

## ME 430 Exam 2, Fall 2014-2015, 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
<b>1a</b>	<b>/ 10</b>	<b>/2</b>
<b>1b</b>	<b>/ 15</b>	<b>/2</b>
<b>1c</b>	<b>/ 15</b>	<b>/2</b>
<b>2a</b>	<b>/ 10</b>	<b>/2</b>
<b>2b</b>	<b>/ 20</b>	<b>/2</b>
<b>2c</b>	<b>/ 20</b>	<b>/2</b>
	<b>/90</b>	<b>/10</b>
<b>Total</b>	<b>/100</b>	

## Problem 1 – Buzzer Fun

Start this problem from “**template.c**”, but rename it to “**Lastname\_Problem1.c**”.

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

- Not make any noise if there is no button pressed.
- Play a note at 300 Hz when RB0 is pressed.
- Play a note at 500 Hz when RB1 pressed (regardless of whether RB0 is pressed).
- Play a note at 700 Hz when RB2 is pressed (regardless of whether RB0 or RB1 are pressed).

For this problem you are required to use an oscillator speed of 2 MHz. Before starting the code do your calculations and write down the values you are using here:

Clock = 2 MHZ

Timer 2 Prescaler = \_\_\_\_\_

OpenPWM1(\_\_\_\_\_); // For 700 Hz when RB2 is pressed

OpenPWM1(\_\_\_\_\_); // For 500 Hz when RB1 is pressed (w/o RB2)

OpenPWM1(\_\_\_\_\_); // For 300 Hz when RB0 is pressed (w/o RB2,RB1)

You can work this problem all together or in parts. 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.

### Part A. Make some noise

Make your board play 500 Hz (just to make sure the buzzer works)

### Part B. Noise ONLY on a button press

Next make your green board only play 500 Hz when RB1 is pressed.

### Part C. All three buttons

Next make RB0 and RB2 work in addition to RB1. Be very careful about the prioritization of the buttons. We WILL test pressing multiple buttons at the same time.

When you complete this task call your instructor over to check off this problem. You can wait to submit this problem to Moodle until you finish all problems.

## Problem 2 – ABCs

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 write the ABCs on line 1 of the LCD. When RB0 is pressed you should add a letter to the ABCs. So it starts empty.


Then a press to RB0 adds the letter A (RB0 must be debounced).

A															

*Note on debounce code:* For full credit you are required to use the version of debouncing that puts the Delay directly into the ISR. To be honest, that is the easier debouncing approach anyway and it works just fine for this problem. So no main event flags ~= | = **stuff**. ☺

Another press to RB0 adds B. Each press to RB0 adds a letter...

A	B														

Up to a maximum of ABCDEF (more presses to RB0 will have no effect).

A	B	C	D	E	F										

However if a user presses RB1 a single letter will be removed. For example if the LCD was displaying ABCDEF, one RB1 press (RB1 must also be debounced) would display:

A	B	C	D	E											

Presses to RA1 should remove letters all the way down to no letters remaining. Presses to RB1 when no letters are present will have no effect. You can work this problem all together or in parts.

### Part A. LCD Display.

In this part we will simply display ABCDEF on the LCD screen like this...

A	B	C	D	E	F										

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

**Part B. Counting up**

Now make your display start empty and have it add letters on RB0 presses up to ABCDEF. Note, this MUST USE INTERRUPTS and DEBOUNCING. When you get this part working you can check it off or continue working.

**Part C. Counting back down**

Finally add RB1 to make the count down feature. Again you must use interrupts and debouncing. Call your instructor over to check off your work.

**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.**