

CSCI 4415 Lab # 1



Hello World

Created: January 17, 2011

Updated: January 27, 2012

- Clarified the requirements

Objective:

- Develop a reusable software library that displays numbers and text on the LED array of the ZNEO Contest Kit.

Time Estimate:

- 10 hours. You should be prepared to start this after class 3 (GPIO, Interrupts, Basic Timers) and this will be due by Midnight class 6. So you have 3 weeks to complete this lab.

Specific Requirements:

1. On start-up program scrolls "Hello World from ZNEO" across the display repeatedly. Scrolling should be 1 character at a time (not 1 LED column at a time)
2. Implement function to display text (scrolling of more than 4 characters), a function to set the scroll rate, a function to display integers in decimal or hexadecimal, and a function to display floating point numbers.
 - a. You can use the library matrix.h (on the class website files page) if you want (not a requirement).
 - b. You should set the Z16 oscillator source to either 5.5Mhz or 18.432Mhz using the code in oscillator_configuration.c (on the class website files page) as discussed in class.
3. Pressing buttons will demonstrate the capability of your program.
 - a. Button 1 board uptime in seconds (as an integer, keeps updating until you press another button).
 - b. Button 2 Displays a floating point number (not a string) to at least 6 digits (pi or e or something)
 - c. Button 3 Displays scrolling message "Hello World from ZNEO" in a loop.
 - d. Button 1 twice displays uptime in MS (as an integer, keeps updating until you press another button).
 - e. Button 2 twice displays a uptime in seconds in hex (keeps updating it).
 - f. Button 3 twice displays your name + "CSCI 4415" in a loop.
4. Display should be able to show any of the 7-bit printable ASCII character (32-126).
5. The proper board orientation (for the message text) should be consistent with the printing on the circuit board, so the LED array is at the top and the buttons are at the bottom.

6. You should scan the display (either vertically or horizontally) rather than drawing the digits.
7. Do not add wires to the board to enable interrupts on the buttons.
8. You might need to implement some method for de-bouncing the buttons.
9. You should only use ONE Interrupt and ONE timer for the whole lab.
10. Your main() function should be in a file named main.c (any other files can be named as you choose). In the main.c file include the following information in a comment block

```

Author: [Your name]
Email:  [Your email]
Class:  [CSCI-4415]
Date:
Lab:    [Lab number]
Description: [short description of program. Include changes you made and
note any enhancements that you made to the lab]
Other files: [list other files that are necessary for this program]
Compile: [provide any special instructions for compiling. Only necessary if
there are special instructions]
Problems: [explaining what you did, what problems you had, how you solved them,
and what you might do differently if you had to do it again]
Comments: [Feel free to provide comments on how this lab went, what you think
is good or bad about it and how it could be improved or anything else you want
to say.]
Enhancements: [describe the extra capability you added to this lab]
Answers: [answer the questions below]

```

11. Turn in a zip file (named lastname-lab1.zip) of the entire project directory (include the **entire** ZDSII project directory and the readme.txt). Use BlackBoard to submit assignments (or if that's not working for some reason email the ZIP file).

Grading

- On time, in proper format (Zip file), named <lastname>-lab1.zip
- Compiles and runs without problems
- Meets lab requirements
- Reasonable code (partitioning of function, coding standard, readable)
- Reasonable comments
- Display is correct
- Proper use of Z16
- Did you set the oscillator speed
- Answered questions
- De-bounce switches

Possible Enhancements:

- Keep track of time and display that (hh mm) when Button 1+2 are pressed (then button 2 lets you increment through hours and button 3 lets you increment through minutes to set the time). Start time at 12:34
- Count seconds on the 4 modem status LEDs (you will need to add 4 wires, use the convention shown in class of PF0, PF1, PF2, PF3 for LED D10, D9, D8, D7)
- Implement reverse scrolling also.
- Implement scrolling 1 LED column at a time (1/5 of a character).
- Implement dimming and brightening of characters.

Questions

Include with your lab submission (in the main.c file in the comments):

1. Did you order the display characters by column or row? Does it matter?
2. Did you find that your display blinked or flickered, or that the characters seemed dim, or that one row or column of dots was brighter than the rest? If you solved that, what was the problem? If not, what do you think the problem is?
3. Briefly describe how did you de-bounce the switches?