

Programación de Sistemas Embebidos

Robles Vázquez Eduardo

INTERRUPCIONES

Un procesador embebido normalmente tiene que:

* Medir tiempos
* Generar eventos basados en tiempo
* Responder en tiempo real a eventos que ocurren en tiempos impredecibles

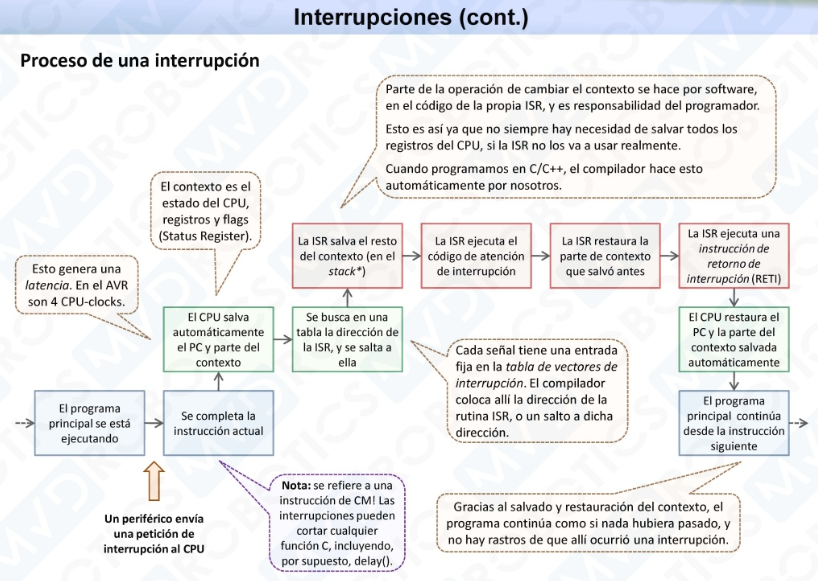
Un sistema embebido tiene, generalmente, que atender varias “tareas”, algunas de ellas periódicas, otras disparadas por eventos. Las interrupciones y los timers son herramientas clave para lograr un manejo efectivo del tiempo y los eventos en un sistema embebido.

En la programación estándar, en una PC, las interrupciones son un recurso del hardware que se usa a nivel del S.O., el kernel, los drivers, lo que llamamos “bajo nivel”. En un µC la programación es siempre de “bajo nivel”. Programamos directamente sobre el CPU, y no sobre la “máquina virtual” del S.O. Asimismo, en la programación normal de la PC, bajo el S. O., existen los threads. El equivalente en el uC son las interrupciones.

Las interrupciones son un recurso esencial de los sistemas embebidos. Básicamente, la interrupción es un mecanismo mediante el cual el CPU puede, ante cierto evento, suspender lo que está haciendo en ese momento y pasar a atender una rutina de alta prioridad. Una vez finalizada ésta, el CPU vuelve a su actividad anterior.

**Ventajas de la interrupción**

* Ahorra CPU. Evita perder tiempo esperando que ocurran cosas.
* Permite al procesador hacer más cosas “simultáneamente”.
* No se pierden eventos.
* Permite tratar algunos eventos con más prioridad que otros.
* Permite responder a los eventos más rápido, si latencia.
* Permite poner el uC en sleep y aun así responder a ciertos eventos.
* Simplifica el diseño de la aplicación, aunque hay que cambiar totalmente el modo de pensar.



**Como definir una ISR**

Para utilizar una interrupción necesitamos hacer 3 cosas:

1. Habilitar las interrupciones globales.
2. Habilitar la máscara de la interrupción especifica.
3. Escribir una función ISR y asociarla (“Linearla”) a esa interrupción.

**Bibliografía:**

* Farias, J. (s.f.). CY3210-PSoCEval1 Modo De bajo Consumo, SLEEP Timer e Interrupciones GPIO en Ensamblador. Recuperado 5 febrero, 2020, de <https://www.academia.edu/34417115/CY3210-PSoCEval1_Modo_De_bajo_Consumo_SLEEP_Timer_e_Interrupciones_GPIO_en_Ensamblador>
* CYPRESS. (s.f.). [PSoC 4 GPIO Interrupt]. Recuperado 4 febrero, 2020, de <https://www.cypress.com/file/227306/download>