

ENTORNO PARA EL DESARROLLO DE SISTEMAS DIGITALES SOBRE EL MICROCONTROLADOR MCF5272: EDCOLDFIRE

R SAN-SEGUNDO, F. FERNÁNDEZ, A. ARAUJO, J.M. MONTERO, J. MACÍAS-GUARASA
Departamento de Ingeniería Electrónica. Escuela Técnica Superior de Ingenieros de Telecomunicación (ETSIT). Universidad Politécnica de Madrid (UPM).

En este artículo se describe el programa EDColdFire. Este programa es un entorno integrado para el desarrollo de sistemas digitales basados en el microcontrolador de Motorola ColdFire 5272 a través de la plataforma hardware DANTARES. Este entorno permite editar, cargar, ejecutar y depurar programas en ensamblador y C para este microcontrolador. El EDColdFire ha sido desarrollado por el Dpto. de Ingeniería Electrónica de la ETSI Telecomunicación (ETSIT) de la Universidad Politécnica de Madrid (UPM) y está siendo utilizado con éxito por más de 400 alumnos en esta universidad y en la Rey Juan Carlos. En esta comunicación se comentan las principales características del programa así como la experiencia de uso por parte de los alumnos: proyectos desarrollados con el entorno, estadísticas de utilización de cada una de las opciones del programa (que revelan diferentes hábitos de programación y depuración) y la valoración que los alumnos hacen en las encuestas.

1. Introducción

Durante los últimos años el Departamento de Ingeniería Electrónica de la ETSIT de la UPM está realizando una importante labor de renovación de los laboratorios docentes basada en su amplia experiencia en proyectos de I+D con microcontroladores de Motorola y en la aplicación de la estrategia de aprendizaje basado en proyectos (PBL: Project-Based Learning) en laboratorios masivos. Como consecuencia de esta labor de renovación se ha definido un nuevo currículo en sistemas electrónicos basado en PBL [1][2][3].

En este proceso de renovación se ha generado nuevo material docente para gran parte de las asignaturas (p.e. en el Laboratorio de Sistemas Electrónicos Digitales, LSED): una nueva plataforma hardware DANTARES [4] (basada en el microcontrolador MCF5272), un nuevo libro docente [5], y un conjunto de herramientas web para la gestión de laboratorios docentes y el seguimiento de los alumnos [6][7][8]. Aunque este material ha sido generado principalmente para la docencia de sistemas digitales, su potencia permite abordar asignaturas de programación a bajo nivel sobre sistemas operativos sencillos. Esta comunicación describe el entorno EDColdFire para el desarrollo de sistemas digitales con la plataforma DANTARES, que ha sido comercializado a otras universidades como la Rey Juan Carlos. En la Figura 1 se muestra un puesto de laboratorio para el desarrollo de Sistemas Electrónicos Digitales utilizando el programa EDColdFire sobre la plataforma DANTARES.



Figura 1. Puesto de laboratorio: ordenador con el entorno EDColdFire y la plataforma DANTARES.

Este artículo está organizado de la siguiente manera. En el apartado 2 se describen las principales opciones de las que dispone el entorno EDColdFire V3.0: edición de código, ensamblado o compilación de programas, ejecución y depuración. El apartado 3 recoge la experiencia de uso por

parte de los alumnos: proyectos desarrollados con el entorno, estadísticas de utilización de cada una de las opciones del programa (que revelan diferentes hábitos de programación y depuración) y la valoración que los alumnos hacen en las encuestas. Finalmente el apartado 4 recoge las principales conclusiones de esta comunicación.

2. Entorno de desarrollo EDColdFire.

El entorno EDColdFire es un entorno visual en el que el alumno puede editar, cargar, ejecutar y depurar programas para el microcontrolador MCF5272 de Motorola. En el diseño de este nuevo entorno se ha considerado el hábito de los alumnos con entornos anteriores [9], mejorando sensiblemente su ergonomía. Este entorno consta de cuatro herramientas de desarrollo integradas que iremos describiendo a lo largo de esta apartado: edición de código en ensamblador y en C, ensamblado y compilado de programas, y carga y ejecución de programas. En este apartado se describen los aspectos más relevantes, para más información sobre el manejo del entorno se recomienda al lector que consulte el libro [10].

