Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación

UNIVERISDAD POLITÉCNICA DE MADRID

INGENIERÍA DE SISTEMAS ELECTRÓNICOS

WatchDog y almacenamiento permanente de información en la memoria FLASH

Departamento de Ingeniería Telemática y Electrónica

Introducción

En esta práctica el alumno añadirá al servidor web compacto, desarrollado en las prácticas 1 y 2, el soporte para controlar el Watchdog como sistema de supervisión de su correcto funcionamiento, así como la posibilidad de que el programa almacene información de forma permanente en la memoria Flash interna.

La duración de esta práctica es de 2 semanas y a su finalización deberá justificar a su profesor que ha resuelto cada uno de los apartados propuestos.

Material necesario

Para la realización de este sistema es necesario haber concluido en su totalidad la práctica 2. También será necesario realizar una lectura de las páginas relativas al Watchdog Timer y a la Memoria Flash del manual de usuario del microcontrolador. Como código de ejemplo y programación también pueden utilizarse los proporcionados por NXP y los ejemplos disponibles en la Web. A continuación, se referencian todos los elementos necesarios:

- Servidor Web compacto (práctica 2) con los componentes de red v7.10 y toda la gestión de la hora (RTC+SNTP).
- Capítulo 28 “LPC176x/5x Watchdog Timer (WDT)” del documento “UM10360 LPC1768 User Manual Rev 4.1 2016-12-19”.
- Capítulo 32 “LPC176x/5x Flash memory interface and programming” del documento “UM10360 LPC1768 User Manual Rev 4.1 2016-12-19”.
- Ejemplos de utilización del Watchdog y de la Flash desde el programa (In-Application (IAP) programming), accesibles en https://community.nxp.com/external-link.jspa?url=https%3A%2F%2Fdrive.google.com%2Fopen%3Fid%3D1qdtXqvLqQoIKSMrtabgRIH5fCs94qsPv

APARTADO 1

En este primer apartado se pretende que el estudiante se familiarice con el manejo básico del Watchdog. Como ya se explicó, el Watchdog es un Timer especial, que permite detectar, en determinadas circunstancias, un mal funcionamiento de la aplicación. El primer paso es el análisis detallado del funcionamiento del Watchdog, leyendo con atención el capítulo 28 del manual de usuario del LPC1768. Como ya sabe, el Watchdog puede configurarse para generar una interrupción o un reset del sistema si no se "alimenta" en el tiempo predeterminado en su configuración.

En este apartado se configurará el Watchdog para generar una interrupción si su estado de cuenta alcanza el 0.

Las especificaciones de la aplicación que debe desarrollar son las siguientes:

- Se creará un nuevo proyecto de Keil desde cero, sin sistema operativo.
- Se configurará el Watchdog para que genere una interrupción si no es "alimentado" en un tiempo de 5 segundos.
- La aplicación deberá iniciarse encendiendo el LED1 durante aproximadamente cuatro segundos. Posteriormente deberá pasar a un estado en el que el LED1 y el LED2 se enciendan alternativamente cada medio segundo, aproximadamente.
- Se deberá "alimentar" periódicamente el Watchdog en el programa principal para que, en circunstancias normales, no se produzca la interrupción del Watchdog.
- Se configurará la rutina de atención a la interrupción del Watchdog para que, si se produce, se cambie el estado del LED3.

Ejecute esta aplicación y compruebe que no se produce nunca la interrupción del Watchdog.

