



Sistemas Operativos

Resumen Lectura 2

Eduardo Flores Gallegos

26/04/2018

Utilizando Arduino Due en la docencia de la entrada/salida

1. Motivación

En 2013 tomamos la decisión de modificar la arquitectura de referencia de las asignaturas relacionadas con la materia de Arquitectura de Computadores. Decidimos cambiar de la arquitectura MIPS a la ARM.

La arquitectura ARM presenta muchas características que la distinguen de otras arquitecturas contemporáneas y, por otro lado, al estar basada en RISC, es relativamente sencilla. Además, el hecho de que ARM sea una arquitectura actual y ampliamente difundida, especialmente en dispositivos móviles, smartphones y tablets, es un factor especialmente motivador para los estudiantes.

En ese caso, el estudiante debería programar en ensamblador el código necesario para interactuar con los dispositivos de entrada/salida proporcionados por el simulador que fuera a utilizarse. el estudiante interactuaría con dispositivos físicos, debería ser consciente de qué es lo que quiere que pase en la realidad y podría observar cómo su aplicación es capaz, o no, de reaccionar adecuadamente ante eventos externos.

Así pues, una vez optamos por la interacción con dispositivos físicos, quedaba por decidir qué componente hardware utilizar en el laboratorio. Puesto que nuestras restricciones era que dicho componente estuviera basado en la arquitectura ARM y que fuera de bajo coste, para que el estudiante pudiera adquirirlo en el caso de querer experimentar por su cuenta, la tarjeta Arduino Due, basada en ARM y con un precio económico, nos dio la impresión de ser una buena elección.

El modelo Due de Arduino3 presenta las siguientes características que consideramos interesantes desde el punto de vista de la docencia de la entrada/salida utilizando ARM como arquitectura de referencia:

- Microcontrolador Atmel SAM3X8E con una CPU ARM Cortex-M3.
- 54 pines de entrada/salidas digitales (de los cuales 12 pueden utilizarse como salidas PWM).
- 12 entradas analógicas.
- 4 UARTs (puertos serie hardware).
- Conexión USB con capacidad on-the-go (OTG).
- 2 conversores digital a analógico.
- Reloj (a 84 Mhz).
- Conectores SPI y JTAG.

la programación de la tarjeta Arduino se realiza habitualmente en C/C++. Es más, el entorno de desarrollo Arduino solo soporta dichos lenguajes. Teniendo en cuenta la gran variedad de modelos Arduino, esta decisión entendemos que está motivada por la intención de que, si alguien desarrolla un código en C/C++ para una determinada tarjeta Arduino, éste podrá utilizarse en otros modelos de tarjetas Arduino sin ningún problema. para el caso que nos ocupa, la gestión de la entrada/salida a bajo nivel, las funciones en C/C++ proporcionadas por el entorno Arduino abstraen una gran parte

de la problemática relacionada con la gestión de la entrada/salida desde el punto de vista de un arquitecto de computadores. Así que aparentemente no podíamos contar con el entorno de desarrollo de Arduino para programar directamente la tarjeta Arduino Due, a no ser que decidiéramos ocultar lo que realmente ocurre a nivel del lenguaje máquina. Afortunadamente, y puesto que el entorno de desarrollo de Arduino es software libre, fuimos capaces de introducir las modificaciones necesarias para que dicho entorno aceptara directamente código en ensamblador.

2. Contexto docente

La asignatura Estructura de Computadores es de formación básica y se imparte en primer curso, primer semestre en los Grados en Ingeniería Informática.

