

Tema 1. Introducción

Tema 2. Recursos de acceso al hardware

Tema 3. Interrupciones

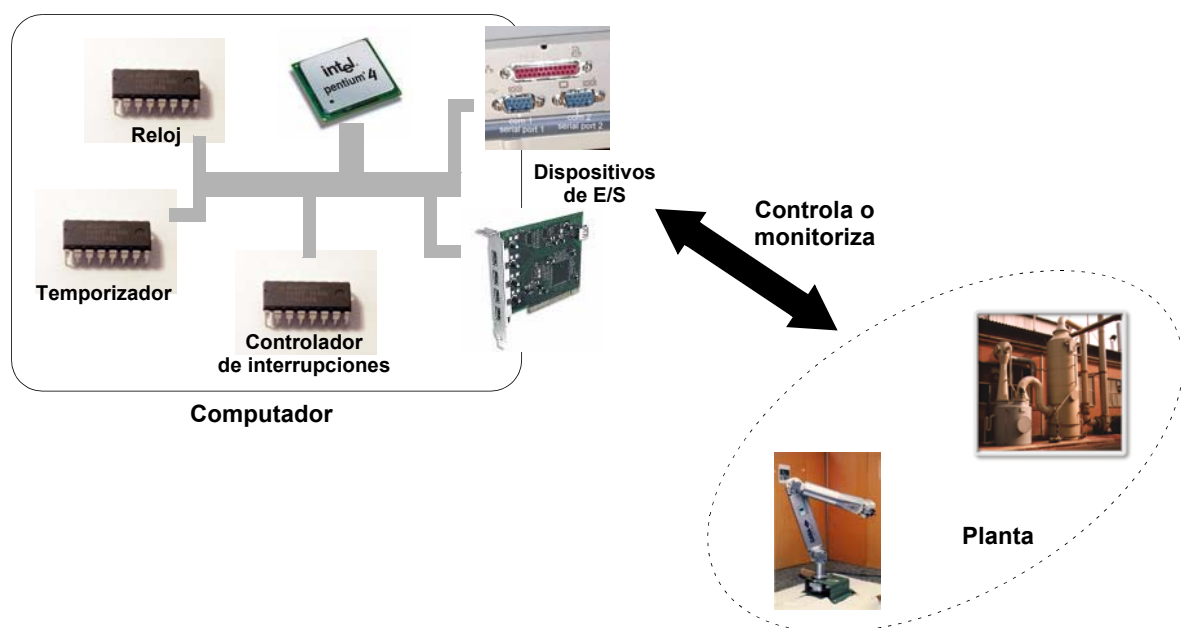
Tema 4. Puertas básicas de entrada/salida (I)

Tema 5. Recursos de temporización de bajo nivel

Tema 6. Multitarea en Ada

Tema 7. Puertas básicas de entrada/salida (II)

Modelo de sistema objeto de estudio de la asignatura



Un sistema de tiempo es una combinación de computador, dispositivos de E/S y software de propósito especial en la que:

- existe una fuerte interacción con el entorno
- el entorno cambia con el tiempo
- el sistema debe controlar y/o reaccionar frente a diferentes aspectos del entorno

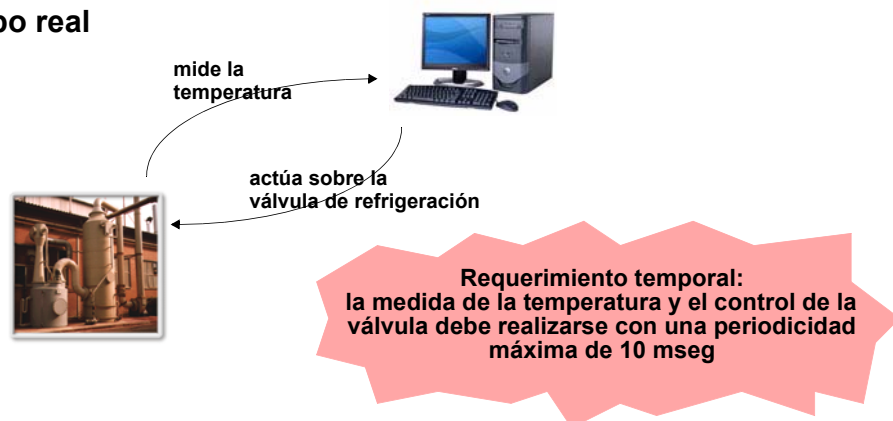
Como consecuencia:

- se imponen requerimientos temporales al software
- el software es de naturaleza concurrente

Sistemas de Tiempo Real (cont.)

“En aplicaciones de tiempo real, la validez de una computación depende, no sólo del resultado obtenido, sino también del instante en el que se obtiene”

Ejemplo de sistema de tiempo real



Necesidad de concurrencia en sistemas de tiempo real

Lo habitual es que un sistema de tiempo real tenga que realizar varias actividades de forma concurrente

Por ejemplo: tareas en el controlador de un coche:

- medir la velocidad cada 200 ms
- controlar la inyección cada 40 ms
- control del ABS cada 80 ms

Para el programador resulta más sencillo trabajar sobre cada una de las tareas de forma independiente

- el sistema operativo o el lenguaje de programación proporcionará el soporte para que las tareas se ejecuten de forma concurrente

Estrategias de aproximación al problema

Programación en lenguaje ensamblador

- acceso directo a los recursos de temporización de bajo nivel (temporizadores, relojes, etc)
- difícil implementar la concurrencia
- apropiado para aplicaciones sencillas
- apropiado para microcontroladores pequeños (y baratos)

Programación con un lenguaje de alto nivel

- acceso directo a recursos de temporización de bajo nivel
- difícil implementar la concurrencia
- apropiado para aplicaciones de complejidad media
- apropiado para microcontroladores de tamaño medio

Estrategias de aproximación al problema (cont.)

Programación con un lenguaje de alto nivel que soporta la concurrencia (Ada) (o con C y un sistema operativo POSIX)

- el acceso a los recursos de temporización de bajo nivel se hace de forma transparente para el programador
- fácil implementar la concurrencia
- apropiado para aplicaciones de complejidad media o alta
- apropiado para controladores relativamente potentes

