

SISTEMAS EMPOTRADOS II

PRACTICA 1

Objetivos:

Medida de tiempos de ejecución. Análisis y diseño basado en asignación de prioridades primero al más urgente (DM). Estructuras básicas de programación tiempo real concurrente.

Material:

Computador PC, Code Composer Studio con SYS-BIOS, tarjeta TM4C123G LaunchPad.

Apartado 1 (proyecto PR11)

El módulo **computos** ofrece la rutina **CS()** cuyo tiempo de cómputo es proporcional a su parámetro de llamada. Medir el tiempo de cómputo de CS(10) mediante dos técnicas:

1. Instruméntese el código para que el comienzo de la actividad suponga la puesta a 1 de un pin del GPIO del procesador y el final de la actividad la puesta a cero del mismo pin. Mídase el tiempo de cómputo con un osciloscopio.
2. Instruméntese el código con primitivas de la herramienta de análisis *Duration* de CCS. Mídase el tiempo de cómputo con el entorno de desarrollo.

Apartado 2 (proyecto PR12)

El programa concurrente **indep** consta de tres tareas periódicas T1, T2 y T3 activadas por retrasos que evolucionan de forma independiente (sin comunicaciones). Los períodos respectivos son: 100 ms, 150 ms y 350 ms. El plazo de respuesta es igual al periodo.

1. Realizar, con los tiempos de cómputo obtenidos, el análisis temporal (demostrando que se van a cumplir, o no, los plazos correspondientes) y asignar prioridades.
2. Verificar el cumplimiento de los plazos monitorizando la ejecución del programa mediante la herramienta *Execution Graph* de CCS.

3. Invertir las prioridades de las tareas y comprobar sobre una traza del programa los problemas que aparecen.

Apartado 3 (proyecto PR13)

El programa concurrente **comun** consta de tres tareas periódicas T1, T2 y T3 activadas por interrupción que comunican entre sí mediante dos servidores, programados en el módulo **servidores** (véanse los respectivos códigos fuente).

1. Realizar el análisis temporal (demostrando que se van a cumplir, o no, los plazos correspondientes que son iguales a los periodos) y asignar prioridades. Calcular los tiempos de finalización de cada tarea.
2. Verificar el cumplimiento de los plazos y los tiempos de finalización sobre una traza del programa, utilizando el protocolo de herencia de prioridad (*GateMutexPri*). Dibujar la traza sobre el papel y señalar las inversiones de prioridad que se producen y su duración.
3. Cambiar a unos mutex que no tengan el protocolo de herencia de prioridad (*GateMutex*). Comprobar sobre la traza los tiempos de finalización y las inversiones de prioridad.