

Práctica 4

Estructura de Computadores

Modos de ejecución, gestión de excepciones y
entrada/salida mediante interrupciones
vectorizadas

Índice general

3.1. Objetivos de la práctica.....	2
3.2. E/S por interrupciones vectorizadas	2
Bibliografía	4

3.1. Objetivos de la práctica

En las prácticas 2 y 3 se ha estudiado la E/S programada por espera de respuesta utilizando los dispositivos más sencillos y la E/S con interrupciones no vectorizadas, donde hemos tenido que programar una subrutina de tratamiento de interrupción (irq_ISR) que realizaba una encuesta para saber qué dispositivo era el que había generado la interrupción .

En esta práctica trabajaremos con interrupciones vectorizadas. Para ello hay que modificar el código de la práctica 3 para que trabaje con interrupciones vectorizadas por la línea IRQ.

Además vamos a añadir un nuevo dispositivo, el teclado matricial, que también trataremos por interrupciones. Para comprender el funcionamiento del teclado debe leerse la sección 8 de [TPGb].

3.2. E/S por interrupciones vectorizadas

El funcionamiento de la práctica 4 será el mismo que el de la práctica 3 donde se le ha añadido lo siguiente:

- Cuando se pulse una tecla del teclado matricial, primero se verá dicha tecla en el display y después volverá a aparecer el led rotante.
- La velocidad de movimiento del led rotante variará en función de la tecla pulsada como se explica al final de la siguiente sección.

Lo primero que hay que hacer es adaptar el código de la practica 3 para que las interrupciones por la línea IRQ se traten en modo vectorizado. Para ello hay que:

- Copiar los archivos **gpio.c**, **leds.c**, **D8Led.c**, **intcontroller.c** y **timer.c** codificados en la práctica 3.
- Copiar el archivo **main.c** y realizar los siguientes cambios:
 - En la función **setup** cambiar la configuración del controlador de interrupciones para que la línea IRQ esté en modo vectorizado.

- Convertir las rutinas **button_ISR** y **timer_ISR** en rutinas de tratamiento de interrupción reales para el tratamiento de las señales **EINT4567** y **TIMER0** respectivamente. Para esta conversión es necesario hacer dos cosas:
 - Registrar las ISRs, es decir, añadir la dirección de la ISRs a la tabla de direcciones de las subrutinas de tratamiento de interrupción (como las funciones están escritas en C, hay que registrarlas también en C).
 - Añadir al final de las funciones el borrado del flag de interrupción.
- Copiar el archivo init.S y realizar los siguientes cambios:
 - Borrar la función **irq_ISR**, puesto que ya no vamos a configurar la línea IRQ en modo no vectorizado, y por lo tanto no se necesita hacer la encuesta para saber que dispositivo es el que había generado la interrupción.
 - Borrar el código en el que se añade la dirección de la **irq_ISR** a la tabla de direcciones de las subrutinas de tratamiento de interrupción

Una vez realizados los cambios anteriores comprobad que funciona como en la práctica 3, pero esta vez sin necesidad de hacer una encuesta para saber que dispositivo es el que ha generado la interrupción porque estamos utilizando interrupciones vectorizadas,

Cuando hayáis comprobado que la práctica 3 adaptada para que las interrupciones por la línea IRQ se traten en modo vectorizado funciona, hay que añadir el soporte para utilizar el teclado. Para ello hay que realizar los siguientes pasos:

- Añadir dos nuevos ficheros **keyboard.h** y **keyboard.c** que definen e implementan el interfaz del módulo de teclado. El alumno deberá **completar la implementación de la función de escaneo kb_scan**.
- Crear una función **keyboard_ISR** para el tratamiento de la interrupción por pulsación de tecla en el teclado matricial. Deberá declararse como ISR y registrarse adecuadamente para el tratamiento de la línea **EINT1**. El esqueleto de dicha función es el siguiente:
 - Esperar 20ms para eliminar los rebotes de presión
 - Escanear el teclado utilizando el interfaz definido en el fichero **keyboard.c**
 - Poner en el display la tecla pulsada
 - Cambiar la configuración del timer 0 para que la generación de interrupciones periódicas tenga un periodo distinto en función de la tecla pulsada:
 - ✓ Tecla 0: periodo de 2s, valor de cuenta 62500 y divisor 1/8
 - ✓ Tecla 1: periodo de 1s, valor de cuenta 31250 y divisor 1/8
 - ✓ Tecla 2: periodo de 0.5s, valor de cuenta 15625 y divisor 1/8
 - ✓ Tecla 3: periodo de 0.25s, valor de cuenta 15625 y divisor 1/4

- Esperar a que se deje de presionar la tecla leyendo el bit 1 del registro de datos del puerto G.
 - Esperar 20ms para eliminar rebotes de depresión.
 - Borrar el flag de interrupción por la línea EINT1.
- En la función **setup** añadir a la configuración del controlador de interrupciones la configuración de la línea **EINT1** por IRQ y la habilitación de esta línea.

Bibliografía

[arm] Arm architecture reference manual. Accesible en <http://www.arm.com/miscPDFs/14128.pdf>. Hay una copia en el campus virtual.

[TPGa] Christian Tenllado, Luis Piñuel, and José Ignacio Gómez. Introducción al entorno de desarrollo eclipse-arm.

[TPGb] Christian Tenllado, Luis Piñuel, and José Ignacio Gómez. Sistema de memoria y de entrada/salida en la placa s3cev40.

[um-] S3c44b0x risc microprocessor product overview. Accesible en http://www.samsung.com/global/business/semiconductor/productInfo.do?fmly_id=229&partnum=S3C44B0. Hay una copia en el campus virtual.