More task management in uC/OS-II
Mutual Exclusion

1 Introduction

This lab is the third lab concerning the real-time kernel uC/OS-II and its use on the `LPC_2378_STK` development board. Once again, it looks at the problems that can arise when tasks share resources. This time the problems are tackled by using semaphores.

2 In the lab

1. Download the file worspace.zip into a suitable directory either on a pen drive or in your University workspace. I suggest you call the directory `EN572/labs/lab07`. Unzip `workspace.zip`.
2. Start up EWARM and load the workspace `workspace/workspace.eww`.
3. Connect a `LPC-2378-STK` board to a USB port on your computer.
4. Make sure that you understand the solution to `lab06` – it's in the project `lab06S`. Download and debug this project. Test the program. Make sure that you understand how the `appTaskPot` task has been added to the project. In particular, pay attention to the declaration of :
   - the priority `APP_TASK_POT_PRIO`
   - the stack size `APP_TASK_POT_STK_SIZE` and the stack `appTaskPotStk`
   - the function prototype `appTaskPot`
   - the function `appTaskPot`

   In addition, you should notice:

   - the use of `lcdDrawRectangle` and `lcdDrawFilledRectangle`;
   - how the potentiometer values are converted into y values for rectangle drawing functions so that adjustments in the potentiometer control are displayed in a natural way.
5. When you understand the code for `lab06S`, you should move on to `lab07a`.

6. Download and debug lab07a. Run the program and observe its behaviour. Study the code. In particular, you should pay attention to:

   - the declaration of the semaphore
   - how the semaphore is created
   - the use of OSSemPend() to acquire the semaphore
   - the use of OSSemPost() to release the semaphore

   Compare this program with lab06a. What do you notice? What do you think is causing the behaviour that you see?

7. Clean the project lab06S. Copy the lab06S directory to a new directory: lab07b. Delete the Flash, and settings directories, and the file lab06S.dep, from lab07b. Rename lab06S.ewp to lab07b.ewp and lab06S.ewd to lab07b.ewd. Add the new lab07b project to your existing workspace.

8. Now download and debug the lab07b project. Run the program. Make sure that it behaves just like the original lab06S project. The remaining exercises use lab07b.

9. Now modify your program so that each of the led tasks reports its status using the LCD. The task should report whether or not it is flashing and its current flashing delay, e.g.

   ```
   (LINK) F:ON  D:3300
   (CNCT) F:OFF D:4100
   ```

10. Do you observe any interference between the led tasks and the potentiometer task? Why does this occur? Remember the LCD is now a shared resource. Use a semaphore to provide mutually exclusive access to the LCD, so that the interference is prevented. Comment on the effectiveness of your solution.

11. Make sure that you understand the assignment specification fully. Now is the time to ask your tutor to clarify any aspects of the specification that you are unsure about.