

Práctica 2. Entrada/salida mediante interrupciones

1. *Objetivos de la práctica*

En esta práctica continuaremos nuestro estudio del sistema de entrada/salida del procesador ARM7TDMI, pero esta vez gestionando interrupciones. El principal objetivo de la práctica es entender y saber manejar el sistema de E/S por interrupciones vectorizadas.

Tras comprobar las limitaciones de la entrada/salida mediante espera de respuesta, en esta práctica analizaremos el sistema de E/S con interrupciones vectorizadas que nos proporciona el controlador de interrupciones del chip S3C44BOX. Además, añadiremos un temporizador.

2. *Lecturas previas obligatorias*

Es imprescindible leer los capítulos 1 a 7 del documento titulado “Sistema de memoria y de entrada/salida en la placa S3CEV40” ([TPGb]) publicado a través del Campus Virtual..

3. *E/S por interrupciones vectorizadas*

Ejecutando el código de la práctica anterior habremos podido observar que en ocasiones tiene un comportamiento poco preciso, debido a que la espera software introducida hace que no siempre atendamos a los dispositivos.

En esta práctica modificaremos el código de la práctica anterior para que funcione por interrupciones vectorizadas por la línea IRQ del procesador. La idea es la siguiente:

- Haremos que el movimiento del led alrededor del display de 8 segmentos esté gobernado por una interrupción periódica (de periodo 2s) generada por el TIMER0.
- Haremos que los botones sean atendidos bajo interrupción.
- La función loop no tendrá por tanto nada que hacer.

4. *Paso a interrupciones vectorizadas y uso de temporizadores*

Para que funcione el proyecto añadimos al proyecto cuatro nuevos ficheros:

- **intcontroller.h** e **intcontroller.c**: interfaz e implementación de funciones del módulo que maneja el controlador de interrupciones. El alumno deberá completar la implementación de todas las funciones de este módulo, siguiendo las indicaciones de los comentarios y la documentación de [TPG, um-].
- **timer.h** y **timer.c**: interfaz e implementación de funciones del módulo que maneja los temporizadores. El alumno deberá completar la implementación de todas las funciones de este módulo, siguiendo las indicaciones de los comentarios y la documentación de [TPG, um-].

Asimismo, en el fichero main.c deberemos realizar los siguientes cambios:

- Función **setup**:
 - debemos cambiar la configuración del puerto G para que los pines 6 y 7 activen las señales EINT6 y EINT7 respectivamente, utilizando el interfaz del puerto G definido en gpio.h.

- debemos configurar el TIMERO para que genere interrupciones periódicas, con un periodo de 2 segundos, utilizando el interfaz definido en timer.h.
- debemos registrar las rutinas button_ISR y timer_ISR como rutinas de tratamiento de interrupciones para el tratamiento de las señales EINT4567 y TIMERO respectivamente.
- debemos configurar el controlador de interrupciones para que active la línea IRQ en modo vectorizado, deje la línea FIQ enmascarada, configure las líneas TIMERO y EINT4567 por la línea IRQ del procesador y las deje habilitadas.
- Función **loop**: estará vacía (dejaremos sólo el return).
- Función **timer_ISR**: será declarada como una rutina de tratamiento de interrupción por IRQ, para que el compilador la genere como RTI. Es la encargada de mover el led en el display de 8 segmentos una posición. La dirección del movimiento será la que esté almacenada en la variable RL. Además, al final de la función se debe borrar el flag de interrupción utilizando el interfaz definido en intcontroller.h.
- Función **button_ISR**: al igual que la anterior será declarada como rutina de tratamiento de interrupciones por IRQ y deberá encargarse de borrar el flag de interrupción. Es la encargada de saber qué botón se ha pulsado consultando el registro EXTINTPND y realizar las tareas asociadas a la pulsación de cada uno de los botones (encender o apagar un led, cambiar la dirección de giro del led rotante y parar o arrancar el timer).

5. Configuración del temporizador

Como ayuda final, veremos qué valores podemos dar a los registros de configuración del TIMERO para que produzca interrupciones periódicas de periodo M segundos (en nuestro caso M=2). De acuerdo con la documentación del chip, la frecuencia con la que trabajan los timers depende de tres factores: la frecuencia del sistema (MCLK) que en nuestro caso es 64 MHz, el factor de pre-escalado (P) que está en el intervalo 0-255 y el factor de división (D) que puede valer 2, 4, 8, 16 o 32. Conocidos estos valores la frecuencia de funcionamiento sería:

$$F = \frac{MCLK}{(P + 1) * D}$$

Si elegimos un factor de división de 8 y uno de pre-escalado de 255 vemos que la frecuencia será:

$$F = \frac{64 * 10^6}{(255 + 1) * 8} = 31250 \text{ ciclos por segundo}$$

Para obtener una interrupción cada dos segundos deberemos contar 2*31250 de estos ciclos, es decir el contador de ciclos lo inicializaremos a 62500.

El valor del registro de comparación puede ser cualquier valor mayor que 0 y menor que 62500.