Carleton University

Department of Systems and Computer Engineering

SYSC 5701 Winter 2014

Operating System Methods for Real-Time Applications

Assignment 1

Assigned: Monday, Jan. 13, 2014 (Version 1)

Due: Thursday, Jan. 30, 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.

Please bring questions about the assignment to class. We will revise it if necessary to clarify any ambiguity. (This only works if you start work on the assignment early.)

Like all software, an operating system requires the execution of instructions by the processor. This means that some of the net execution time taken to meet deadlines will be due to operating system execution. The code executed by the operating system is generic, in the sense that it has been designed to support all applications. From the perspective of applications, the operating system introduces processing overhead that is not accounted for in application code.

Measuring program execution is essential to ensuring that a program meets timing requirements. Measuring execution time is not usually an easy task. The process often requires executing software associated with taking the measurements, and this software in turn increases the execution time. Fortunately, microcontrollers usually include a set of hardware timers that can be used to off-load some of the timing functionality to hardware (software still needs to initialize the timer and read the timer to gather timing information).

Goal: familiarize you with your microcontroller board and development tools. You will design and execute tests that measure the execution time of a loop iteration. This assignment does **not** involve using FreeRTOS. In this assignment, all tests will run as stand-alone programs.

All test programs must be written in C.

Begin your assignment submission with a brief description of the target board and development environment that you are using. Include only the information that you think is relevant to the assignment.

A simple for-loop is often used to "waste" time by delaying the execution of a program. For example:

```
for ( var i = 0; i < max_count; i++ )
{
        // do something simple here
}
```

By varying the number of loop iterations (i.e. the value of max_count), the amount of execution time taken by the loop can be increased or decreased.

Design and implement a test program that uses a hardware timer to measure the execution time of the loop. Run the program several times with different numbers of iterations and estimate the time taken for one iteration.

In the description of the test program, include:

- The source code.
- A brief statement about how test measurements are retrieved after running the program.
- The compiler settings used, and a discussion of any settings that are relevant to this test.
- A brief description of the initialization of the system hardware. Note: this must include any initialization performed by any board-level libraries that you include.
- A description of the hardware timer (and its initialization) used to measure execution time.
- A summary of the test runs and measurements taken.
- An analysis of the measurements to arrive at your estimate of the time taken for one loop iteration.

Obtain a listing of the assembly language generated for the test program. Isolate the part of the code that deals with starting the timer, the execution of the loop, and reading the timer. Annotate that part to describe what is happening at the level of register and memory use.

Based on the measurements and the assembly language listing, predict the accuracy of your estimate for one loop iteration.

If you ran the program with only one loop iteration (i.e. max_count = 1), would you expect the measured value to match your estimate? Why?