ME 430 Exam 2, Winter 2016 - 2017, All Sections

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	/ 20	/2
1b	/ 10	/2
1c	/ 20	/2
2a	/ 20	/2
2b	/ 20	/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 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 1a – Analog readings

Start this problem from "**template.c**", but rename it to "**Exam2_Lastname_P1.c**". This problem will use the LCD display so perform all necessary steps to use "LCD Module.h" and "LCD Module.c".

Use the vertical reading of the joystick potentiometer (which is RA3) to control the PORTC LEDs. Use the following table to determine which LEDs will be on:

Table 1: ADC readings table

Analog reading	LED state
850 - 1023	RC0 and RC7 are on
700 - 849	RC1 and RC6 are on
550 - 699	RC2 and RC5 are on
500 - 549	RC3 and RC4 are on
0 - 499	No LEDs are on

When you get this part woking you can check it off or continue onto Problem 1b. All parts of of Problem 1 build on each other, so you can do a single checkoff for all of Problem 1. If you choose to checkoff now, program your green board, remove the PICkit3, and call your instructor over.

Problem 1b – LCD display

Continue this problem in the same file as Problem 1a. Display the word Exam 2 on the top line of your LCD display. The light functionality of Problem 1a should also still work.

E	x	a	m		2										

When you get this part working you can check it off or continue onto Problem 1c. All parts of Problem 1 build on each other, so you can do a single checkoff for all of Problem 1. If you choose to checkoff now, program your green board, remove the PICkit3, and call your instructor over.

Problem 1c – LCD + Analog readings

Continue this problem in the same Problem 1 file.

Use the horizontal reading of the joystick potentiometer (which is RA2) to control two asterisk symbols ** on the LCD. When the horizontal joystick is in the middle the LCD will display

E	x	a	m		2										
							*	*							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

When the joystick is pushed left (past some threshold) the asterisks print on the left

E	x	a	m		2										
*	*														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

When the joystick is pushed right (past some threshold) the asterisks print on the right

E	x	a	m		2										
														*	*
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

So the two asterisk symbols are either center, left, or right (no other positions). The light functionality of Problem 1a should also still work.

In Problem 1a the ADC values were divided into 5 bins (bins sizes were not all equal). Go glance at that table now. In this problem you need to divide the range for ADC values into **three equally spaced bins.** Write the 6 required values to accomplish this into the table below (again see Problem 1a for an example of an ADC readings table).

Analog reading	Asterisk state
-	Moved over to the left (the lowest ⅓ of the range)
-	In the middle (the middle ⅓ of the range)
-	Moved over to the right (the highest ⅓ of the range)

When you complete this task program your green board, remove the PICkit3, and call your instructor over to check off this problem. Your instructor will check for correct functionality **and** look at your answers above. If either is incorrect you will not get checked off.

Instructor area	☐ Attempt 1	☐ Attempt 2	☐ Attempt 3

Problem 2a – Flashing lights

Start this problem from "**template with interrupts.c**", but rename it to "**Exam2_Lastname_P2.c**".

Make sure the piezo buzzer is NOT connected to your green board.

Make the RC2 light flash at 2.7 Hz and a 30% duty cycle using the PWM peripheral. Before starting the code do your calculations and write your answers in the blanks below:

Clock = _____

Timer 2 prescaler = _____

Parameter passed into OpenPWM1 = _____

Parameter passed into SetDCPWM1 = _____

When you get this part working you can check it off or continue onto Problem 2b. All parts of Problem 2 build on each other, so you can do a single checkoff for all of Problem 2. If you choose to checkoff now, program your green board, remove the PICkit3, and call your instructor over.

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

Instructor area	☐ Attempt 1	☐ Attempt 2	☐ Attempt 3

Problem 2b – Timer Interrupt

This problem starts from your Problem 2a code.

Using Timer 1 create a timer interrupt that will fire every 3 seconds. You are required to use Timer 1 as a **high** priority interrupt (add the line of code to set the priority explicitly even if that is the default) with a **1:1 prescaler**.

Clock = _____(must match your previous answer in 2a)

Timer 1 Prescaler = _____1:1_____

Parameter passed into WriteTimer1 = _____

When the timer interrupts fires change the duty cycle of the PWM. The duty cycle needs to toggle back and forth between 0% or 30%. This will have the effect of making your LED flash for 3 seconds (while the duty cycle is 30%), then be off for three seconds (while the duty cycle is 0%), flash for 3 seconds (30%), be off for three seconds (0%), continuing that pattern forever.

When you demo this part (or any part) make sure you PROGRAM the PIC. Your instructor will check for correct functionality **and** look at your answers above. If either is incorrect you will not get checked off.

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

Instructor area	☐ Attempt 1	☐ Attempt 2	☐ Attempt 3