

Práctica 3: Creación funciones de retardo y un reloj a tiempo real mediante el uso de Timers y interrupciones

Eudald Elias , Manuel Lecha

1

1. ¿Qué se quiere hacer?

El objetivo principal de esta práctica es partiendo de la base que ya implementamos para la práctica 2, ampliar el funcionamiento de nuestro robot indagando en una nueva clase de periférico, distinta a los GPIO vistos hasta el momento, los timers (temporizadores/-contadores).

La estructura o finalidad de la práctica se dividía en 2 tareas. En primer lugar, teníamos como objetivo programar un Timer que generase una interrupción cada milésima de segundo con el fin de aumentar o disminuir el tiempo del intervalo de encendido y apagado de un led de la secuencia implementada en la práctica 2. En segundo lugar se buscaba configurar un reloj que pudiese mostrar la hora los minutos y los segundos por pantalla y que dicha hora fuese configurable, además se buscaba la opción de poder configurar una alarma que hiciese saltar una notificación cuando la alarma y la hora coincidiesen.

A continuación comentaré de manera genérica aquellas cosas que creo destacables de la implementación de cada una de las dos partes de la práctica.

Parte 1: Lo primero consistía en configurar el timer de la manera adecuada, activando los bits necesarios. Nos decantamos por usar el Timer A0, por lo que toda la configuración que comentaré recae sobre el mismo. Por otro lado dentro del Timer A0 seleccionamos el ACLK para poder representar correctamente el segundo (Activando el BIT 8 del TA0CTL), por otro lado tal y como se pedía la implementación que saltase la interrupción cada milésima de segundo creímos oportuno operar en modo up del timer, de esta manera podíamos ordenar que la interrupción se lanzase cuando el contador del timer llegase a una determinada cifra, en este caso 33, que corresponde aproximadamente a una milésima de segundo. Esto lo conseguimos accionando el BIT 4 de TA0CTL. El valor que corresponde a la milésima de segundo en ticks del timer A0 con ACLK (33) lo asignamos a TA0CCR0, valor asociado a límite en el que se llama a la interrupción en el modo up. Como último paso de configuración activamos interrupciones en TA0CCTL0 activando el bit CCIE. Una vez terminada la configuración del timer hacia falta configurar la interrupción del timer, cada vez que se accione (milésima de segundo) aumenta en uno un contador que tendremos definido como variable local. De esta manera con la ayuda de una variable que ya teníamos definida en la práctica 2 llamada tiempo (asociada al tiempo entre el encendido de un LED y el encendido de su contiguo en la secuencia), modificamos el delay para que se mantenga en bucle hasta que el contador alcance el valor del tiempo. Por último añadir que al aumentar o disminuir el tiempo de la secuencia, mediante el joystick hemos decidido hacerlo proporcional (no restando y sumando) ya que consideramos que de esta forma es más visual.

Parte 2: Para la modificación cabe destacar que hemos decidido que el botón S1 nos permita entrar en la configuración de la hora y el botón S2 en la configuración de la

alarma y el botón central del joystick para confirmar dicha configuración. Como pasos importantes de esta parte destacamos la creación de 2 métodos que nos permiten incrementar o decrementar el tiempo, que tendrán una función o otra dependiendo del estado en el que se encuentre el flujo del programa. He de añadir que para que todo funcione coherentemente hemos añadido una variable global estado que nos permite conocer dependiendo de su valor si estamos en modificación de hora, de alarma o en su defecto en ninguno de los casos. Se llamará el método incrementar tiempo cada vez que el contador de veces que la interrupción se ha activado llegue a 1000, para que aumente un segundo el tiempo, (segundos o horas dependiendo del tiempo previo), y a su vez se refresque la pantalla imprimiendo los nuevos valores de segundo minuto y hora (Todo esto en caso que la variable de estado comentada anteriormente indique que no se está modificando ni la hora ni la alarma). Luego en caso que estemos en modo modificar alarma o hora (mediante interrupción botón S1 o S2 para no parar la secuencia de leds) almacenamos otro estado que nos permite conocer que es lo que actualmente estamos modificando sean segundos, minutos o horas, en cada uno de los dos casos. Por último para confirmar la configuración como he comentado con anterioridad necesitaremos accionar el joystick central que nos devolverá al estado que indica que nada se está modificando. Por último, hemos decidido imprimir un mensaje por pantalla cuando el valor de la hora y el de la alarma coincidan que también desaparecerá con el joystick central.

2. ¿Qué recursos del Microcontrolador, Placa de Experimentación y Robot se usan?

Lista de recursos:

- Joystick Derecha P4.5
- Joystick Izquierda P4.7
- Joystick Centro P4.1
- Joystick Arriba P5.4
- Joystick Abajo P5.5
- Pulsador S1 P5.1
- Pulsador S2 P3.5
- LED R P2.6
- LED G P2.4
- LED B P5.6
- 8 LEDS de la Placa de Interficie
- Timer A0 ACLK modo UP

3. ¿Cómo se han configurado los diferentes recursos?

Partimos de la base que el código de esta práctica reutiliza el de la anterior, y que las configuración de los GPIO usados fue comentada en la práctica anterior. Por tanto en este apartado solo comentaré la configuración del Timer, que en parte ya se ha explicado en el apartado anterior. Usamos el Timer A0 con el ACLK en el modo UP. Las instrucciones que nos permiten inicializar el método son:

```
TA0CTL = BIT8 + BIT4 + BIT1;  
TA0CCTL0 = CCIE;  
TA0CCR0 = TICKSMILI;
```

4. ¿Cómo y para que funciones se usan?

Usamos el timer para crear un delay que tenga como referencia un valor temporal real, para el retraso del encendido en la secuencia, y por otro lado para poder contar segundos y generar un reloj.

5. Problemas que han surgido y solución implementada

Ningún inconveniente en la práctica a niveles generales. Lo que quizás más tiempo tuvimos que invertir fue en la primera sesión configurar el timer de la manera idónea, haciendo valer del manual.

6. Conclusiones

En esta práctica hemos aprendido a usar y configurar un Timer y hemos profundizado los conocimientos que ya teníamos en las interrupciones. Tanto la propuesta de la práctica como resultado obtenido consideramos ha sido satisfactorio.