

REFLEXIONES SOBRE LA ENSEÑANZA EN EL DISEÑO DE SISTEMAS EMBEBIDOS

REFLECTIONS ON TEACHING IN THE DESIGN OF EMBEDDED SYSTEMS



Acevedo González Geoffrey

Escuela de Ciencias Básicas, Tecnología e Ingeniería, Universidad Nacional Abierta y a Distancia, Medellín, Colombia, georffrey.acevedo@unad.edu.co

Recibido: 12/08/2012 • Aprobado: 15/11/2012

RESUMEN

En el diseño de soluciones electrónicas en áreas tan diversas como automóviles, automatización industrial, aplicaciones médicas, comunicaciones, control de potencia, domótica, electrodomésticos, entretenimiento, iluminación, periféricos para PC, productos de consumo, seguridad, toma de mediciones, captura de señales, entre otras, surge la necesidad de disponer de circuitos integrados que cumplan con funciones específicas. Es aquí donde el diseñador debe elegir entre una amplia variedad de opciones entre fabricantes y entre lo que se denomina familias de circuitos integrados de cada fabricante. Este artículo pretende aportar elementos para el debate sobre los criterios que determinan la elección de un microcontrolador y las competencias que, en consecuencia, la Universidad debe buscar desarrollar en el profesional de la electrónica.

Palabras clave: *atmel, AVRAtudio, code warrir, freescale, microchip, motorola*

ABSTRACT

In the design of electronic solutions in areas as diverse as automotive, communications, automation, electronics, entertainment, industrial automation, medical applications, PC peripherals, consumer products and lighting among others, comes the need for integrated circuits that meet specific functions. This is where the designer must choose from a wide variety of options between manufacturers and between families called integrated circuit manufacturer. This article aims to provide elements for discussion on the criteria according to a particular need determine the choice of a microcontroller and the powers that result, the University should seek to develop in the professional electronics.

Keywords: *atmel, AVRAtudio, code warrir, freescale, microchip, motorola*



I. INTRODUCCIÓN

En la búsqueda de la mejor herramienta para el desarrollo de sistemas embebidos, no son pocas las opciones que el diseñador tiene en cuanto a la elección del microcontrolador: *Analog Devices, AMD, Atmel, Cirrus Logic, Cygnal Integrated Products, Cypress, Freescale (Motorola), Dallas Semiconductor, Fujitsu, Infineon, Intel, Maxim (Dallas), Microchip, NS (National Semiconductor), OKI Semiconductor, Philips, Rabbit Semiconductor, Renesas, ST Microelectronics (SGS-Thomson), SMC, TDK, Temic Semiconductor, Texas Instruments, Toshiba, Triscend, Ubicom (Scenix), Zilog* son, entre otros, los fabricantes. Esto implica, que el profesional de la electrónica cuente con las competencias necesarias para tomar la mejor decisión con la mayor objetividad.

A continuación se analizan algunos de los criterios que determinan la elección del microcontrolador.

II. CONSIDERACIONES PARA LA TOMA DE DECISIONES

A. Necesidades del proyecto

Este factor comprende dos aspectos: los requerimientos en cuanto a periféricos y los requerimientos en cuanto a capacidad de procesamiento; así mismo, puede implicar la necesidad de realizar cálculos específicos para la aplicación, como son los cálculos matemáticos con coma flotante. En cuanto a módulos, un proyecto exige identificar las prestaciones mínimas del microcontrolador, el cual, deberá cumplir con unas características de memoria, líneas de E/S, UARTsetc, I2C, SPI, RS232 temporizadores, Conversores Analógico/Digital (CAD), Conversores Digital /Analógico (CDA), unidades de comparación /captura /PWM, USB, receptores/transmisores de RF, *Ethernet*, Unidad de punto Flotante (FPU), entre otras.

B. Naturaleza de la aplicación

En una aplicación orientada a la investigación, los factores que toman parte en una producción comercial suelen reducirse a la necesidad de cumplir requisitos en tiempo real.