A continuación, se incluirá el siguiente código, correspondiente a la rutina de atención a interrupciones externas, de forma que, cuando se produzca una pulsación del gesto DOWN del joystick, se conmute

el estado del LED4. Este comportamiento será así para las 10 primeras pulsaciones, mientras que las siguientes se ignorarán (tenga en cuenta que no están eliminados los rebotes del joystick en este punto, ni hace falta eliminarlos). Deberá configurar correctamente el resto de los recursos del sistema para el uso del joystick.

```
void EINT3_IRQHandler (void) {
   static uint32_t cuenta;
   if (cuenta++ < 10) {
      estado = ~estado;
      GPIO_PinWrite(PORT_LEDS,PIN_LED4, estado);
      LPC_GPIOINT->IO0IntClr |= 1<<PIN_DOWN;
   }
```

Ejecute la aplicación y observe cuál es el nuevo comportamiento. Identifique si se produce en algún caso la interrupción del Watchdog y explique cuál es la causa. Corrija el problema detectado y vuelva a ejecutar la aplicación para determinar si el funcionamiento es correcto.

Identifique para su subida a Moodle ambos proyectos, con la aplicación que presenta el problema y con él corregido.

APARTADO 2

En este apartado se modificarán las aplicaciones creadas en el apartado anterior (con y sin problema) para que, en vez de generar una interrupción, el Watchdog genere un reset cuando su cuenta llegue a 0 si el sistema presenta un problema.

Además, en el arranque del sistema, se deberá identificar cuál ha sido la última la causa de reset. Deberá diferenciarse si se ha producido un reset normal (por conexión de la alimentación o pulsación del botón de reset) o provocado por el Watchdog. Deberá indicarse, durante aproximadamente los 3 primeros segundos de funcionamiento de la aplicación, dicha causa, encendiendo el color

verde del LED RGB si el reset ha sido por alimentación y el color rojo del LED RGB si ha sido provocado por el Watchdog.

Identifique para su subida a Moodle ambos proyectos, con la aplicación que presenta el problema y con él corregido.

APARTADO 3

En este apartado debe integrar la configuración del Watchdog en la aplicación desarrollada en la Práctica 2. Deberá configurar el mismo para que produzca un reset del sistema si no se alimenta en 5 segundos, e introducir en el código de la aplicación las sentencias necesarias para que, en funcionamiento normal, el Watchdog no llegue a su estado de cuenta 0.

Describa claramente en un documento cuáles han sido los cambios que ha introducido en la aplicación desarrollada en la práctica 2 y prepare un nuevo proyecto para su subida a Moodle.

APARTADO 4

La gestión de la memoria Flash que se encuentra en el microcontrolador LPC1768 puede realizarse en dos modalidades, programación off-chip (desde fuera del chip) y programación desde el programa (*in-application programming*, IAP). En esta práctica nos vamos a centrar en este último tipo.

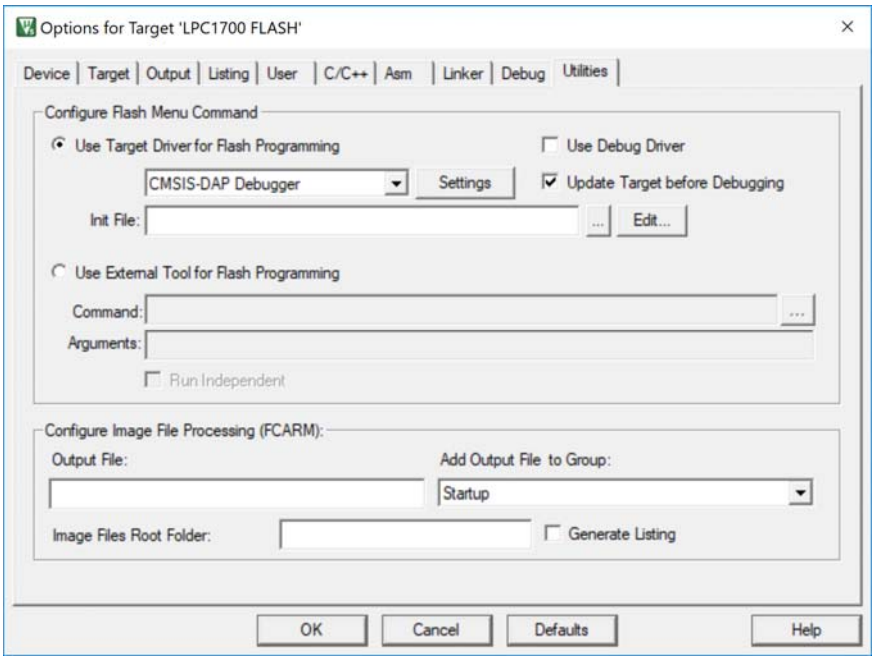
La programación de la Flash siempre se realiza en sectores que son bloques de memoria contigua. El concepto de sector y sus implicaciones se deben analizar leyendo la parte correspondiente del capítulo 32.5.