2.1 Edición de código en ensamblador y en C

El entorno dispone de un editor de texto multiventana, que permite escribir y editar varios archivos fuente en los distintos lenguajes de programación soportados: ensamblador de Motorola (*.asm), ensamblador de GNU (*.asg) y lenguaje C (*.c y *.h). Dependiendo de la extensión del archivo el entorno lo interpretará como código de uno u otro tipo. El editor de código dispone de las mismas herramientas de cualquier editor:

- Crear, abrir, guardar e imprimir archivos fuente.
- Seleccionar una o varias líneas de texto completas.
- Copiar y mover texto entre archivos.
- Copiar y mover texto usando arrastrar y soltar.
- Buscar y reemplazar texto.
- Deshacer la última operación de edición.

La edición de código incluye además dos ayudas al programador: ayuda emergente y un coloreado sintáctico (ver Figura 2).

The screenshot shows the EDColdFire IDE interface. The main window displays assembly code for 'Cadena.asm'. The code is color-coded: comments are green, labels and instructions are black, and registers and constants are blue. A pop-up help window is visible over the code, providing details for the 'MOVE' instruction. The help window has a table with the following columns: 'Sintaxis', 'Tamaños', 'Modos', 'OS MNZVC', and 'Función'. The table lists various forms of the MOVE instruction, such as 'ORG MOVE f,d', 'PPAL MOVE CCR,d', 'BUCLE MOVE f,CCR', 'BR MOVE SR,d', 'RTS MOVE f,SR', and 'MOVE L ACC,d'. The 'Función' column describes the operation, such as 'Mover datos', 'Mover de CCR a registro', 'Mover a CCR', 'Mover SR a registro', 'Mover a SR', 'Mover ACC a registro', 'Mover MACSR a registro', 'Mover MASK a registro', 'Mover a ACC', 'Mover a MACSR', 'Mover a MASK', and 'Mover MACSR a CCR'. The status bar at the bottom indicates 'Línea 1' and 'NUM'.

Figura 2. Ejemplo de código coloreado y ayuda emergente de las instrucciones MOVE en ensamblador

La ayuda emergente contiene información sobre las instrucciones de ensamblador: operando fuente y destino, tamaño de los operandos,..., y aparece cuando se mantiene el cursor sobre una instrucción del código durante un tiempo superior a 1 segundo. Si se observa el código en la Figura 2,

las instrucciones y directivas de ensamblador se marcan en azul y los comentarios en verde, diferenciándose así del resto del código.

2.2 Ensamblado y compilación de programas

El entorno dispone de dos programas ensambladores (Motorola y GNU) para ensamblar el código fuente, y dos compiladores cruzados que permiten la mezcla de código en C y en ensamblador de Motorola o GNU. Estas herramientas están disponibles sin salir del editor. Partiendo de un archivo con el código fuente (en ensamblador o C), el programa genera dos archivos adicionales:

- Archivo de depuración (*.dep): contiene el listado del código en ensamblador junto con las direcciones de memoria donde se situará tanto el código como los datos.
- Archivo objeto en formato S-record (*.hcf): archivo binario con la representación de las instrucciones y datos del programa desarrollado. Este archivo es el que se carga en la plataforma.

Para generar el módulo objeto de los programas durante la compilación y ensamblaje es preciso saber en qué regiones de memoria de la plataforma se ubicarán las instrucciones del programa y los datos. En el caso de los programas escritos en ensamblador de Motorola la ubicación vendrá determinada por los valores que fijemos utilizando la directiva ORG. En el caso de programas en C, estas regiones se definen en el menú Configurar construcción. En este cuadro se puede configurar la dirección de inicio del programa, la dirección de inicio de los datos y el nivel de optimización del compilador de C.

Una vez realizado el ensamblado o la compilación del programa, los avisos o errores detectados se presentan en una ventana. Con una doble pulsación sobre cada error el entorno localiza la línea de código donde se detecta el error (ver Figura 3). Esta funcionalidad permite conectar los avisos/errores, y la causa del aviso/error en el archivo fuente, facilitando su localización y corrección.

