EE475 Lab #8 Fall 2005

MicroC/OS-II on the HC12

This week we will develop a real time project using the MicroC/OS-II software ported to run on the HC12. The kernel features required to create tasks, manage their execution, and communicate among the tasks will be investigated. There are literally dozens of other features provided by the μC/OS-II kernel: you should feel free to experiment as much as you like!

Preliminaries

1. Make a temporary local folder for your work: `c:\EEClasses\EE475\tempxxx` .
2. Copy the Lab8 project folder from the class web site to your temp directory.
3. Launch Code Warrior and open the Lab8 project.
4. Take a look at the `main.c` file and determine what it does.
5. Build the `main.c` program. Note that there will be lots of warnings due to the MicroC source code, but there should be no errors. Launch the debugger, and try the code. Does it work as expected?
6. Look through the various project `.c` and `.h` files to answer the following questions:

How many tasks can be supported with the current `#define` settings? _____

Which of the eight available free-running output compare timers is used for the μC/OS-II clock tick? _____

How many kernel clock ticks per second are there? _____

Exercise #1: Creating MicroC tasks

The existing program creates a single task that toggles CSM LED 1 once per two seconds. Now modify the program so that it has 3 (three) tasks and implements the following features:

(a) Task 1: modify `myTask1()` to toggle CSM LED 1 once every 500 milliseconds.
(b) Task 2: create a task that toggles the state of CSM LED 2 once every 2 seconds.
(c) Task 3: create a task that toggles the state of **SLK** LED 1 once every 20 *clock ticks*.

Be sure to set up the stack space required for each task, and call the proper startup functions with unique priorities for each task. Use the MicroC functions such as `OSTaskCreate()`, `OSTimeDly()`, and `OSTimeDlyHMSM()`.

→ ***Demonstrate and explain the results of this exercise for the instructor.***

Exercise #2: Using a semaphore to coordinate tasks

Edit your `main()` program to define a μC/OS-II *semaphore*.

The semaphore must be used as follows:

- Rewrite your Task 3: inside the infinite loop, have the task *pend* on the semaphore. When the semaphore is received, toggle the state of SLK LED 1, *post* the semaphore, and then delay for 20 clock ticks before *pend*ing again.
- Add a new task, Task 4, with the following behavior: *pend* on the semaphore. When the semaphore is received, turn on SLK LED 2 and then delay for 5 seconds. After the 5 seconds, turn off SLK LED 2, *post* the semaphore, then delay for 5 seconds before *pend*ing again.

(Task 1 and Task 2 are unaltered from Exercise #1.)

Verify that Task 3 does not run during each 5 second interval that Task 4 has the semaphore.

Exercise #3: Use a Mailbox

To investigate another inter-process communication method, create a μC/OS-II *mailbox* and use it, instead of the semaphore, to synchronize two tasks.

- Create the mailbox.
- Remove (comment out) the code in Task 3 and Task 4 dealing with the semaphore.
- Rewrite the control in Task 3 and Task 4 so that the mailbox is used as a binary semaphore. See section 10.07 (pp. 244-245) in the text for more information.
- The other tasks should be unaltered from Exercise #2.

→ ***Demonstrate and explain the mailbox and semaphore controls for the instructor.***

Extra: can you pass some information via the mailbox pointer mechanism?

Instructor Verification Sheet
Lab #8 Fall 2005

Student Name: _____

	Instructor Signature	Date
Ex. #1 Three μC/OS-II tasks running simultaneously		
Ex. #3 Semaphore and mailbox operating properly		

Lab Report

The lab report is to be written up in the Memo format. Be sure to put the *lab number* in the Memo header along with your name and date. For each exercise, answer the given questions and demonstrate your understanding of the exercise. Include **commented** file excerpts and this instructor verification sheet to get credit for the lab.

→ This lab report is due by 5PM next week on Tuesday, November 15.