Carleton University

Department of Systems and Computer Engineering

SYSC 5701 Winter 2014

Operating System Methods for Real-Time Applications

Assignment 2

Assigned: Tuesday, Feb. 04, 2014 (Version 1)

Due: Tuesday, Feb. 25, 2014, at the start of class (6:05 pm)

Submit: Hardcopy (paper) of your solution. No email solutions please.
Please ensure your name and student number appear on your solution.

Goal: further familiarization with your microcontroller board and development tools (interrupts), and do some simple FreeRTOS programming.

All programs must be written in C.

Begin your assignment submission with a brief statement of the target hardware and development environment that you are using. (No details, just the board, processor and development tools.)

PART 1

In assignment 1, the execution time of a simple for-loop was measured. By varying the number of loop iterations, the amount of execution time taken by the loop could be increased or decreased.

Using your results from Assignment 1, design a test loop that executes for about 2 seconds.

Verify the timing of your loop by embedding it in a simple test program that initializes a general purpose timer, reads the timer, executes the loop, and reads the timer again. Use the two timer readings to calculate the execution time.

Submit the (documented) test program, the two readings obtained, and the calculations that conclude that the loop executed for approximately 2 seconds.

PART 2

Extend you program in PART 1 to include the initialization of an additional timer (in addition to the one used to time the loop). Program this timer as a periodic interval timer with an interval of approximately 10 milliseconds. Program the interrupt system to generate an interrupt each time the periodic interval expires. The ISR should increment a variable that counts the number of times the ISR executes (hint: you might have to declare the variable as "volatile" for the compiler to generate code that "works").

Run the extended test program, and measure the execution time. The execution time should include the 2 seconds spent executing the loop (as in PART 1) + the time spent servicing the additional timer interrupts. Calculate the (approximate) overhead associated with servicing one periodic interval timer interrupt.

Submit the (documented) test program, the two readings obtained, and the calculations that result in the overhead result.

PART 3

Install FreeRTOS on your board and run a Blinky test program. The program should create a task that runs as a loop. The loop should start by toggling the state of an LED (i.e. if it was ON, turn it OFF, and vice-versa), and then the task should delay (xTaskDelay) for about 1 second.

Submit the (documented) test program and a brief description of any FreeRTOS configuration that was required beyond the "out of the box" installation for your board.