EGR: 226 Microcontroller Programming and Applications

Winter 2021

Instructor Prof. Trevor Ekin

**Lab 7: Interfacing the MSP432 with a Liquid Crystal Display (LCD)**

Gabriel Gasbarre

March 16th, 2021

Table of Contents

[1. Objectives 3](#_Toc66820145)

[2. Equipment 3](#_Toc66820146)

[3. Introduction 4](#_Toc66820147)

[3.1: Pre-Lab: 4](#_Toc66820148)

[3.2: Part 1- Blinking the cursor on the LCD 4](#_Toc66820149)

[3.3 Part 2 - Displaying name on the LCD 4](#_Toc66820150)

[3.4 Part 3 – Scrolling Marquee 5](#_Toc66820151)

[3.5 Part 4 - EXTRA CREDIT Printing Float numbers to the screen 5](#_Toc66820152)

[4. Procedure: 5](#_Toc66820153)

[4.1: Part 1 - Blinking the cursor on the LCD 5](#_Toc66820154)

[4.2: Part 2- Displaying name on the LCD 6](#_Toc66820155)

[4.3 Part 3 - Scrolling Marquee 7](#_Toc66820156)

[4.4 Part 4 - EXTRA CREDIT Printing Float numbers to the screen 7](#_Toc66820157)

[5. Results: 8](#_Toc66820158)

[5.1 Part 1 Results 8](#_Toc66820159)

[5.2: Part 2 Results 8](#_Toc66820160)

[5.3: Part 3 Results 8](#_Toc66820161)

[5.4: Part 4 Results 9](#_Toc66820162)

[6. Conclusions. 9](#_Toc66820163)

1. Objectives

The objectives for lab 7 are as follows:

To interface an LCD with an MCU using I/O signals for data and control

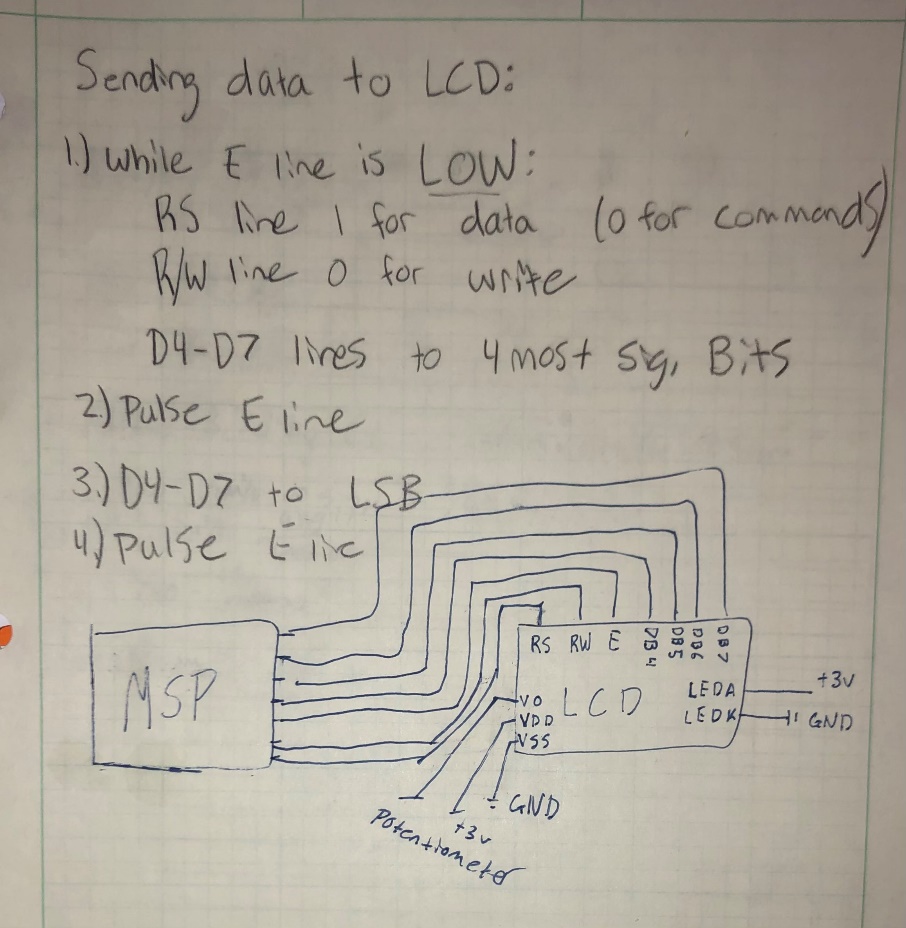
To employ the MSP432 SysTick Timer peripheral in creating precise timing intervals

