

**Sistemas embebidos**  
**2012 - Segundo Cuatrimestre**  
**Trabajo Práctico N° 1 – RTOS**

**Hardware a utilizar: LPC1769 + Baseboard**

**Firmware a utilizar: FreeRTOS + CMSIS**

- 1) Crear una tarea periódica, de nombre "Tarea 1", que cada 100ms envíe por la UART 3 el mensaje constante "Tarea 1 - Enviando información por la UART 3".
- 2) Habilitar la interrupción de recepción de la UART 3. Cuando se ejecuta la interrupción liberar un semáforo para que deje de estar bloqueada la "Tarea 2". Utilizar funciones del RTOS que sean "ISR safe". Forzar la ejecución del "scheduler" al finalizar la ISR.
  - a. ¿Por qué motivo se deben utilizar funciones del RTOS que son "ISR safe"?
  - b. ¿Qué operaciones del RTOS no son permitas dentro de una interrupción?
  - c. Como se fuerza la ejecución del "scheduler" inmediatamente después de finalizar la interrupción.
  - d. Si no se sincroniza la ejecución del "scheduler" inmediatamente después de finalizar la interrupción, ¿cuándo se ejecuta el mismo?
- 3) Crear una tarea asincrónica, de nombre "Tarea 2", que se ejecute cuando **(2)** libera el semáforo. Esta tarea debe hacer un eco del carácter recibido enviando el siguiente mensaje por la UART 3: "Tarea 2 – Carácter recibido: c" (donde c es el carácter recibido).
- 4) Al iniciar el sistema, configurar el módulo RTC del microcontrolador LPC1769. Establecer como valor inicial "0" horas, "0" minutos y "0" segundos.
- 5) Habilitar la interrupción externa para un pulsador disponible en el "baseboard". Cuando se ejecuta la interrupción enviar a una cola de tamaño "n" (donde n es un "define" de la aplicación) la información de la hora, los minutos y los segundos en los que se presiono el pulsador. Utilizar funciones "ISR safe". Forzar la ejecución del "scheduler" al finalizar la ISR.
- 6) Crear una tarea asincrónica, de nombre "Tarea 3", que se ejecuta cuando la cola escrita por **(5)** tiene información disponible. La tarea debe quitar los datos de la cola y enviar el siguiente mensaje por la UART 3: "Tarea 3 – datos" (donde *datos* es la información que estaba disponible en la cola).
- 7) Para enviar la información por la UART 3, crear un procedimiento con el siguiente formato":

*void UART3\_Enviar (uint8\_t\* buffer, uint32\_t size);*

La función *UART3\_Enviar* debe utilizar un mutex para administrar el recurso compartido por las diferentes tareas.

- 8) Justificar la elección de prioridades de las diferentes tareas. Hacer un diagrama de las tareas/interrupciones en función del tiempo, explicando que ocurre en los diferentes cambios de contexto.