

## ME 430 Exam 2, Fall 2011-2012, 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).
- A pencil/pen.
- 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	Checkoff
<b>1</b>	<b>/15</b>	<b>/2</b>
<b>2a</b>	<b>/15</b>	<b>/2</b>
<b>2b</b>	<b>/20</b>	<b>/2</b>
<b>2c</b>	<b>/8</b>	<b>/2</b>
<b>3a</b>	<b>/15</b>	<b>/2</b>
<b>3b</b>	<b>/15</b>	<b>/2</b>
<b>Sub-Totals</b>	<b>/88</b>	<b>/12</b>
<b>Total</b>	<b>/100</b>	

**Problem 1 –Potentiometer ADC LED “Meter”**

In this problem you will be creating a program that causes the PortB LEDs on the PICDEM board to light up from left to right, representing the current value read from the RA0 potentiometer.

IMPORTANT NOTE: For the RB3-RB0 LEDs to function you need to **make sure the jumper, J6, is connected**.

Start this problem from “template.c” but rename the file “Problem 1 (your name).c”.

When the ADC reading from potentiometer RA0 is between 0-250, only the left-most LED should be lit up. When it is 251-500, the two left-most LEDs should light up. When it is 501-750, the three left-most LEDs should light up. Any reading 751 and above should light all four LEDs. The lights should continuously update based on the current ADC reading.

**Problem 2 – PWM and RB0 Interrupts**

In this problem you will be playing notes using the piezo buzzer.

IMPORTANT NOTE: For the RB0 button to function you need to **make sure the jumper, J6, is disconnected**.

Start this problem from “template with interrupts.c” but rename the file “Problem 2 (your name).c”.

**Part A** – When the RB0 button is pressed, play 440 Hz for 1 second with a 10% duty cycle. You should code this using the 4 MHz external clock. You must use interrupts to initiate playing the note for full credit. (Hint: You may use a delay command to hold the note for 1 second.)

**Part B** – Play a minor pentatonic scale in sequence starting from 440 Hz A—each press of RB0 plays the next note in the scale for 1 second. The frequencies are approximately [440, 523, 587, 659, 784, 880] Hz. After playing the 880 Hz, the next press will return to 440 Hz. Again, you must use interrupts to initiate playing each note for full credit. (Hint: You may use a delay command to hold each note for 1 second.)

NOTE: Do not attempt to generate the frequencies by an algorithm; instead store them (or their corresponding timer ticks) in a static array.

**Part C** – Repeat Part B so that each note continues to play on the buzzer as long as RB0 is pressed and stops when RB0 is released.

**Problem 3 – LCD and timer interrupt**

In this problem you will be creating a program that displays a series of perfect cubes to the LCD display at specific time intervals.

Start this problem from “template with interrupts.c” but rename the file “Problem 3 (your name).c”. Also include LCD Module.h and LCD Module.c.

**Part A** – Write an the message ‘F ( 4 ) = 6 4’ to the top row of LCD using variables in your code to represent the numbers. *Note there are two spaces to the left of each number.*

**Part B** – Have your code display the perfect cube series (1, 8, 27, 64, 125, ...) using the format above. Update the LCD display at 0.5 second intervals using a timer interrupt (Hint: The calculations and display function call should both happen within the interrupt). The formula for the perfect cube sequence is

$$F(n) = n^3 \quad (1)$$