ELEC 5260/6260 Problem Set - STOPWATCH
Due Wednesday, 2/27/2019

To work with interrupts and to work further with software drivers, in this project you are to use Board Support Package (BSP) and Hardware Abstraction Layer drivers to assist you in designing a main program that implements a software-controlled counter, and an interrupt-driven stopwatch. These are to be displayed as decimal numbers on the board's LCD as follows.

$$S_{10}\,S_1\,.\,H_{10}\,H_1\,:\;C_{10}\,C_1$$

with the left-most four digits showing stopwatch time and the right-most two digits showing the software counter value. A decimal point should appear between the stopwatch seconds and hundredths as shown, and a colon between the stopwatch time and the counter.

The main program is to operate as a "background program", implementing a decimal counter that increments from 00 to 19, and repeats, with the count displayed as $C_{10}\,C_1$ (tens and units digits.) When the program starts, the count is to increment continuously, once per second, with timing created by executing a delay function that lasts approximately one second. For example, this function might execute some number of nested "for loops". If the stopwatch begins running and/or is reset, this counter should also reset to 00 and then continue counting. This will allow comparison of the counter and stopwatch timing.

The stopwatch is to be interrupt driven, with JOYSTICK interrupts starting, stopping and resetting the stopwatch, and capturing/displaying "split times". Precise timing is to be produced by SYSTICK timer interrupts, with time displayed as $S_{10}\,S_1\,.\,H_{10}\,H_1$ (second and hundredths of a second.) The displayed time should go from 00.00 seconds to 19.99 seconds, and then roll over to 00.00 seconds and repeat, with $S_{10}S_1$ being the seconds digits and $H_{10}H_1$ the hundredths of a second.

Interrupts from the JOYSTICK are to control the stopwatch as follows.

- **SEL = Start/Stop Button**. If pressed, the stopwatch should toggle from running to stopped, or vice-versa. While stopped, time should "freeze" on the display and continue from that value if the stopwatch is restarted, unless the Reset Button is activated.

- **UP = Reset Button**. Reset the time to 00.00 and discard any saved split times if the stopwatch is stopped; ignore this button if the stopwatch is running.

- **DOWN** = **Split Button**. If the stopwatch is running, capture and save the time in a circular buffer. The buffer should allow the four most recent "split times" to be saved, with any older splits overwritten. If the stopwatch is not running, display one saved split time per button press, up to the number of saved values (in place of the stopwatch time.) After displaying the number of saved split times, further button presses should restore the current stopwatch time to the display. (If no saved split times, then simply leave the current stopwatch time on the display.)

Print and submit your source programs (but not “startup” or other Keil/STM-provided driver files) ***and also email them to me***. Bring your programmed board to my office to demonstrate the program or provide a video demonstration.