CSC 230 - SUMMER 2017
INTRODUCTION TO COMPUTER ARCHITECTURE
ASSIGNMENT 4
UNIVERSITY OF VICTORIA

Due: Sunday, July 30th, 2017 before 11:55pm. **Late assignments will not be accepted.**

1   Overview

This assignment covers low level C programming on the AVR architecture, and is the last piece of the applied programming component of the course. You are already familiar with the task: Implement an accurate stopwatch. In fact, the overall specification of the program is identical to assignment 3; the only difference is that your implementation must be written in C instead of assembly.

Consult the assignment 3 specification for details on the expected functionality of the program. Section 2 below describes the evaluation process for this assignment.

2   Evaluation

Submit all `.c` and `.h` files needed to assemble your assignment electronically via conneX (including all of the necessary LCD library files, even if you did not modify them). Your code must compile, upload and run correctly on the ATmega2560 boards in ECS 249 using the toolchain and methodology described in Lab 7. If your code does not compile as submitted, you will receive a mark of at most 2.

If your implementation is not written in C, or does not use the C programming environment covered in Lab 7, you will receive a mark of zero.

This assignment is worth 6% of your final grade and will be marked out of 12. The evaluation will not be demo-based.

The marks are distributed among the aspects of the assignment as follows. The distribution of marks between components is given in the table below. Note that most of the entries refer to sections of the assignment 3 specification.

Marks	Aspect
6	The basic, non-interactive timer described in Section 2 of the A3 specification functions correctly with reasonably accurate timing (such that a human observer cannot notice any discrepancy). Timing functionality must be implemented using one of the hardware timers, not software delay loops (including the `_delay_ms` function).
1	The timing is theoretically accurate to within 1 second out of 10 minutes when the 16Mhz system clock is assumed to be exact.
1	Global variables are used **only** for global, persistant data (such as the active timer state). Local variables are used for all other data and temporary storage.
1	The `SELECT` button correctly implements the start/stop feature described in Section 3 of the A3 specification. The code must ensure that a single press of the button be read only once (regardless of how long the button is held down).
1	The `LEFT` button correctly implements the clear feature described in Section 3 of the A3 specification.
1	The `UP` button correctly implements the 'set lap' feature described in Section 4 of the A3 specification. Note that unless the display of lap times is correct, it is impossible to evaluate this component (and the component below).
1	The `DOWN` button correctly implements the 'clear lap' feature described in Section 4 of the A3 specification.

Your code should be readable and, at minimum, contain header comments for each function (except main) describing the purpose of the function (and its arguments and return value, if applicable). You should also ensure that your source files contain header comments with your name and student number. If your code does not meet this criteria, a deduction of up to 2 marks may be applied.

You are permitted to delete and resubmit your assignment as many times as you want before the due date, but no submissions or resubmissions will be accepted after the due date has passed. You will receive a mark of zero if you have not officially submitted your assignment (and received a confirmation email) before the due date. Ensure that each submitted file contains a comment with your name and student number.

Ensure that all code files needed to compile, upload and run your code in ECS 249 are submitted. Only the files that you submit through conneX will be marked. The best way to make sure your submission is correct is to download it from conneX after submitting and test it. You are not permitted to revise your submission after the due date, and late submissions will not be accepted, so you should ensure that you have submitted the correct version of your code before the due date. conneX will allow you to change your submission before the due date if you notice a mistake. After submitting your assignment, conneX will automatically send you a confirmation email. **If you do not receive such an email, you did not submit the assignment.** If you have problems with the submission process, send an email to the instructor **before** the due date.