Con el objetivo de dar respuesta a dicha competencia, la asignatura se ha estructurado en tres temas, el tercero de los cuales abarca los sistemas de entrada/salida. Para dicho tema se han definido los objetivos formativos que se detallan a continuación. El estudiante, al finalizar la asignatura, deberá ser capaz de:

1. Describir la problemática general de la entrada/salida.
2. Definir los términos de latencia, tasa de transferencia y productividad y ser capaz de estimar o calcular sus valores partiendo de datos suficientes.
3. Caracterizar los sistemas de entrada/salida en función de su comportamiento, interlocutor, tasa de transferencia y latencia.
4. Describir la estructura general de los dispositivos de entrada/salida.
5. Enumerar y describir las formas por medio de las cuales el procesador puede acceder a los dispositivos de entrada/salida.
6. Enumerar los tipos de registros de los dispositivos de entrada/salida y explicar cómo y para qué se utilizan.
7. Explicar y evaluar el funcionamiento, así como las ventajas y desventajas, de los dos sistemas de sincronización de la entrada/salida: I) consulta de estado (polling) e II) interrupciones.
8. Explicar y evaluar el funcionamiento, así como las ventajas y desventajas, de los dos sistemas de transferencia de datos de entrada/salida: I) transferencias por programa realizadas por el procesador y II) acceso directo a memoria (DMA).
9. Describir las características generales de la entrada/salida de propósito general (GPIO) y de los mecanismos de temporización de los ordenadores.
10. Analizar programas escritos en ensamblador Thumb 2 que gestionen dispositivos de entrada/salida.
11. Desarrollar programas en ensamblador Thumb 2 que gestionen dispositivos de entrada/salida.

Para la consecución de los anteriores objetivos formativos, se ha optado por dividir la materia objeto de estudio en los siguientes cinco bloques:

1. Introducción y problemática general de la entrada/salida.
2. Aspectos generales de la entrada/salida de propósito general (GPIO) y de la temporización.
3. Gestión de la entrada/salida por medio de consulta de estado.
4. Gestión de la entrada/salida por medio de interrupciones.
5. Transferencia de datos por procesador y por acceso directo a memoria.

3. Entrada/salida con Arduino

El recurso docente presentado en este artículo pretende proporcionar material teórico y de laboratorio con el objetivo de facilitar la comprensión por parte del estudiante de la gestión de la entrada/salida desde el punto de vista de la arquitectura de computadores. Consta de:

- Los tres últimos capítulos de 3.
- Programas de ejemplo.
- El entorno de desarrollo de Arduino modificado para aceptar ficheros en ensamblador.
- Esquemático de un circuito sencillo con un pulsador y un diodo LED RGB.
- Solución del profesor.

Conviene tener en cuenta que para poder realizar las prácticas propuestas, tan solo será necesario disponer, por cada equipo de estudiantes, del siguiente material:

- 1 Computador con GNU/Linux, MacOSX o Windows.
- 1 Tarjeta Arduino Due.
- 1 Tarjeta de entrada/salida basada en el esquemático proporcionado

3.1. Capítulos de entrada/salida

Se ha organizado en tres capítulos:

El primero de ellos proporciona una introducción teórica a la entrada/salida. Comienza hablando de las generalidades y problemática asociada a la entrada/salida. Describe la estructura de los sistemas y dispositivos de entrada/salida. Continúa con la gestión de la entrada/salida por medio de consulta de estado y de interrupciones. Por último, describe las transferencias de datos y la DMA.

El segundo de los capítulos se centra en los dispositivos de entrada/salida. Primero desde un punto de vista genérico y después, particularizado a la GPIO del ATSAM3X8E (el microcontrolador incorporado en la tarjeta Arduino Due). El segundo apartado abarca la gestión del tiempo. Al igual que antes, en un primer lugar de forma genérica, para después particularizarlo a la gestión del tiempo del ATSAM3X8E. El tercer apartado cubre la gestión de excepciones e interrupciones en el ATSAM3X8E. El último apartado, el controlador de DMA del ATSAM3X8E.

El último de los tres capítulos está destinado al entorno de prácticas. Comienza describiendo la tarjeta Arduino Due, la tarjeta de entrada/salida, el entorno de desarrollo y la creación de proyectos en dicho entorno. A continuación, se proponen ejercicios para las cuatro sesiones de laboratorio que se detallan seguidamente.

Introducción a la entrada/salida. En esta primera sesión se muestra cómo encender o a apagar el LED de la tarjeta de entrada/salida.

Entrada/salida por consulta de estado. En esta sesión se incide sobre la consulta de estado y su problemática.

