

## ME 430 Exam 2, Winter 2012-2013, All Sections

Name \_\_\_\_\_

Section \_\_\_\_\_

### Ground rules for the exam:

7:00pm - 8:15pm

Group #1, Last name earlier in the alphabet

8:30pm - 9:45pm

Group #2, Last name later in the alphabet

Group #1 may not leave the room until 8:15.

Group #1 should bring the PICDEM 2 kits.

Group #2 should return everything to the locker.

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 PICDEM2 board and its accessories.
- ANGEL 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 (or before), place the code in the ANGEL drop box.

Problem	Points	Check off
<b>1a</b>	<b>/10</b>	<b>/1</b>
<b>1b</b>	<b>/15</b>	<b>/2</b>
<b>1c</b>	<b>/10</b>	<b>/2</b>
<b>1d</b>	<b>/10</b>	<b>/0</b>
<b>2a</b>	<b>/10</b>	<b>/1</b>
<b>2b</b>	<b>/15</b>	<b>/1</b>
<b>2c</b>	<b>/10</b>	<b>/1</b>
<b>2d</b>	<b>/10</b>	<b>/2</b>
	<b>/90</b>	<b>/10</b>
<b>Total</b>		<b>/100</b>

**Problem 1 – Timer interrupt lights**

Start this problem from “template with interrupts.c”, but rename it to “Lastname\_Problem1.c”. This problem has been broken into parts. Each part builds on the prior parts. You can check off each part individually *OR* get the entire problem checked off at the end.

This program will turn the PORTB LEDs on and off. The LEDs should turn off if RA4 is 0 and a Timer0 interrupt will turn the LEDs on. ***Read all parts of the problem before you start planning and writing your code.***

Make sure that the jumper (J6) next to the LEDs on the green board is **connected**.

**Part A.** Turn on all the PORTB LEDs when the program starts. Then add an *if* statement within the while loop to turn off all the LEDs if RA4 is 0.

**Part B.** Using the 4 MHz external clock, with a 64 prescaler, create a Timer 0 interrupt that goes off every **three seconds**. When the timer interrupt goes off turn **on** all the PORTB lights.

This should create a battle between RA4. Press RA4 to turn the lights off. Then some amount of time later (0 to 3 seconds) the lights will come back on.

Show your code to your instructor to be sure your exact starting value is correct.

**Part C.** While someone holds the RA4 button down (*i.e.*, any time the level is low). Reset the timer value to the 3 second starting point and clear all the PORTB lights.

Now when you release RA4 there should be **exactly** three seconds before the lights come on. If you hold down RA4 or press it every 2 seconds the lights should never come on.

**Part D.** Change Timer 0 into a low priority interrupt that uses the low isr.

Note, this part will be graded by seeing that the demo still works as before and by **looking at your code**. The demo will be the exact same as the prior part, so the instructor will need to see your screen when you do the check off.

When you complete this task call your instructor over to check off this problem. You can demo parts individually if you like or wait and demo just the last part.

## Problem 2 – Twinkle Twinkle Little Star

Start this problem from “template.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 turn your green PICDEM boards into a simple piano using the RA4 and RB0 buttons. Additionally you’ll put the sheet music statically displayed on the LCD. **Read all parts of the problem before you start planning and writing your code.**

Make sure that the jumper (J9) next to the piezzo buzzer on the green board is **connected**. Make sure that the jumper (J6) next to the LEDs on the green board is **NOT connected**.

**Part A.** Display this text on the LCD screen.

```
Twinkle Twinkle
4 4 0 0 40 40 0
```

This text will be used later (see Part D). The *4* represents that you should press the RA4 button. The *0* represents that you should press the RB0 button. The *40* represents that you should press both RA4 & RB0. Pretend like the buttons are piano keys.

**Part B.** Make the buzzer play 262 Hz (middle C).

**Part C.** Make the buzzer play 262 Hz only when RA4 is low (i.e. being pressed).

**Part D.** Make the buzzer play 392 Hz (middle G) when RB0 is low. Make the buzzer play 440 Hz (middle A) when both RA4 and RB0 are both low.

Now play your piano RA4, RA4, **RB0**, **RB0**, RA4 & RB0, RA4 & RB0, **RB0**

```
Twinkle Twinkle
4 4 0 0 40 40 0
```

Did it sound like Twinkle, Twinkle Little Star?

Now play your piano while singing...

```
RA4 (Twin-), RA4 (kle), RB0 (Twin-), RB0 (kle)
RA4 & RB0 (Lit-), RA4 & RB0 (tle), RB0 (Star)
```

When you complete this task call your instructor over to check off this problem. You can demo parts individually if you like or wait and demo just the last part.