# ECE 4180 Lab 3 – Using the timer and mbed RTOS

**Due Date: Odd, Tuesday Feb 24**
**Even, Thursday, Feb 26**

**Names:**_____ _____

| Item | Lab Demo | Extra Credit |
|---|---|---|
| **Ticker blinking 4 LEDs** | | **n/a** |
| **Mbed RTOS - LCD, LED, & Sound Effect Code** | | |
| **Semaphore for the LCD mutex lock** | | **1%** |
| **Multiple RGB LEDs** | | **2%** |
| **Image or Video on LCD** | | **2-4%** |

(**Part 1** *20%*) Blink mbed's four built-in LEDs at different rates of 1, 2, 3, and 4 seconds using the Ticker function (no *wait()* allowed!). Ticker uses a hardware timer on mbed to periodically generate an interrupt that triggers a function to run. Ticker is similar to Timeout, but Timeout only works one time.

A periodic timer interrupt can be used to time slice the processors among various tasks and such hardware is used in an OS scheduler for time slicing among tasks. The mbed RTOS uses the timer to provide its 1 ms. time slice. With an RTOS, the scheduler makes it possible to run multiple threads and the RTOS also provides the needed synchronization primitives when threads share global variables or I/O devices. So once things get a bit more complex, an RTOS may be needed. Read the LED lighting effects wiki & code examples for an introduction to using the RTOS and threads.

(**Part 2** *80%*) Next, use the mbed RTOS to run at least four threads. Two threads put something on the uLCD display. Since the display requires a complex sequence of commands, mutual exclusion on the display is required and a mutex synchronization lock must be used in each thread before it can access the display. This avoids errors that could occur when the RTOS forces a switch between threads when they are in the middle of writing a command to the display. So, setup a single mutex that is always used in each LCD thread to lock and unlock access to the uLCD. Minimize the scope of the mutex lock by putting as many calculations as possible outside of the lock. Main always runs the first thread, so you only need to create three more threads.

The other two threads should do the following:

1. Use the RGB LED or Shiftbrite to display a new lighting effect. Some ideas are one of the multicolor airport beacons in the LED lighting effect wiki or perhaps a RGB fire effect that changes intensity and color. Another idea is a Sonic screwdriver (any Doctor that has a flashing one with sound).
2. Play a wave file from an SD card or USB flash drive on the speaker with sound effects appropriate for the LED light. Sound effect web sites have sound clips to use. The cookbook's waveplayer code will run in the RTOS without changes per LED effects wiki page.
3. The two LCD threads should display something related to the lighting and sound effects with two different time update rates in different areas of the LCD. Could be time, status, image, or video.

The easy way to set this up in the RTOS is by using Thread::wait(ms); in each thread at the end. There is a similar RTOS thread idea for a different LCD hardware setup on the mbed application board at http://mbed.org/users/4180_1/notebook/mbed-application-board-hands-on-demos/ if you need to see more RTOS code examples with threads, waits, and mutexes. There are similar graphics member functions for the uLCD display as in this example, but they do not have exactly the same names.

**Extra Credit: (1%)** Use a Semaphore to provide mutual exclusion instead of the mutex lock.
**Extra Credit: (2%)** Use more than one RGB LED in a meaningful way to improve the lighting effect. One idea would be to use perhaps three or more to also provide some spatial movement.
**Extra Credit: (2-4%)** Display an image (2%) or video clip (4%) on the LCD from the LCD's SD card in your solution. Can only do one of the two options and it must be related to the LED lighting effect selected. See uLCD wiki page for examples and more instructions on image and video file conversions for the raw format uLCD's SD card.