C. Consumo de energía

La vida útil de una batería está vinculada a la reducción del tiempo en el procesamiento de operaciones complejas y, con ello, en aumentar el tiempo que esta podrá permanecer en estado de reposo.

D. Costo

¿Cuál es el precio que el cliente estaría dispuesto a pagar por el producto? A pesar de las necesidades particulares del proyecto, cuando de producción comercial se trata, el costo aparece como el factor determinante. Este es el punto en el que se considera la viabilidad del proyecto; en el que se fija el precio del producto de acuerdo con los resultados de la investigación de mercado, la cual toma su parte como el factor decisivo, cuando se trata de producir en cantidad y de considerar la competencia.

E. El mercado local

A pesar de que hasta aquí se pueda encontrar que familias como *Atmel, Freescale o Microchip* puedan satisfacer las necesidades particulares del proyecto en cuanto a prestaciones vs costo, es el momento en el que aparecen consideraciones de peso, como la disponibilidad del producto en el mercado local y el soporte para la región.

F. Tiempo

Se hace referencia al tiempo disponible para el desarrollo del proyecto. De la misma manera que un ingeniero de sistemas, en aras del tiempo o la potabilidad, puede verse en la necesidad de programar en Java, sacrificando con ello la eficiencia que resta usar una máquina virtual [1], el ingeniero electrónico podrá verse en la necesidad de elegir por una programación en lenguaje de



alto nivel, de acuerdo con la disponibilidad de información.

G. Características físicas

Estas suelen determinar la necesidad de un tipo de encapsulado de acuerdo con el espacio disponible. Esta puede resultar en una restricción en la gama de opciones.

H. Flexibilidad del proyecto

Bajo este criterio suele elegirse un lenguaje de programación que sea común entre diferentes fabricantes; actualmente, el lenguaje C representa la mejor opción. No obstante, un programador con las competencias que desarrolla el manejo de un lenguaje en bajo nivel, no encontrará en este su mayor obstáculo. En este campo, *Atmel* con *AVRStudio*, *freescala* con *CodeWarrior*, y *Microchip* con *MPLAB* ofrecen excelentes entornos de desarrollo en lenguaje C.

I. Licencias

El sistema operativo bajo el cual la empresa hace sus desarrollos también tiene su parte en la toma de una decisión. Los costos que representan las licencias del sistema operativo Windows entran en consideración cuando la empresa naciente ha de tener cuidado en la inversión de sus recursos. Este aspecto involucra factores como la seguridad, la estabilidad y el bajo costo. Para los usuarios del sistema operativo Linux, la respuesta pareciera ser única; no obstante, aunque tanto *Freescala* como *Microchip* ofrecen soluciones bajo estos sistemas, ambas plantean actualmente sus propias dificultades.

III. COMPETENCIAS DEL DISEÑADOR

Son comunes las discusiones apasionadas por una u otra solución: hardware o software. Estas discusiones son animadas generalmente por un docente que, tradicionalmente, opta por un único sistema, de acuerdo con su escuela, experiencia, simplicidad o preferencias.

Así como el estudiante de sistemas debe contar con las competencias para migrar de un lenguaje de programación a otro, sea este Visual Basic, C, JAVA o PHP entre otros, de acuerdo con unas necesidades de tiempo o recursos, el estudiante de electrónica, en el campo del diseño de sistemas embebidos, deberá estar en capacidad de evaluar las necesidades particulares del proyecto, con el fin de tomar la mejor decisión, como una acción objetiva y no basada en prejuicios.

En lugar de alimentar ideas temerarias frente a una herramienta clasificándola como una solución de técnicos o ingenieros, la toma de decisiones se debe basar en criterios sujetos a cada proyecto. Un entrenamiento orientado de manera exclusiva a un único fabricante o en sistemas de programación exclusiva de alto nivel, crea barreras para lograr la mejor solución.