1. Equipment

|  |  |  |
| --- | --- | --- |
| **Part** | **Description** | **Model** |
| CCS (Code Composer Studio) | Integrated development environment to develop applications for Texas Instruments embedded processors. | 10.0.00010 |
| MSP432 | Mixed-signal microcontroller family from Texas Instruments. | MSP432P401x |
| EGR:226 Lab 6 Exercise | Interfacing a keypad with the MSP432 | N/A |
| LCD | EGR 226 Lab Kit LCD display with HD44780 controller | HD44780 |

1. Introduction

3.1: Pre-Lab:



3.2: Part 1- Blinking the cursor on the LCD

Part 1 of the lab simply asks students to use the sequence of operations given in the lab manual, along with a user-created Systick timer delay peripheral in order to initialize the LCD display correctly.

## 3.3 Part 2 - Displaying name on the LCD

For part 2 of the lab, students are asked to modify their program such that it displays their first name, last name, “EGR” and “226” on each of the four lines of the LCD respectively. These messages must be centered, and students must use C strings in their code in order to get full credit.

## 3.4 Part 3 – Scrolling Marquee

For part 3, students must modify their program such that it displays the message “LABORATORY OVER” first scrolling from right to left by one character every second, then back left to right. It was suggested by the instructor to, instead of using the hardware display shift functions, to instead use a custom function in order to avoid strings moving to the next line.

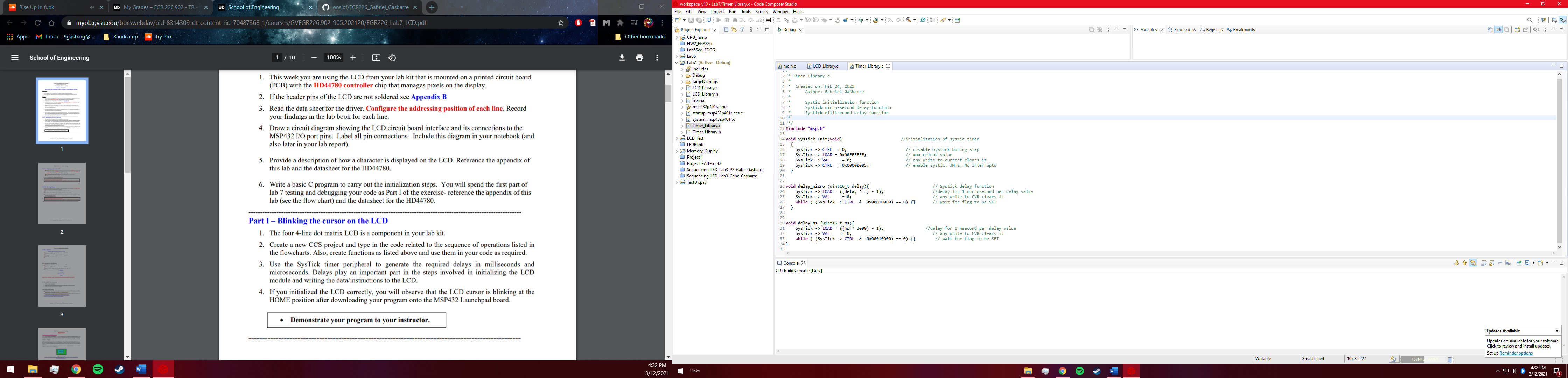
## 3.5 Part 4 - EXTRA CREDIT Printing Float numbers to the screen

For part 4, students are asked to print floating-point numbers to the screen. Though the lab manual offers a suggestion of a simple mathematics problem to display these numbers, the instructor has allowed some freedom such that other methods of calculating and displaying floating point numbers is allowed.

