Lab4 Assignment

Note that all students are required to work on this lab assignment in teams of two. Teams cannot change during labs. The deadline for team member changes (email instructor) is Tuesday, March 21. A change in team members will result in a loss of bonus points for consistency/teamwork (i.e., same partner for all labs and project)!

For Lab4, one report per group is sufficient. All software development for Lab4 must be done in assembly language. Note that each team member must be able to explain all hardware and software components.

Objective
Gain some experience with advanced timer/counter functionality, pulse width modulation (PWM), interrupts, and LCDs.

Mid-lab review (March 29) - show the TA your progress towards completing the lab (see below)
Deadline: April 5 (see calendar on ICON for checkoff sign up)

Lab Description

In this lab, you will implement hardware and software to operate a cooling fan (see a TA for check out) at different speeds by utilizing a PWM approach. For user interaction, the system utilizes a LCD, a rotary pulse generator (RPG), and a pushbutton switch (PBS). To debounce the RPG and PBS, you can use a hardware-based approach. Suitable approaches have been previously discussed in lecture. The duty cycle (DC) of the PWM signal used to control the fan must be adjustable by turning the RPG (CW rotation: increase; CCW rotation: decrease). All details regarding hardware-based PWM will be covered in class, and the corresponding lecture slides and other relevant resources (e.g., data sheets) will be available on ICON.

The functionality to be implemented is given below.

- Build a suitable circuit connecting the fan, RPG, LCD, and push button to the microcontroller. Justify your design decisions.
- The PBS enables toggling the fan on and off. The switch action must occur immediately when the PBS is pressed. Adjusting the DC value should only be possible if the fan is on.
- Use Timer/Counter0's Waveform Generation functionality for generating a PWM signal on pin OC0B (PWM signal for fan). The PWM frequency must be set at 80 kHz (fixed), and the duty cycle must be adjustable between 1% and 100% by turning the RPG (increments <=1%). Alternatively, the 8-bit Timer/Counter2 can be used instead of Timer/Counter0.
- **Optional (5% bonus points):** Implement a fan speed monitoring functionality by using the fan's tachometer signal to detect when a) the fan stops running and b) the fan speed is below 2580 rpm (43 Hz).
- Use interrupts for implementing the functionality described above (PBS, RPG, …).

At all times, the following information must be displayed on the LCD: duty cycle in % and status. Follow the formatting examples given below.

LCD messages **without** speed monitoring implemented	LCD messages **with** speed monitoring implemented
DC = 78.5% **Fan: ON**	**DC = 78.5%** **Fan: RPM OK**
DC = 22.0% **Fan: OFF**	**DC = 22.0%** **Fan: stopped**
	DC = 3.0% **Fan: low RPM**
	DC = 15.0% **Fan: OFF**

Note: do not touch rotating parts! Keep number labels on the fan!

After completing the assignment, upload your source code and report to ICON (note that one source code file and report per group is sufficient). If you plan to see a TA on the day the lab is due, a sign up for check off is required (see calendar on ICON).

Mid-lab review — Show a TA your progress towards completing Lab4 (use ICON for sign up). Specifically, it is expected that at this point in time you can demonstrate that your program successfully utilizes the LCD as well as an 8-bit timer/counter for PWM signal generation to adjust the fan speed.