ME 430 Exam 2, Winter 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
1	/ 15	/2
2	/ 20	/2
3a	/ 10	/2
3b	/ 25	/2
3c	/ 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 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.

Problem 1 – ADC Lights

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

Create a program on your green board that will continually display the state of the vertical joystick (which BTW is connected to pin RA3) as follows…

ADC value	LEDs that are on
0 – 99	RC3, RC2, RC1, RC0
100 – 199	RC2, RC1, RC0
200 – 299	RC1, RC0
300 – 399	RC0
400 – 599	No LEDs on
600 – 699	RC4
700 – 799	RC4, RC5
800 – 899	RC4, RC5, RC6
900 – 1023	RC4, RC5, RC6, RC7

When you complete this task program your green board, remove the PICKit, and 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 – One second LED

Start this problem from "**template.c**", but rename it to "**Lastname_Problem2.c**".

Create a program on your green board that will…

- Start with all the LEDs off
- Turn on the LED for RC7 for exactly 1 second from the moment RB3 is pressed
- If RB3 is pressed very quickly (for say 0.1 seconds or less) the LED should be on for a total of one second.
- If RB3 is held down and simply not released (for say 10 seconds) the LED should be on for a **total** of one second. In this case YES the LED goes out while the button is still pressed. So 1 press = 1 second of LED on time, regardless of the press length.

For this problem you are **required** to use delays (instead of timers) and an oscillator speed of 250 kHz. Before starting the code do your calculations and answer the questions below:

Clock = _____250 kHZ_____

Number of instruction cycles needed for 1 second = _____

What is the code you will write to cause this 1 second delay:

When you complete this task program your green board, remove the PICKit, and call your instructor over to check off this problem. Your instructor will watch your demo **and** look at your answers above. You can wait to submit this problem to Moodle until you finish all problems.

Problem 3 – Feed the Fish

Start this problem from "**template with interrupts.c**", but rename it to "**Lastname_Problem3.c**". Also add "**LCD Module.h**" and "**LCD Module.c**" to this project.

In this problem, you will display the number of seconds remaining until it's time to feed the fish again. The fish need to be fed every 6 seconds (they are hungry fish). So the LCD will start with this message (and the LEDs will be off):

F	E	E	D		T	H	E		F	I	S	H		I	N
6		S	E	C	O	N	D	S							

The number on the bottom will decrement every second. For example, after the program has been running for 5 seconds it will say:

F	E	E	D		T	H	E		F	I	S	H		I	N
1		S	E	C	O	N	D	S							

*(Yes, I realize it says 1 SECOND**S**, but that's fine)*

After one additional second (6 seconds total) display nothing on the LCD, but have **ALL the PORTC LEDs turn on.** Have the LEDs and LCD stay in this state until the fish are fed.

You feed the fish by pressing RB2. At the moment you press RB2 the LEDs turn off again and the LCD returns to the initial state with 6 seconds and counting starts fresh. A press to RB2 will reset the state to a fresh 6 seconds regardless of the current state. For this problem you are **required to use a debounced pushbutton interrupt** falling edge, low priority.

Note on debounce code: For full credit you are required to use the version of debouncing that puts the delay directly into the ISR using a 30 millisecond debounce time. To be honest, that is the easier debouncing approach anyway and it works just fine for this problem. So no main event flags ~= |= **stuff**. ☺

Additionally for the 1 second timer you may not use a delay function, instead you **are required to use a timer 0 interrupt** low priority with a 1:32 prescaler and a 4 MHz clock.

Clock = _____4 MHZ_____

Timer 0 Prescaler = _____32_____

WriteTimer0 (_____); // To get the interrupt to fire in 1 second

You can work this problem all together or in parts. You instructor will watch your demo and check the value above. The breakdown for parts is on the next page.

Part A. Initial LCD display

In this part display the starting message on the LCD screen like this…

F	E	E	D		T	H	E		F	I	S	H		I	N
6		S	E	C	O	N	D	S							

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

Part B. Countdown

Now make your display count down from 6 to 1. Free hint: Make sure the first second is right by setting up the timer initially! When the time remaining hits 0 clear the LCD

and make the all the PORTC LEDs come on. Note when the time remaining is >0, the LEDs are all off.

Part C. Feeding the fish

Finally add RB2 to reset the state back to the initial value (6 seconds remaining with the LEDs off). You must use interrupts and debouncing.

When you complete any part or the entire project, program your green board, remove the PICKit, and call your instructor over to check off your demo.

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.