La gestión de los sectores de memoria en modo IAP se realiza mediante una serie de comandos tal y como se explica con detalle en el apartado 32.8 IAP *commands*.

Antes de realizar este apartado debe leer detalladamente el capítulo 32 del manual de usuario y comprobar el funcionamiento del ejemplo IAP que se encuentra en lpc175x_6x_cmsis_driver_library.

Seguidamente debe analizar cada una de las funciones que componen el ejemplo e interpretar el procedimiento que se emplea para escribir/leer en la memoria flash.

Para poder ejecutar este proyecto en la placa debe modificar el emulador a emplear tanto en la pestaña de debug (como hasta ahora) como en la pestaña “utilities”.

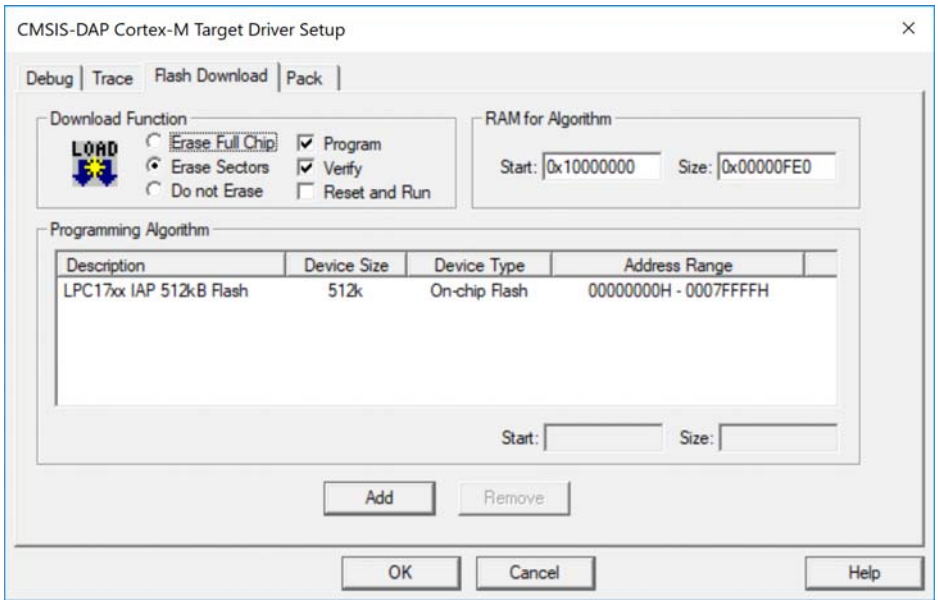


También debe tener en cuenta las opciones que se emplean cuando se realice la carga del programa mediante el botón “Load” ya que por de defecto, y tal y como muestra la figura, se borran los sectores de la Flash.

Una vez que haya cargado el programa en la placa debe ejecutarlo paso a paso para comprender su funcionamiento.

Finalmente tenga en cuenta que tras este ejercicio debe tener claro,

- direcciones de memoria en las que se lee/escribe, sector empleado, tamaño del sector, etc.
- Funciones auxiliares para borrar la memoria Flash.
- Relación de comandos disponibles en la programación IAP.
- Direcciones en las que es necesario escribir los diferentes comandos para gestionar la memoria Flash.



