Timer Interrupts

The purpose of this lab is to demonstrate the use of Interrupts by using the system Timer interrupt.

Part 1: Coding

Write a C-language program to run continuously on your TMS 432 board to do the following functions:

- Initialize: Configure the System Timer (textbook section 7.5) to generate a SysTick interrupt request (IRQ) at exactly 4 times every second.
- Perform other system initializations as needed.
- Write an Interrupt Service Routine (ISR) to handle the SysTick IRQ:
    - o Every 4th time through the interrupt (i.e. once every second of time) toggle the Blue LED on the BoosterPack board. This LED should blink with a period of 2 seconds: on for 1 second of time, off for 1 second of time.
- In the mainline code:
    - o Continually scan the S1 and S2 switches on the BoosterPack board and light the main board Red LED1 when S1 is pressed and main board Red LED2 when S2 is pressed.
    - o This copying of the switches to the LEDs is just like your earlier Lab on Basic I/O.
    - o The copying of the switches to the LEDs should run in the main code, independently of the Interrupt code.

- For full credit, the blinking of the Blue LED can be done within the ISR.
- For bonus credit, the blinking of the Blue LED should be done in the mainline code by the Safe Message Passing techniques discussed in class, where the ISR sends a message to the mainline code once every second of time.

Your program should perform any and all initializations to make sure the hardware is in the right configuration to do these tasks. You should ONLY touch the hardware that you need to manipulate to do this job! If you change any other bits of I/O hardware, your program is wrong, and would create problems in a bigger system.