CpE5160 Experiment #5
Software State Machine Design
(40pts, Due Wednesday, December 4)

System Description

The purpose of this experiment is to create a simple light display using the simple Embedded Operating System. The light display will be controlled by the switches and they will needed to be debounced for easier control of the display. The display will operate as described below:

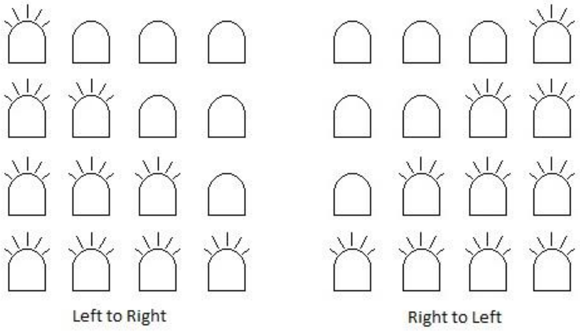

Figure 1: LED Sequence

Selecting a display:

SW1: When SW1 is pressed, the LED display will flash from left to right as shown in figure 1. When SW1 is pressed again, the display will switch off. **(Func: 3pts)**

SW4: When SW4 is pressed, the LED display will flash from right to left as shown in figure 1. When SW4 is pressed again, the display will switch off. **(Func: 3pts)**

The programmer may choose to have the display jump from one display to another when that switch is pressed or only jump to a display from the off position. The display can have an all off stage, if the programmer wants one.

Adjusting the delay:

SW2 and SW3: Pressing SW2 or SW3 will cause the system to enter the delay adjustment mode and the delay amount to be displayed on the LEDs as 1 to 15 in binary. **(Func: 4pts)** This represents about 60ms to 900ms of delay. While in the delay adjustment mode, pressing SW2 will cause the delay to increase by one (about 60ms for each increment) up to the maximum value of 15 (900ms). Pressing SW2 at the maximum value should not cause the delay to increase. **(Func: 4pts)** Pressing SW3 will cause the delay to decrease by one (about 60ms for each decrement) down to the minimum value of 1 (60ms). Pressing SW3 at the minimum value should not cause the delay to decrease. **(Func: 4pts)** Pressing SW1 or SW4 will exit the delay adjustment. **(Func: 2pts)**

The programmer may wish to have the system enter the delay adjustment at any time that SW2 or SW3 is pressed or only when the display is off. The switches need to be debounced to prevent multiple increments or decrements when setting the delay.

Name:________________________________

Step #1: (10pts)
Draw a state diagram of the system described above. As you program, you may need to mark modifications or redraw the state diagram to accurately represent your system. You can submit using the space provided below or on your own paper.

Step #2: (20pts)
Implement the system describe above on your project board using the simple Embedded Operating System (sEOS) described in chapters 7 and 8 in "Embedded C" by Pont.

Organization & Readability: 5pts
Correctness: 5pts

Total: 40pts

Upload the code to Canvas or to younger@mst.edu

State Diagram: