ECE 241 **Lab 4** **Fall 2017**
September 19-25, 2017 **Points: 20**
Lab 4: LCD Displays

Objective: Setting up the library software and using it to control the LCD.

Description of Lab 4:

In many cases we will want a simple way to communicate with the user, using something simpler than a terminal connection. In these cases a Liquid Crystal Display (LCD) is a common option. Arduinos have a software library that supports the use of LCDs and in this lab we will be employing this library.

A standard interface has been established for LCDs and uses either an eight or four bit interface. Rather than writing up a long description of the LCD, you are referred to the following web page.

https://www.arduino.cc/en/Tutorial/LiquidCrystal

Another reference that refers to the hardware for this course is posted on the course webpage at

Files->CourseNotes->Hardware Documentation->A_LCD_Demonstration.pdf

In our case, we will be using a four bit interface to reduce the number of pins used. The following program description will show how the software is to be setup up to match the hardware for this class.

Program Structure:

```
// Import LCD Library
#include <LiquidCrystal.h>

// Defines for LCD
LiquidCrystal LcdDriver(11, 9, 5, 6, 7, 8 );
int count;

SetUp:
    Start LCD
    clear the display
    zero count

Loop:
    Every 500 milliseconds
        Increment count
        Clear Display
        Display count
```

*Notice that the first lines are exactly what is needed to interface to the LCD. The pin numbers in the definition of LcdDriver are critical and are based on the traces from the schematic.

ECE 241 **Lab 4** **Fall 2017**
September 19-25, 2017 **Points: 20**

Functions that may be needed for this lab.

LcdDriver.begin(columns, rows), LcdDriver.setCursor(column,row), LcdDriver.clear(), and LcdDriver.print( )

Lab Assignment:

Prelab: Write the program described in the above program structure section and upload it to the prelab link on Canvas.

Lab 4:

1) Load the prelab program to your Arduino and have it checked off by the lab instructor.

2) Rewrite the program to have it display the time on the second line of the display, using the HH:MM:SS format. Use the 24-hour format for the hours, in other words, have the time go from 00:00:00 to 23:59:59.

   Code example of clock can be found at Files->CourseNotes->B_Programming-> Support Code->ClockBasics.h An example of using this code is included in a posting called G_SimpleClock_STD.pdf, which will be discussed in class. It should be noted that the function SendClock() will need to be adapted to work with the LCD.

   You will need to test the functioning of your clock. Be sure to document how the clock's operation was tested.

Questions:

1) How would you test the accuracy of the clock you have built? In other words, discuss how you would test if your clock is counting seconds accurately?
   How long would it need to run for you to be confident that it is accurate?

2) In order to test most of the roll over cases (such as 23 hours to 0) to what time would you initialize your clock, so that you can observe this quickly?