

PRÁCTICA P2:

TEMPORIZADORES: DISEÑO DE UN CRONÓMETRO

Descripción

En esta práctica vamos a utilizar el temporizador T7 para controlar determinados intervalos de tiempo y así diseñar un cronómetro simple cuyo funcionamiento mostraremos visualmente a través de los leds: uno de los leds conmutará cada décima de segundo, y el otro cada segundo.

El temporizador T7 lo controlaremos de dos formas: primero, en el apartado P2_A, mediante encuesta, y después, en el apartado P2_B, mediante interrupción.

Objetivo

Aprender a controlar y medir tiempos con el fin de saber, por ejemplo, cuánto tiempo se necesita para realizar determinadas funciones, o cuánto tiempo ha transcurrido entre dos eventos determinados, o cuánto tiempo pasa la CPU desocupada etc.

Antes de comenzar con la práctica: Es necesario analizar la estructura del temporizador T7 para entender su funcionamiento. Además de eso, hay que entender el significado de los diferentes bits del registro de control para inicializarlos convenientemente. Por último, para controlar bien los tiempos es fundamental comprender el funcionamiento del sistema de reloj. Vamos a utilizar el fichero `oscillator.c` que realiza los cambios necesarios para que el microcontrolador funcione a una frecuencia de 80 MHz.

Aquí tienes algunas preguntas sobre las que reflexionar antes de comenzar con la práctica:

1. El reloj del microcontrolador (F_{osc}) es de 80 MHz pero para ejecutar una instrucción hacen falta 2 ciclos de reloj, para lo que el hardware genera una nueva señal de reloj, de frecuencia F_{CY} . ¿Cuál es la frecuencia de esa señal? ¿Cuánto tiempo se necesita para ejecutar una instrucción?
2. ¿Qué valor hay que cargar en el registro PR7 del temporizador T7 para medir 1 ms? (Dicho de otro modo: para que genere una petición de interrupción cada milisegundo).
3. ¿Y para 10 ms? ¿Genera eso algún problema? En caso afirmativo, ¿cómo se puede resolver?
4. Analiza cómo influye el valor del prescaler en la precisión de la medida de tiempo que se quiera realizar.

Realización:

1. Crea una nueva carpeta, de nombre P2_A_v1, para el primer proyecto de esta práctica y copia en ella los ficheros que hemos dejado en eGela.
2. Conecta la placa Explorer 16 y el emulador REAL ICE con el PC.
3. Ejecuta la aplicación MPLAB X y crea un nuevo proyecto con el mismo nombre que la carpeta, P2_A_v1.
4. Añade al proyecto los ficheros `.c` → 'Source Files'; y los `.h` → 'Header Files'.
5. Edita los ficheros y completa la información que falta para el funcionamiento del programa sea el descrito anteriormente:
 - Inicializa T7 y escribe una función para detectar por encuesta el paso de 10 ms.

- Tomando como referencia ese tiempo de 10 milisegundos, escribe otra función que cuente milisegundos (una variable), y al llegar a 100 (una décima de segundo) conmute un led (encender/apagar).
 - Utiliza otra variable para contar el paso de cada segundo y conmutar otro led.
6. Compila comprobando que no hay errores y ejecuta el programa en modo 'Debug'.
 7. Verifica el funcionamiento del programa paso a paso.
 8. Añade los comentarios pertinentes.

P2_B_v1:

En esta nueva versión el comportamiento del programa debe ser el mismo que el de la versión anterior, pero ahora el control del temporizador T7 debes hacerlo por interrupción. Además añadiremos varios pulsadores que tengan efecto sobre el cronómetro y sobre la ejecución del programa. Es decisión vuestra cómo controlar esos pulsadores, por encuesta o por interrupción, pero conviene pensar cómo hacerlo para conseguir un buen rendimiento y eficiencia del programa.

- Controlar mediante un pulsador la puesta en marcha del cronómetro, de modo que hasta que no se pulse el pulsador elegido no se visualice en los leds el paso del tiempo.
- Controlar con un nuevo pulsador el parar o continuar con el crono: si se pulsa el pulsador con el cronómetro en marcha éste se detiene, y si por el contrario está parado, se pone en marcha partiendo del instante en el que se había detenido. Opcionalmente puede añadirse un tercer pulsador para inicializar el cronómetro, de modo que al pulsarlo se empiece de nuevo desde 0.

P2_B_v2:

Define dos nuevas funciones en el fichero **timers.c** (nos van a ser de utilidad en la siguiente práctica). Van a ser las funciones `Delay_ms` y `Delay_us`, cada una de las cuales tiene un parámetro que indica respectivamente el número de milisegundos o microsegundos que deben transcurrir hasta que la función acabe. Utiliza el temporizador T9 para esas funciones. Cada una de ellas debe inicializar el temporizador según el tiempo indicado por parámetro, ponerlo en marcha, esperar por encuesta a que transcurra el tiempo para el que ha sido programado, y pararlo de nuevo.

La comprobación del funcionamiento de estas dos funciones la vamos a hacer en clase mediante una utilidad que ofrece MPLAB y que cuenta el número de ciclos que transcurren entre dos puntos de ruptura (*Breakpoint*).

Qué debéis subir a eGela. En la tarea correspondiente de eGela indicaremos las versiones a entregar, siempre con su correspondiente informe.