “LCD Module.h” and “LCD Module.c”.

Logan and Avery are in a debate. Whenever Logan wants to talk, he presses RB1. Whenever Avery wants to talk, he presses RB2. Whenever one of them begins talking, they are given 4 seconds before their time is up. Create a program that keeps track of who is talking and how much time he has left. To help you solve this problem you can work it one part at a time as shown below. You can get each part checked off when you complete it or check off the entire problem all at once after completing Part C.

Part A. Initial LCD Display

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

N	o	b	o	d	y		t	a	l	k	i	n	g		

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

Part B. Who's Talking?

Set up two pushbutton interrupts: one for Logan to start talking (RB1) and one for Avery to start talking (RB2). **You must use the OpenRB1INT() and OpenRB2INT() functions.** Avery is actually louder than Logan, so make **RB2 a high** priority interrupt and make Logan **RB1 a low** priority interrupt (include the lines of code necessary to set the priorities even if it is a default). **Include a 30ms debounce in your pushbutton interrupts** to avoid multiple interrupt calls for a single button press. When RB1 is pressed, the screen should look like this:

L	o	g	a	n		i	s		t	a	l	k	i	n	g	
T	i	m	e			l	e	f	t	:		4		s	e	c

When RB2 is pressed, the screen should look like this:

A	v	e	r	y		i	s		t	a	l	k	i	n	g	
T	i	m	e			l	e	f	t	:		4		s	e	c

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

Part C. Countdown

Finally, modify your code from part B so that whenever Logan or Avery starts talking, the timer counts down the seconds from 4 to 0. After each second, the LCD screen should update to show the time remaining (4 sec → 3 sec → 2 sec → 1 sec → 0 sec). Once the timer reaches 0 seconds, the screen should return to the initial “Nobody talking” display.

Interruptions:

- If Avery pushes his button (RB2) while Logan’s timer is counting down, Avery starts talking and the timer resets to 4 seconds.
- If Logan pushes his button (RB1) while Avery’s timer is counting down, Logan starts talking and the timer resets to 4 seconds.
- If either person pushes their own button while they are talking (interrupting themselves), the timer resets to 4 seconds.

To implement this requirement, set up a timer interrupt that will fire every **1.0 seconds** using Timer 0, a 1 MHz clock, and a 1:16 prescaler. You are required to use Timer 0 as a **high** 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 = 1 **MHz**

(make sure to set the clock to 1 MHz in your code as well)

Timer 0 Prescaler = **1:16**

Parameter passed into WriteTimer0 = _____

When you get this part working, call your instructor over to check it off.

At the end of this exam makes sure to submit your .c files to Moodle, regardless of whether you finished or not.

Problem 2 – Joystick Controlling Light Show

Start this problem from “**template.c**”, but rename it to “**Lastname_Exam2_P2.c**”.

Create a program on the green board that will alternate between lighting all of the even LEDs (RC0, RC2, RC4, and RC6) and all of the odd LEDs (RC1, RC3, RC5, and RC7) to make them “dance”.

Part A. Fixed dance speed

For this part of the problem you will make the LEDs “dance” at a fixed rate of 2Hz (i.e. each state is on for 0.25 seconds). Using a 4 MHz clock, what delay command(s) would you use for a 0.25 second delay?

Clock = **4 MHz**

Delay command(s) used to generate a 0.25 second delay: _____

Additionally, implement an OFF switch on RA4 so that **while the joystick is pressed down**, the LEDs stop switching and are **all off**. Releasing the joystick should allow the lights to begin switching again.

Part B. Variable dance speed

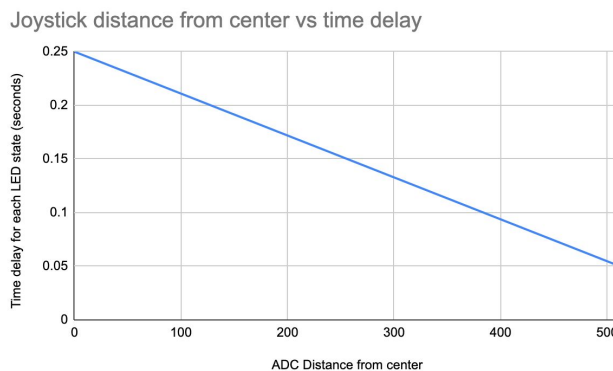
Now make the dance speed variable based on the horizontal joystick (RA2).

- If the joystick is in the center (neutral) position, the LEDs should switch back and forth at a rate of 2Hz (0.25 seconds per light state, 0.5 for a full even-odd cycle).
- If the joystick is moved to the left or right, the rate at which the LEDs switch should **increase linearly** until it reaches a maximum of 10Hz (0.05 seconds per light state, 0.1 seconds for a full even-odd cycle) at either extreme (all the way to the left or all the way to the right).

Fill in the blanks for the number of instruction cycles for a 4 MHz clock:

- ADC of 512 → 0.25 second delay per state Delays: _____ k
- ADC of 0 or 1024 → 0.05 second delay per state Delays: _____ k

Use that knowledge to make a linear interpolation. It is OK if your formula is **not EXACT**; you should be able to linear interpolate close enough with simple numbers.



When you complete this task, program your green board, remove the PICKit, and call your instructor over to check off this problem. **At the end of this exam makes sure to submit your .c files to Moodle, regardless of whether you finished or not.**