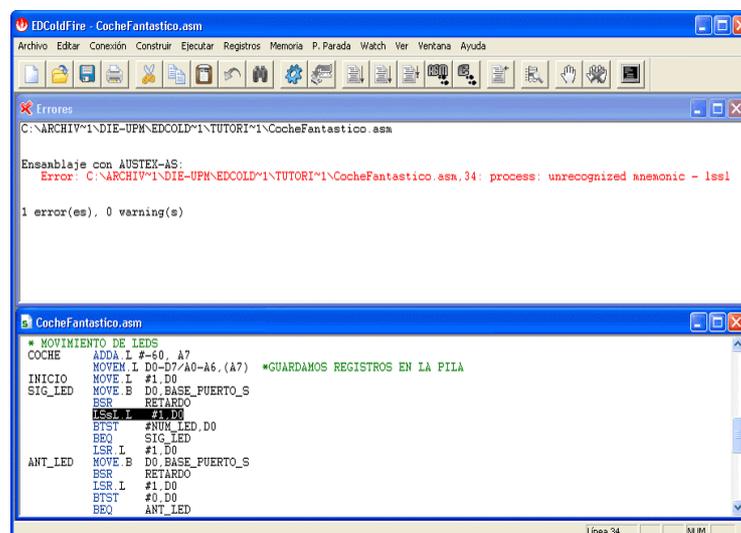


Figura 3. Ejemplo de localización del error en el código fuente.

2.3 Carga y ejecución de programas

Por último, el entorno dispone de un depurador de código en ensamblador y en C para cargar, ejecutar y depurar el programa desarrollado. Antes de cargar y ejecutar un programa en la plataforma DANTARES es necesario conectar el entorno EDColdFire y dicha plataforma. Al abrir una conexión entre ambos se visualizan en pantalla los valores de los registros internos del microcontrolador: registro de estado, contador de pila, registros de datos (D0-D7) y registros de direcciones (A0-A7) (ver Figura 4).

Al realizar la carga de un programa en la plataforma DANTARES se muestra el archivo de depuración (*.dep) que contiene el listado del código en ensamblador y que será la referencia sobre la que se apoyará el entorno para ir ejecutando el programa según los diferentes tipos de ejecución: paso a paso (ver Figura 4), paso a paso en C, ejecución hasta el cursor, y con/sin puntos de parada. La línea de código a ejecutar se remarca en color azul mientras que los puntos de parada se resaltan en color rojo. La opción de paso a paso permite ir ejecutando una por una las instrucciones en ensamblador, mientras que la opción de paso a paso en C ejecuta de manera conjunta el grupo de instrucciones (en ensamblador) asociadas a una línea de código en C.

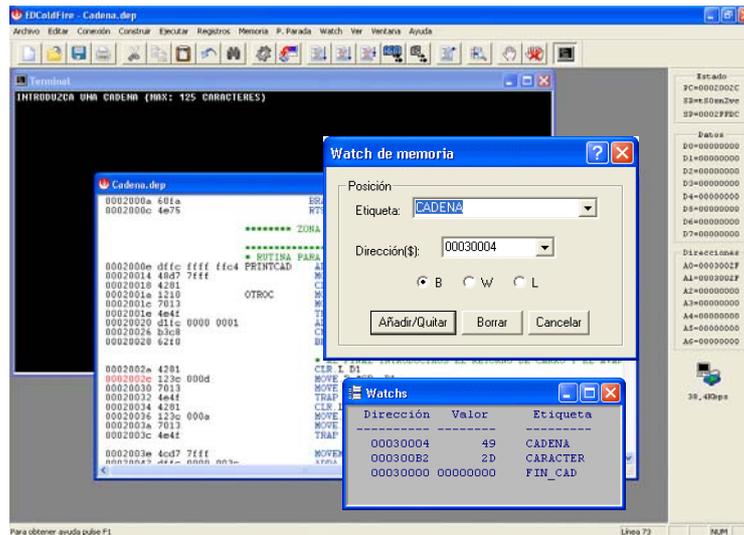


Figura 4. Ejecución detenida en un punto de parada. A la derecha se visualizan los registros de la CPU

Durante la ejecución de un programa y siempre que dicha ejecución esté detenida en un punto concreto, el entorno ofrece herramientas para visualizar y modificar tanto los registros de la CPU (acceso a través de la barra de la derecha), como variables o posiciones de memoria. Para acceder a una variable se puede hacer especificando la dirección de dicha variable o seleccionando su nombre de una lista (mayor comodidad). En el caso de un programa desarrollado en C, sólo se tiene acceso por el nombre a las variables globales al programa. Para acceder a las variables locales a una función es necesario acceder a la dirección de la pila correspondiente.

