Lab 7 – LCD
October 15, 2015

In this lab you will use the HCS12 microcontroller on the SSMI board to display characters on a liquid crystal display (LCD). At periodic intervals, an analog voltage will be sampled and the resulting digitized value will be displayed on the LCD. The periodic intervals will be generated using the timer from the real-time interrupt (RTI) system.

1 Prelab Questions (complete prior to coming to lab)

In the lecture we covered how to initialize the LCD and send a nibble to it. On the homework, you should have written the following two C functions (if you didn’t, please do so prior to lab):

- `void writeByteToLCD(char b, char rs, int t)` - This function sends a byte to the LCD. It should call `writeNibbleToLCD` twice, first for the high nibble, then for the low nibble, then wait the desired delay time t (where t is in microseconds).
- `void printLCD(char mystr[])`. This function sends a character string to the LCD.

Also do:

1. Create a schematic for the connections between the HCS12 and the LCD, indicating the names and numbers of the pins. Include on your schematic a 20KΩ “trimpot” to adjust the contrast of the LCD.

2. Write a C statement that calls the standard C function “`sprintf`”. Your statement should create a character string called `mystr`, that looks like this: “My number is xxx”, where xxx is the value of a number in a character variable `ch`. If you are not familiar with “`sprintf`”, you can look in any book on C, or check out the description at http://www.elook.org/programming/c/sprintf.html.

2 Connections

Make the connections between the HCS12 and LCD, according to your diagram, and show the instructor. Include the little trimpot potentiometer to adjust the contrast. (Note - it is always a good idea to verify the leads of a potentiometer before using it, using the multimeter.) Then connect your power supply.
3 Print a string

Write a C program that initializes the LCD and then writes a string to it, by calling the function “printLCD”. The string that you pass in can be a constant string, such as that created by a C statement such as:

```
char mystr[] = “This is a string”;
```

Sign-Off 1: Demonstrate your program to the instructor.

4 Photocell

A photocell (or photoresistor) is a resistor whose resistance decreases with increasing incident light intensity. We will use cadmium sulphide (CdS) cells, which are very inexpensive and can be found in many items such as street lights, clock radios, and alarm devices (i.e., applications where the slow response of the sensor is not important). The photocell in the kit has a resistance of about 1k Ohm in full brightness, and a resistance of about 10k Ohm in full darkness. (You may want to verify this with the ohm meter.)

By putting the photocell in a voltage divider circuit, you can measure its response. Referring to the circuit shown in Figure 1, what do you expect Vout to be when the photocell is in full brightness? What do you expect it to be when the photocell is in full darkness?

Place the photocell on the protoboard area of the SSMI board, and connect it to a 5V power supply (you can use the Vsharp output of the SSMI board). Connect a 10k Ohm resistor as shown (a discrete resistor or the DIP part), and verify with the multimeter that Vout is what you expect.

5 Digitized value

Connect the output of the photocell circuit to pin AN02 of the HCS12. Add instructions to your C program to initialize the A/D converter and continuously digitize an input voltage on pin AN02 (see your work in Lab #5 if you forgot how to do so).
Now print the digitized value from the A/D converter to the LCD. You will create the character string using “sprintf”.

Capture the waveforms of DB4 of the LCD, along with the E (enable) signal. What are your observations? Identify where the bits of a nibble are, and where they are clocked in.

**Sign-Off 2:** Demonstrate your program and data waveform to the instructor.

### 6 Periodic Timeouts

Finally, use the RTI system to time intervals of approximately 0.5 seconds. You can use a polling approach to test the RTI flag, rather than interrupts. At intervals of 0.5 seconds, get the latest A/D value and print the value to the LCD.

**Sign-Off 3:** Demonstrate your program to the instructor. Include in your report:

1. The C language program, along with a corresponding flowchart or pseudo code.
2. An explanation of the circuit and program.
Lab 7: LCD

Name: ________________________________ Name: ________________________________

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Initials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign-Off 1</td>
<td>Print a String</td>
<td></td>
</tr>
<tr>
<td>Sign-Off 2</td>
<td>Digitized Value</td>
<td></td>
</tr>
<tr>
<td>Sign-Off 3</td>
<td>Periodic Timeouts</td>
<td></td>
</tr>
</tbody>
</table>

7 Rubric

<table>
<thead>
<tr>
<th>Deliverables</th>
<th>20 pts</th>
<th>Pre-Lab</th>
<th>5 pts</th>
</tr>
</thead>
<tbody>
<tr>
<td>C code with flowchart/pseudocode</td>
<td>/ 10</td>
<td>Questions</td>
<td>/ 5</td>
</tr>
<tr>
<td>Circuit and program Explanation</td>
<td>/ 5</td>
<td><strong>Demonstrations</strong></td>
<td><strong>20 pts</strong></td>
</tr>
<tr>
<td>Scope traces</td>
<td>/ 5</td>
<td>Sign-Off 1: Print a String</td>
<td>/ 5</td>
</tr>
<tr>
<td><strong>Composition</strong></td>
<td><strong>5 pts</strong></td>
<td>Sign-Off 2: Digitized Value</td>
<td>/5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sign-Off 3: Periodic Timeouts</td>
<td>/10</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>50 pts</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
