Escuela Técnica Superior de Ingeniería Informática

Ingeniería Informática. Ingeniería de Computadores.

MEMORIA PRÁCTICA I

Asignatura: Sistemas Empotrados y de Tiempo Real I
Profesor: Gabriel Jiménez Moreno

Alumno: Álvaro José Gullón Vega

Índice

1. Objetivos

En esta primera práctica distinguimos dos objetivos:

1.1. Académico

Como objetivo académico, la práctica nos introduce al sector de la programación de microcontroladores a través de la instalación y posterior uso de el entorno de desarrollo STM32CubeIDE, que se puede complementar con el simulador QEMU para realizar proyectos en los que no haya posibilidad de adquirir el hardware a usar.

1.2. Práctico

En cuanto al objetivo práctico, la práctica se fundamenta en dar una explicación sobre el entorno para ir familiarizándonos con él junto a unas instrucciones detalladas para la correcta instalación de este junto al simulador a usar, incluyendo nociones básicas sobre configuraciones del proyecto y cómo debemos ejecutarlo.

2. Introducción

En esta práctica por motivos de calendario, se ha hecho de forma no presencial sin ninguna explicación de profesor (quitando el documento proporcionado). Como hemos mencionado anteriormente, esta práctica trata sobre la introducción al sistema de desarrollo STM32CubeIDE, que será el entorno de desarrollo que usaremos durante todo el curso, junto a la instalación de un simulador llamado QEMU para la implementación de aplicaciones cuando no tenemos recursos hardware disponibles.

La placa con la que vamos a probar el sistema es la STM32 Cortex M0, pese a que en el resto de las prácticas vamos a usar una STM32 Cortex M4 con gran cantidad de periféricos. Como parte esencial de esta y de todas las prácticas, usaremos STM32CubeIDE, que está basado en Eclipse, para programar nuestros microcontroladores.

Vemos diferenciadas varias fases o etapas a lo largo de la práctica que nos indican el desarrollo que debemos llevar. En la primera fase tenemos que llevar a cabo la instalación del entorno de desarrollo STM32CubeIDE. Es una mezcla de varias herramientas de desarrollo que han ido evolucionando e integrándose en un solo entorno, SW4STM32 + Atollic TrueStudio + STM32CubeMX, los primeros eran entornos de desarrollo IDE y el último un generador automático de código de configuración de los microcontroladores STM. Particularmente, he usado la última versión disponible en la página web del fabricante, la versión 1.17.0.

Tras esto, nos disponemos a comenzar con la creación del proyecto y posteriormente las fases contenidas en la práctica recogidas en el documento a descargar de la Enseñanza Virtual.

3. Desarrollo de la práctica

3.1. Fase 1: Instalación del sistema de desarrollo STM32CubeIDE y del simulador QEMU

En esta primera fase, el objetivo es la correcta instalación del entorno de desarollo. En primer lugar, debemos visitar la página web de STMicroelectronics y descargarnos el software, en mi caso he descargado la versión 1.17.0. La instalación de este es muy sencillo, pues basta con abrir el ejecutable descargado y darle a siguiente hasta acabar el proceso.

Una vez instalado, lo abrimos y procedemos a instalar el simulador QEMU para, valga la redundancia, simular el hardware de esta práctica. Para instalarlo, nos vamos al menú superior y pulsamos en: *Help* → *Eclipse Marketplace* y dentro de esta venta buscamos *GNU MCU Eclipse* y el resultado es lo que debemos instalar. Por último, debemos ir al enlace de GitHub del enunciado de la práctica donde están los paquete de simulación para los ARM Cortex e instalarlos en nuestro equipo. Con estos pasos ya estaría todo preparado para programar.

3.2. Fase 2: Encender un LED

Tras proceder con la correcta instalación del entorno de desarrollo y los ajustes necesarios para el funcionamiento de este, se nos propone probar su funcionamiento desarrollando un simple programa que nos permite encender y apagar un LED de la placa que estamos usando para comprobar que todo funciona correctamente. En la guía nos indica cómo crear un proyecto para este caso y la placa que debemos usar, que se trata de la **NUCLEO-F103RB**. Es importante que cuando al crear el proyecto nos pregunte sobre si queremos inicializar los periféricos debemos decir que sí, de esta forma existe un determinado pin, como el del LED que queremos encender, se coloca por defecto como salida.

Una vez creado el proyecto, disponemos de una funcionalidad que nos permite generar el código de configuración de nuestro microcontrolador en base a si están los valores por defecto o si hemos modificado elementos como el reloj o los GPIO. Es importante que si tenemos una configuración y por cualquier motivo hemos tenido que modificar por ejemplo la frecuencia del reloj, debemos volver a generar el código para que las modificaciones se vean efectuadas. Para esta fase, no vamos a modificar nada de la configuración inicial, por lo que pulsamos en *Project* → *Generate Code* para generar nuestro fichero de configuración.

Procedemos a escribir el código en el fichero *main.c*. Debemos escribir el código entre los comentarios de **/*USER CODE BEGIN... *//* USER CODE END ... *** para evitar que, en caso de que se vuelva a hacer, cuando generemos el código de configuración de nuestro proyecto se pierda.

La estructura es sencilla, primero los includes y luego el código. Entonces, arriba del main, en la sección de los includes debemos escribir lo siguiente:

```
1    void SystemClock_Config(void); // sirve para inicializar el reloj
2    static void MX_GPIO_Init(void); // sirve para inicializar los GPIO
3    static void MX_USART2_UART_Init(void); // sirve para inicializar un
        puerto serie
```

Posterior a esto debemos ir a nuestra función main, donde se encuentra el while(1) tan importante y fundamental de los microcontroladores en el que debemos escribir nuestro código:

```
1    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 0); pone a cero el LED
2    HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13); devuelve el valor del boton
```

3.3. Fase 3: Utilizar la interrupción de pulsar botón para cambiar el estado del LED

3.4. Fase 4: Hacer que le LED se encienda y apague a un determinado ritmo

4. Contestación de preguntas

5. Conclusiones