2.4 Otras herramientas

Adicionalmente a las herramientas comentadas, el entorno incorpora otras que facilitan la labor del desarrollador:

- Un conversor para convertir un número en base decimal a hexadecimal y viceversa.
- Una ventana terminal que sirve de interfaz entre el programa y un posible usuario del programa. A través de esta ventana el programa puede imprimir mensajes de texto y solicitar del usuario la pulsación de datos en el teclado del ordenador.
- El programa incorpora también una ayuda detallada de todas las opciones del entorno así como un tutorial de manejo del entorno.
- Finalmente el entorno ofrece los programas en ensamblador y en C correspondientes a las guías de autoaprendizaje desarrolladas en el libro [5]. Estas guías introducen al alumno en el manejo de los principales recursos de la plataforma DANTARES: puertos digitales de entrada-salida, módulos de PWM (Pulse Width Modulation), interrupciones externas, temporizadores, conversores AD (Analógico-Digital) y DA (Digital-Analógico).

3. Experiencia y resultados

En este apartado describiremos la experiencia de uso por parte de los alumnos del entorno EDColdFire, incluyendo una breve descripción de los proyectos desarrollados con el entorno, un análisis de las estadísticas de utilización de cada una de las opciones del programa (que revelan diferentes hábitos de programación y depuración) y la valoración que los alumnos hacen en las encuestas de dicho programa.

3.1 Proyectos desarrollados con el entorno EDColdFire sobre la plataforma DANTARES

El EDColdFire ha sido utilizado por más de 700 alumnos a lo largo de los cursos 2004-2005 y 2005-2006 en el LSED: Laboratorio de Sistemas Electrónicos Digitales.

En el curso 2004-2005 el proyecto básico a desarrollar consistió en un sistema de control de la velocidad de un motor de corriente continua. Para realizar el control del motor debían utilizar el módulo PWM para la excitación del motor, y las interrupciones externas para contar las vueltas del motor y calcular su velocidad en RPS (revoluciones por segundo). Adicionalmente los alumnos debían incorporar un sensor de temperatura para hacer corresponder la velocidad del motor a la temperatura ambiente, y un módulo de cálculo de estadísticas que permitiera obtener la velocidad media, máxima y mínima del motor (ver Figura 5).

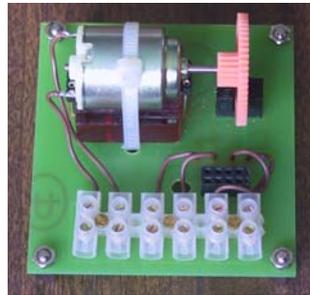


Figura 5. Motor de corriente continua y par optoacoplado para contar las vueltas

En el curso 2005-2006 el proyecto básico ha consistido en un sistema domótico de monitorización en el hogar. Un módulo externo transmite sincronamente tramas de datos con la información de tres sensores: agua, humo y gas. El sistema de monitorización (basado en el ColdFire) debe recibir dichas tramas y generar diferentes alarmas si el valor de los sensores supera un nivel durante un tiempo especificado. El sistema de monitorización debe permitir configurar tanto los niveles de los sensores como los tiempos límite para la generación de las alarmas. El proyecto incluía tanto el desarrollo del emisor como el sistema de monitorización basado en el ColdFire.



Figura 6. Placa de interfaz con un teclado matricial y un LCD (Liquid Cristal Display)

Para el desarrollo de ambos proyectos los alumnos tenían que utilizar como interfaz del programa-usuario una placa externa con un teclado matricial y un LCD (Liquid Cristal Display) (facilitada por el Departamento) que debían programar correctamente. En la Figura 6 se puede ver un ejemplo de dicha placa.

Aparte de los proyectos básicos, los alumnos que lo desearan podían desarrollar proyectos especiales. A lo largo de estos dos años se han desarrollado una gran variedad de proyectos especiales mostrando una visión amplia de los sistemas electrónicos como la base para el desarrollo de proyectos en el resto de disciplinas:

