Lab #7: LCD Display and Real Time Interrupt (Stop Watch)

Purpose

In this lab you will add an LCD array to your latch you memory mapped in the last lab. You will then write code to send out your name to this new display. In addition to this hardware/software you will also create an interrupt routine to create a simple stopwatch using the 6812's RTI.

Part I. LCD Display

A short technical document is attached to this lab that describes how to interface to an LCD array.

1. Consult the attached LCD document for pin-out information and an example circuit. Instead of connecting the LCD directly to the 68HC12 data bus, you will interface it to the output of your 574 latch as follows:

<table>
<thead>
<tr>
<th>574</th>
<th>LCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q0</td>
<td>=&gt;</td>
</tr>
<tr>
<td>Q1</td>
<td>=&gt;</td>
</tr>
<tr>
<td>Q2</td>
<td>=&gt;</td>
</tr>
<tr>
<td>Q7:4</td>
<td>=&gt;</td>
</tr>
</tbody>
</table>

   DB7:4

   Note: You should ground /OE on the 574.

2. We will create a synthetic write cycle by setting the pins appropriately on the 74HC574. You should look at the documentation for timing diagrams for the LCD read & write cycles.

3. Consult Software/Docs link under LCD for further information on the part. The LCD will be used in 4 bit mode but automatically powers up in 8 bit mode. See the 4 bit mode initialization details.

4. The LCD can be used as a ‘write-only’ device where a software delay is required between the writes of each nibble. Thus this is why we can get away with connecting the LCD data bus directly to our latch outputs instead of the 6812’s data bus. By placing the 574 between the LCD, we add an extra measure of protection to the 6812. Upon wiring mistakes, we prefer you destroy the 574 instead of the 6812.

5. Upon wiring in the LCD, write code to send out your name to the LCD. This must be a minimum of 8 characters with no maximum character amount specified.

Additional Notes:

It is possible to initially test the LCD using the 6812 monitor commands. See the initialization procedure in the LCD Interface Note. More detailed information (initialization sequence and control commands can be found at:

http://mil.ufl.edu/4744/docs/lcdmanual/lcdmanual.html

You can find the ASCII table in your M&M text book Appendix A.

Part II. Stopwatch

We will now use the Real Time Interrupt module in the 6812 to create a stopwatch using your keypad and the LCD display.

Use your keypad such that if a "1" is hit, start the stopwatch measuring time. This time should then be displayed on your LCD in the format:

Elapsed Time: YY:Z

;where YY are seconds and Z is tenths of a second.

Hit “1” on your keypad to start the stopwatch and “1” to stop/pause the stopwatch and then use “0” to clear the stopwatch to zero. To operate, press “1” to start the counting. Once running, press “1” again to stop or pause the counting.

“0” => Clears stop watch at any time.

“1” => Pressed the very first time, acts as a start button. Pressed a second time after running, it stops the time. If pressed again, the time resumes. Pressed again, the time is paused and this repeats.
Lab #7: LCD Display and Real Time Interrupt (Stop Watch)

Specifics:

Your main program should read the keystroke pressed to start/stop/clear the stopwatch. Global variables should be used for these operations such that information can be easily obtained from inside the interrupt service handler.

The ISR should measure time and update the LCD display.

RTI Interrupt Handler Hints:

1. See your 6812 manuals (or UF6812 pseudo interrupt vector table) to determine where the hard vector is for the RTI. This address will be somewhere in the high memory of the EEPROM.

2. Read the data bytes at this address using your monitor. This should now point to somewhere in internal 6812 SRAM. This is the re-directed vector location.

3. Write code to allow the assembler to automatically place a jump instruction and your ISR's label at the locations corresponding to the redirected vector in SRAM.

4. See our 6812 documentation to learn how to enable the RTI and allow it to fire again once an interrupt has been serviced (ISR code is executing).

5. To test if the interrupt is running initially, just write test code that reads a memory location and then increments it by one every interrupt. Run the interrupt for several seconds, and then reset your board. Next, use our monitor to see if this test location has incremented. The interrupt must be firing regularly before you can proceed on to any additional code.

What is the resolution of our Stopwatch?

Part I. Lab Requirements:

1. Show your TA a hand drawn schematic for how the LCD is connected. (5%)

2. Draw a simple timing diagram of how you will write nibbles to the LCD. (5%)

3. Show your TA the list files for you LCD code required to send out your full name. (Pre-Lab Materials) (15%)

4. Demonstrate writing your name to the LCD display to your TA. Answer questions that your TA will ask. (25%)

Part II. Lab Requirements:

1. Show your TA the list file for the RTI code written to create the stopwatch. (Pre-Lab Materials) (25%)

2. Demonstrate a functional stopwatch to your TA. Answer questions that your TA will ask. (25%)