# 4. Procedure:

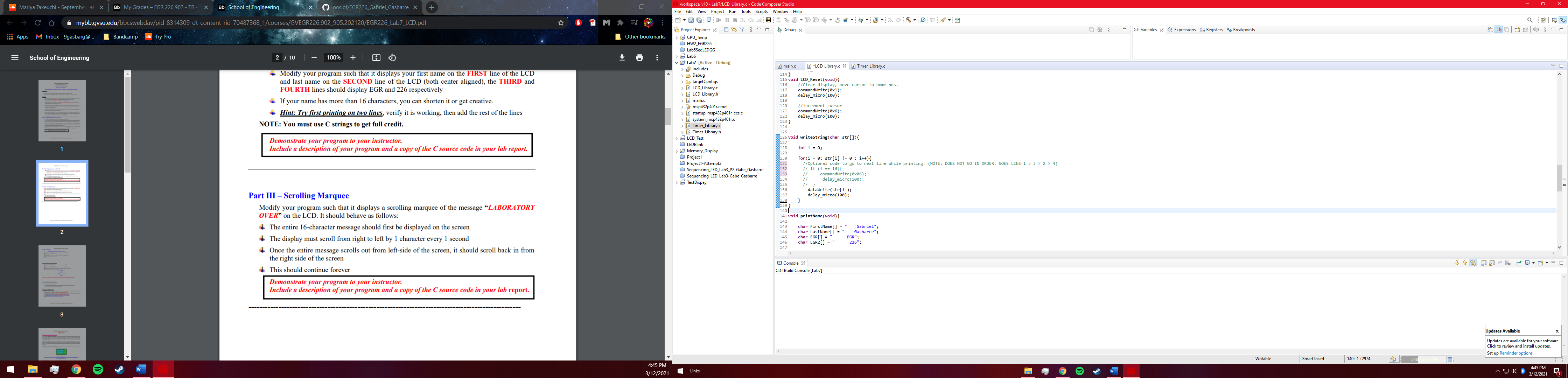
4.1: Part 1 - Blinking the cursor on the LCD

In order to complete part one of the lab, there are a couple of pre-requisites to simply writing the code sequence of operations as listed in the lab manual flowcharts. First, students must initialize all I/O pins in accordance with their pre-lab circuit diagram. Next, students must have the proper SysTick delay functions that allow for microsecond and millisecond delay to properly initialize the LCD. Proper usage of the SysTick timer is a key objective of this lab, and for that reason a sample of the SysTick delay functions has been included below. Finally, students must have working functions for writing commands and data, which is done by setting the RS line to 0 (commands) or 1 (data) and then passing a single nibble of information at a time. After all of these pre-requisites have been met, it is time to initialize the LCD by passing commands such as setting the LCD in 4 bit mode, turning display on and activating the cursor blink.