APARTADO 5

En este apartado se pretende modificar la aplicación de ejemplo para desarrollar varias aplicaciones parciales que permitan controlar el contenido de la memoria Flash de forma aislada. Debe desarrollar las siguientes aplicaciones:

- Borrar por completo el sector que emplea el ejemplo
- Escribir las primeras 16 posiciones del sector con los datos que declare en un *array*. Visualice si el resultado es correcto en una ventana de memoria del depurador.
- Leer las primeras 16 posiciones del sector, guárdelas en otro *array*. Compruebe los valores leídos visualizándolos en una ventana de *watch* del depurador.
- Leer las primeras 16 posiciones del sector, modificar una de ellas y volver a almacenarlas. Basicamente se trata de tener una aplicación que permita modificar sólo uno de los valores almacenados en la Flash.

Almacene cada uno de estos proyectos en un directorio diferentes y genere un ZIP que será subido en el enlace correspondiente de Moodle.

APARTADO 6

Añada al proyecto obtenido en la Práctica 2 (ojo, sin incluir el WatchDog que ha realizado en apartados anteriores), el soporte que permita guardar información en la memoria Flash del microcontrolador. Lo habitual es que esta memoria se emplee para guardar información de la configuración inicial del sistema o la situación en la que se encuentra el sistema antes de apagarse.

En esta práctica debe almacenar la siguiente información en la memoria Flash.

- Dirección MAC y dirección IP del dispositivo. Para obtener estos datos debe identificar las funciones del MDK que permiten obtenerlas a partir de la configuración realizada en Keil. Para esta información debe emplear los primeros 10 bytes del sector. Esta información se debe almacenar cada vez que se arranque el sistema.
- Situación en la que se encuentran los LEDs de la aplicación. Se considera que los LEDs pueden estar en modo manual o en el modo automático (rotatorio). Si estaban en modo automático, se debe arrancar en este mismo modo (no es necesario empezar por el mismo LED que estaba encendido cuando se apagó el sistema); si por el contrario estaban en modo manual, se deben encender los LEDs que estaban encendidos cuando se hizo un *reset* (o se apagó la placa).

  Al arrancar se debe leer el estado anterior de los LEDs y reprogramarlos para que continúen en el mismo estado en el que estaban. Emplee la codificación que desee para almacenar el estado de los LEDs en la FLASH. Debe emplear el byte 11 del sector para almacenar esta información.

APARTADO 7

En este último apartado se trata de integrar el código que gestiona la memoria Flash con la información proporcionada por el ADC que se empleó en la práctica 1 (apartado 5) y que se encuentra conectado a uno de los potenciómetros de la placa de aplicaciones.

Cada vez que se produzca una lectura del ADC, el valor obtenido se debe convertir a un valor entre 0 y 255. Dicho valor se debe comparar con el valor almacenado en la posición 12 del sector de la Flash.

En el caso de que el valor convertido sea superior al almacenado en la Flash se debe encender el LED RGB con el color azul, en caso contrario se debe apagar el LED RGB.

Para poder realizar pruebas debe guardar un valor en la posición 12 del sector de la memoria flash empleando las aplicaciones desarrolladas en el apartado 5.

Una vez finalizado el proyecto debe almacenarlo y subirlo a través del enlace de Moodle correspondiente.

APARTADO 8 (opcional)

Debe completar la aplicación del apartado anterior con algunas (o todas) de las siguientes funcionalidades:

- crear una nueva página Web que permita modificar el valor del umbral que hace que se encienda el LED RGB.
- mostrar en la página web identificador del chip (PartiID) de la memoria Flash así como el número de serie (Serial Number)
- mostrar en la página web toda la información almacenada en la memoria flash
- integrar desde la página creada la posibilidad de borrar toda la memoria flash.
- integrar desde la página creada la posibilidad de modificar el contenido de cualquier posición de la memoria Flash.