CPE 325: Embedded Systems Laboratory
Laboratory Assignment #6

Assignment [100 pts]

1. **[50 pts]** Interrupts are useful when the main loop is busy doing some task, but the microcontroller needs react on some external events as soon as possible. Write an assembly program that implements such a scenario.

   Blink LED1 in the main loop with the frequency of 0.5 Hz (1 second on, 1 second off). Use software delay to produce the required time intervals. Once SW2 is pressed LED2 must toggle. LED1 must keep blinking. Your implementation should meet the following requirements:

   a. The blinking frequency of LED1 should not change due to debouncing delay.
   b. Pressing SW2 should cause an ISR call.
   c. LED2 must toggle as soon as possible (in just a few clock cycles after SW2 is pressed).
   d. No delays in the ISR are allowed. Remember: you want to exit your ISR asap, so that other hypothetical interrupts can be processed with no delays. Calling a delay subroutine in the ISR counts as doing it in the ISR.
   e. Just a single LED2 toggling must happen when SW2 is pressed (do the press debouncing).
   f. No toggling must happen when SW2 is released (do the release debouncing).

   You may use the following pseudocode as the basis for your program. You may also use your own approach if it meets the requirements above.

```
main_loop:
    sleep for 1ms
    if (...):
        toggle LED1
    if (press_debounce_counter > 0):
        press_debounce_counter--
        if press_debounce_counter == 0
            release_debounce_counter = 20
    if (release_debounce_counter > 0 AND sw2 is released):
        release_debounce_counter--
        if (release_debounce_counter == 0):
            clear interrupt flag for SW2
            allow interrupts for SW2
ISR:
    If caused by SW2:
        forbid interrupts for SW2
        toggle LED2
        clear interrupt flag for SW2
        press_debounce_counter = 20
```

2. **[50 pts]** Implement the following functionality in C:
    a. Set the clock frequency of the microcontroller to 2,097,152 Hz (~2 MHz).
    b. Toggle LED2 in the main loop with the delay of 1048576 clock cycles.
    c. When SW1 is pressed change the microcontroller's frequency to 4,194,304 Hz (~4MHz).
    d. When SW2 is pressed change the microcontroller's frequency to 8,388,608 Hz (~8 MHz).
    e. Calculate the LED's blinking frequency for each clock frequency and show your work in the report.
    f. Use interrupts to react on the switches. Do not worry about debouncing for this problem.
3. **[Bonus 10 pts]** Rewrite the first program in C and extend it in the following manner:
    a. Pressing SW1 should stop and start blinking of LED1.
    b. When stopping, LED1 should remain in the same state. That is, if it was turned on when SW1 was pressed, it should remain on, until SW1 is pressed again.
    c. Process SW1 in the ISR.
    d. Take care of SW1 debouncing. All requirements from part 1 should apply.

Topics For Theory

1. Interrupts and Interrupt Vectors
2. Clock Module in MSP430

Deliverables

1. Your calculations
2. Source files