Entrada/salida por interrupciones. En esta sesión el estudiante debe configurar la tarjeta Arduino para activar las interrupciones y probar las distintas configuraciones disponibles.

Modulación por ancho de pulso (PWM), acceso directo a memoria (DMA) y puerto USB. En primer lugar, el estudiante verá cómo se puede disminuir la intensidad percibida del LED de la tarjeta de entrada/salida sin más que modificar la proporción de tiempo que éste permanece encendido en un ciclo de trabajo. En la segunda parte de la práctica, explorará la utilización del controlador de acceso directo a memoria (DMA) para la transferencia de bloques de memoria.

3.2. Programas de ejemplo

El objetivo de las prácticas es el de realizar programas en ensamblador de ARM que sean capaces de configurar e interactuar con los dispositivos de entrada/salida presente en la tarjeta Arduino Due.

Dichos programas de ejemplo se proporcionan también como parte de este recurso docente. En ellos se han ocultado fragmentos de código y valores concretos de etiquetas, máscaras y direcciones de E/S para que los alumnos los completen, manteniendo la estructura del programa y los comentarios descriptivos de las acciones que debe realizar cada sección del código y cada instrucción para facilitar la comprensión por parte del alumno.

3.3. Entorno de desarrollo modificado

El entorno de desarrollo de Arduino, que puede descargarse desde su página web, no soporta la realización de proyectos Arduino en otros lenguajes de programación que no sean C o C++. Utilizando esta versión modificada es posible desarrollar proyectos que tengan ficheros fuente tanto en C/C++, como en ensamblador de ARM. Esto nos ha permitido graduar la dificultad de las

prácticas de tal forma que determinadas acciones se realizan inicialmente en C/C++ y posteriormente se muestra qué acciones se deben realizar a bajo nivel para conseguir el mismo efecto.

3.4. Circuito sencillo con un pulsador y un diodo LED RGB

Aunque existen multitud de extensiones hardware (shields) de entrada/salida para la tarjeta Arduino Due. Llegamos a la conclusión de que, puesto que la tarjeta Arduino Due ya proporciona otros dispositivos de entrada/salida como un temporizador, un reloj de tiempo real y un controlador de DMA, era suficiente con un circuito simple como el mostrado en la placa. Dicho circuito está formado por 4 resistencias, un pulsador y un diodo LED RGB.

De esta forma, es posible provocar eventos en la tarjeta por medio del pulsador y visualizar su salida por medio del diodo LED RGB. Hay que tener en cuenta que, al tratarse de un diodo LED RGB, se dispone en realidad de tres salidas, cada una de ellas asociada a uno de los colores: rojo, verde y azul. Además, se ha diseñado una placa con dicho circuito de tal forma que pueda insertarse directamente en la tarjeta Arduino Due,

Un shield más complejo (por ejemplo una pantallaLCD) podría resultar más vistoso, pero además de ser más caro, una de dos, o se habría añadido una complejidad innecesaria en el caso de gestionar su entrada/salida a bajo nivel, o se hubiera ocultado el funcionamiento de la entrada/salida en el caso de haber utilizado las funciones en C/C++ proporcionadas por el correspondiente fabricante.

3.5. Otros dispositivos interesantes

La diversidad de dispositivos presentes en el microcontrolador ATSAM3X8E hace que sea necesario buscar un equilibrio adecuado entre qué es necesario para abarcar todos los objetivos de la asignatura y qué puede resultar útil y vistoso, motivando al estudiante, pero añade una complejidad excesiva para primer curso.

Nuestra elección se concretó en el uso, como dispositivos fundamentales, de la GPIO (entrada/salida de propósito general) y del reloj de tiempo real como elemento de temporización. Ambos permiten su gestión mediante prueba de estado o interrupciones, y son suficientemente sencillos y fáciles de comprender.

Para el segundo grupo, se utilizó el PWM (modulación de anchura de pulsos) y la interfaz de dispositivo USB. Mediante modificaciones sencillas del código que se les proporcionaba, los estudiantes pudieron ver un uso más avanzado de la entrada/salida, sin que hubiera una gran complejidad conceptual en los ejercicios.