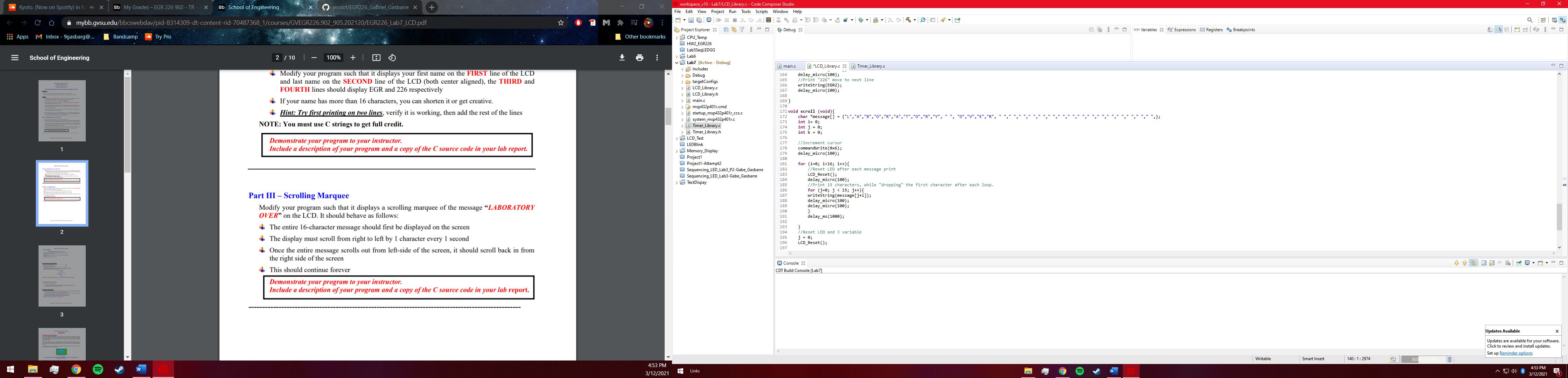
4.2: Part 2- Displaying name on the LCD

For part 2, most of the tough work has been done as students should have made proper functions to pass data and commands. To streamline the process of writing strings, a “Write\_String” function has been made that receives a string and prints character by character in order to display whatever the user desires. Luckily, this function also passes empty space characters, allowing us to easily center text by simply adding spaces in front of out strings. Alternatively, students could use hardware commands to move the cursor to the starting position of the centered text. In order to print each string on its desired line, the LCD receives a command after each string that moves it to the next line. Included below is the “Write\_String” function that receives each of the four stings required for part 2, and prints them respectively.



## 4.3 Part 3 - Scrolling Marquee

In order to manage certain issues introduced by hardware commands for part 3, a custom function was created to make the illusion of scrolling text on the LCD display. Instead of actually shifting the text over by one character per second, the program instead displays one character less of text every second, thus essentially moving the character off the screen and giving the illusion that the whole line of text is moving to the right when in actuality it is being re-printed every second. This function was then re-engineered to create the same effect while moving right to left. Included below is a snippet of the code that performs the right to left text movement.



## Part 4 - EXTRA CREDIT Printing Float numbers to the screen

Part four of the lab should, in theory be one of the easiest, however after certain altercations to the deliverables, something much more advanced was presented. Instead of simply displaying the result of a simple division problem, it was decided to display a live feed of the users active available CPU and memory percentages as floating point numbers. This was done though two programs. Firstly, a custom Python program was created to find and store the CPU and memory usage percentages in a file folder, updating about every quarter second. This file was then accessed by the MSP432 program and read in and stored as floating point numbers. Next, we convert each value to a string using the “sprintf” command, which also allows the addition of other text, such as the “%” symbol and “CPU = ’’ before the floating point number. After all of this was set up, it is as simple as passing this string to the LCD every quarter second in order to display a live, floating point indication of the CPU and Memory usage of the users computer.

# Results:

* 1. Part 1 Results

After much failure and frustration, The LCD initialization was eventually completed and showed promise for the rest of the lab. Everything appeared to work correctly, with the cursor blinking at the home position and the rest of the LCD cleared. This portion of the lab was by far the largest learning opportunity as there were many function that had to be combined in order to create this simple result, however it was very rewarding as it paved the road for the rest of the lab.

5.2: Part 2 Results

Part 2 of the lab presented more difficulties, such as correctly displaying text on the desired line, along with centering the text, however these problems were eventually solved with additional hardware commands, leaving the LCD correctly displaying the four lines of text. One issue with this program is that it does not automatically center the text, it must be done manually with empty space characters or cursor placement hardware commands. It would be interesting to eventually receive the number of characters in a string and automatically determine where to place the cursor to automatically center text.

## 5.3: Part 3 Results

Part 3 of the lab was simple in theory, all that was to be done was shift the text over on the screen using the display shift hardware command, however this created multiple problems as text was then being shifted to the next line and displayed in places that were undesirable. This led to the pseudo solution of making a custom function, however this was not very visually appealing and much less efficient than using a single command function. By far this was the least successful part of the lab (even though all requirements were met) simply because of the amount of code and the quality of code that was made in order to create such a simple output.

## 5.4: Part 4 Results

Part 4 was by far the most successful part of the lab from the standpoint of going beyond the lab deliverables. Instead of doing the bare-minimum, a working program was created that actually does something useful by displaying valuable information to the user. Though it does work consistently, there are still certain issues that should eventually be ironed out, such as infrequent display glitches that require screen refreshes, along with the fact that two separate coding programs in two different languages must be used to get the LCD properly working. Eventually, it would be nice to have this program in a single concise program that combines all the functions of each, for easy launch and upload.

# Conclusions.

By far this was one of if not the most successful and entertaining lab to date, as it provided significant education in library function combination along with open ended deliverables that allowed for significantly above requested results, specifically in the extra credit section. Though there are many efficiency related bug fixes that could be implemented, the fact that a working program has been provided is personally satisfactory.