El lenguaje ensamblador constituye la representación más directa del código de máquina específico para cada arquitectura física de microcontrolador, lo que permite la manipulación directa del hardware. Esto se logra mediante el manejo directo de los registros del procesador, la administración de la memoria y los periféricos.

Esto hace que la programación en bajo nivel sea necesaria cuando se requieren altos rendimientos, o una máxima optimización en cuanto al uso de los recursos; de la misma manera, permanecer cerrado a la programación bajo una única herramienta o permanecer cerrado en la elección de un microcontrolador de una familia y fabricante específico puede llevar al fracaso a un proyecto orientado a fines comerciales.

Al programar en bajo nivel, se adquieren las competencias para migrar a sistemas de programación de alto nivel, debido a que no se dispone de estructuras de control como *if else else if*, *while*, *do ... while*, *for*; en su lugar, el programador deberá trabajar con pocas instrucciones.

De allí que el estudiante así entrenado, pueda migrar con facilidad a lenguajes de alto nivel, como el lenguaje C, o herramientas de programación mediante una interfaz gráfica. Lo contrario no ocurre al iniciar al profesional de la electrónica en sistemas de programación de alto nivel en los que sólo interpreta bloques que lo abstraen cada vez más de la comprensión del sistema.

El estudiante deberá tener la oportunidad de descubrir desde la academia los pros y los contras de programar para una arquitectura RISC (*Set reducido de instrucciones*) frente a una arquitectura CISC (*Set complejo de instrucciones*), ensamblando tanto en C, C++ como en lenguaje de máquina para diferentes familias y fabricantes; igualmente, podrá comparar la capacidad de procesamiento en cada caso, como por ejemplo, el de espacio ocupado en memoria. Deberá poder trabajar con periféricos como unidades de tiempo o memorias que demanden interpretar correctamente un *datasheet* específico y adaptar códigos diseñados para otro microcontrolador o en otro lenguaje, como diseñar retardos precisos en *assembler* considerando para ello los ciclos de reloj de cada instrucción.

De esta manera, la institución educativa debe, desde un aprendizaje significativo, ofrecer a sus aprehendientes, el desarrollo de las competencias para el diseño de la mejor solución, en contraste a posiciones cerradas.

AGRADECIMIENTOS

Agradecimientos al Ingeniero Jhon Arroyave por sus valiosos aportes.

REFERENCIAS

- [1] U. Stern., (2001). Java vs. C++. Research Associate in the Computer Science Department at Stanford University. [en línea]. Disponible en: http://verify.stanford.edu/uli/java_cpp.html
- [2] Analog Devices, Inc (1995 - 2001). DSPs and Analog Microcontrollers. [en línea]. Disponible en: <http://www.analog.com>
- [3] Atmel Corporation (2011). Atmel AVR 8- and 32-bit Microcontrollers [en línea]. Disponible en: <http://www.atmel.com>.
- [4] Freescale Semiconductor, Inc. (2004 – 2011). Design Resources. [en línea]. Disponible en: <http://www.freescale.com>.
- [5] G. Galeano, (2009). *Programación de Sistemas Embebidos en C*. Alfaomega S.A. Bogotá D.C. Colombia., 2009.
- [6] Microchip Technology Inc (1998-2011). Why buy PIC® Microcontrollers (MCUs) & dsPIC® Digital Signal Controllers (DSCs). [en línea]. Disponible en: <http://www.microchip.com>.
- [7] Texas Instruments Incorporated (1995-2011). Microcontrollers for Ultra-Low Power, High-Performance Real-Time Control and ARM-based Embedded Design Solutions. [en línea]. Disponible en: <http://www.ti.com>.
- [8] National Semiconductor Corporation (2011). COP8 Summaries and Selection Guide Microcontrollers Integrated Op Amps And Temp Sensor.[en línea]. Disponible en: <http://www.national.com/appinfo/mcu/>