

ME 430 Exam 2, Fall 2013-2014, 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.
- Email 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, email your three *.c files to your instructor.

Problem	Points	Check off
1	/27	/3
2a	/18	/2
2b	/18	/2
3	/27	/3
	/90	/10
Total	/100	

Problem 1 – Lights

Start this problem from “**template.c**”, but rename it to “**Lastname_Problem1.c**”.

Create a program on your green board that will...

- Turn on all four LEDs on the left (**RC0-RC3**) when both **RB0 & RB1** are pressed
 - If either **RB0** or **RB1** are unpressed no lights on the left will be on
- Turn on all four LEDs on the right (**RC4-RC7**) when both **RB2 & RB3** are pressed
 - If either **RB2** or **RB3** are unpressed no lights on the right will be on

When you complete this task call your instructor over to check off this problem.

Problem 2 – Joystick Follower

Start this problem from “**template with interrupts.c**”, but rename it to “**Lastname_Problem2.c**”. Also add “**LCD Module.h**” and “**LCD Module.c**” to this project.

In this problem, you will make symbols on the LCD follow the joystick.

Part A. Display this text on the LCD screen (blank boxes show a space on the LCD).

						*	*	*	*						
						*	*	*	*						

Each line has 6 spaces, 4 asterisk symbols, 6 spaces.

When you get this part working you can check it off or continue working. The next part builds on this code.

Part B. ADC reading.

Add code to read the horizontal joystick position. The horizontal joystick is ADC channel 2. If the ADC reading is less than 300 (i.e. the joystick is to the left) then change the LCD to display:

*	*	*	*												
*	*	*	*												

If the ADC reading is over 700 (i.e. the joystick is to the right) then change the LCD to display:

												*	*	*	*
												*	*	*	*

If the joystick is not less than 300 or greater than 700 (i.e. the joystick is in the middle) the LCD should display the original pattern:

						*	*	*	*						
						*	*	*	*						

Your code should loop forever updating the LCD whenever the joystick moves.

When you complete this task call your instructor over to check off your work.

Problem 3 – Button Counter

Start this problem from “**template with interrupts.c**”, but rename it to “**Lastname_Problem3.c**”.

In this problem you will display a binary number using the 8 LEDs on the green board. When the program starts all lights will be out, which represents the number 0.

Use interrupts to make the lights count **up** when **RB1** is pressed and count **down** when **RB0** is pressed. Set up RB0 and RB1 to call the `high_isr` when a button is pressed.

- If **RB1** is pressed make the binary number that shows on the lights **one higher**.
- If **RB0** is pressed make the binary number that shows on the lights **one lower**.
 - However, don't let the number drop below zero!

Example

Program starts	0000 0000 (no lights on)
Press RB1	0000 0001 (RC0 is on)
Press RB1 again	0000 0010 (RC1 is on)
Press RB1 again	0000 0011 (RC0 and RC1 are on)
Press RB0	0000 0010 (RC1 is on)
Press RB0	0000 0001 (RC0 is on)
Press RB0	0000 0000 (no lights on)
Press RB0	0000 0000 (still no lights on)
etc.	

You must use interrupts to accomplish this task. You will receive no credit if you do not use interrupts.

You do not need to worry about debouncing your buttons at all. If the pushbutton bounces many times per press that's fine. Of course, if you'd like to add debounce code, that's fine too (no extra credit. 😊)

When you complete this task call your instructor over to check off this problem.