- *Procesado de señal.* En esta línea cabe resaltar el proyecto de un sistema de reconocimiento de voz. El sistema desarrollado está formado por varios módulos: grabación de voz, detección de actividad vocal, extracción de características (energías de la señal de voz filtrada con un banco de filtros) y comparación de patrones acústicos mediante un algoritmo de programación dinámica.
- *Aplicaciones multimedia.* En este punto el proyecto principal ha sido el control de un teléfono móvil mediante los comandos AT. Para ello se ha utilizado una conexión serie a través del módulo de la UART disponible en el ColdFire. El proyecto desarrollado permitía el envío y recepción de SMS, y la realización de llamadas con reproducción de mensajes de audio.
- *Comunicaciones.* En esta faceta cabe resaltar el desarrollo de un cliente TFTP (Trivial FTP) utilizando el módulo Ethernet disponible en el ColdFire. La aplicación desarrollada permite la transferencia de ficheros entre un servidor TFTP y la memoria del ColdFire. Además del manejo del módulo Ethernet fue necesario implementar los protocolos UDP e IP.
- *Robótica.* Finalmente los proyectos relacionados con robótica han consistido en el control de una hormiga mecatrónica que camina en la búsqueda de una fuente de luz y un brazo articulado para la pulsación de un teclado o el movimiento de objetos (ver Figura 7).



Figura 7. Robot hexadópodo y Brazo articulado (www.lynxmotion.com)

3.2 Estadísticas de uso por parte de los alumnos

La ventaja más importante de utilizar un entorno como el EDColdFire, desarrollado por nuestro propio Departamento, es disponer del código fuente para ir modificando el programa según los hábitos y necesidades de los alumnos. Con este objetivo, durante el curso académico 2005-2006 el entorno se ha modificado para poder obtener información sobre el uso que hacen los alumnos de las diferentes herramientas y opciones. Gracias a esta información es posible detectar diferentes hábitos en los alumnos y deducir posibles mejoras a incorporar en el entorno. Las principales estadísticas de la información obtenida se describen a continuación.

La realización de la parte software del proyecto se puede realizar tanto en lenguaje ensamblador como en C. En el curso 2005-2006, sólo el 20% de los alumnos realizaron la práctica en C a pesar de ser un lenguaje de alto nivel. El motivo principal es que en la asignatura teórica de Sistemas Electrónicos Digitales en la que se describe el microcontrolador CF5272, se dedican hasta 10 horas lectivas de programación con ensamblador mientras que el lenguaje C no se imparte en ninguna

asignatura previa (en las asignaturas de programación de los primeros cursos se imparte Java). Ante esta situación los alumnos prefieren mayoritariamente el ensamblador por resultarles más familiar.

A lo largo de la descripción se diferenciará entre los alumnos que han realizado el proyecto en ensamblador y los que lo han desarrollado en C. En este análisis se han considerado más de 1100 sesiones de laboratorio. Entendiendo por sesión de trabajo el tiempo transcurrido desde que se abre hasta que se cierra el entorno de desarrollo EDColdFire. En un turno de 4 horas de laboratorio los alumnos pueden realizar una o varias sesiones de trabajo.

En la Tabla 1, se muestran los valores de duración media de una sesión y los porcentajes de tiempo dedicados a desarrollar el código o a depurarlo.

Lenguaje	Duración media de una sesión	Tiempo Escritura, Ensamblado o Compilado	Tiempo Depuración y Ejecución
Ensamblador	1:13:00	69,7%	30,3%
C	1:14:19	69,9%	30,1%

Tabla 1. Porcentaje de tiempo dedicado a desarrollo de código y a depuración.

En ambos casos el porcentaje dedicado a depuración es bastante menor que el tiempo dedicado a la escritura, ensamblado o compilado del código. Este resultado nos lleva a dos conclusiones importantes:

- De cara al curso que viene debemos fomentar más el trabajo en casa donde se puede escribir, ensamblar y compilar el código, y dejar las fases de ejecución y depuración para el laboratorio.
- Parece ser que los alumnos invierten mucho tiempo observando el código en busca de posibles errores mientras que no dedican suficiente tiempo a ejecutar y depurar que es la mejor manera de detectar dichos errores. De cara a cursos posteriores es necesario formar al alumno en las diferentes estrategias de depuración.

Con el objetivo de profundizar en los hábitos de los alumnos a la hora de ejecutar y depurar un programa, hemos estudiado el tipo de estrategia utilizada. En la Tabla 2 se muestran los valores de los porcentajes de número de veces y tiempo dedicado a cada tipo de ejecución del programa.

Lenguaje		Paso a Paso	Paso a Paso en C	Con Puntos de Parada	Ejecución hasta el cursor	Sin Puntos de Parada
Ensamblador	Nº	71,4%	0,0%	19,1%	2,1%	7,4%
	Tiempo	16,8%	0,0%	32,6%	3,6%	47,0%
C	Nº	28,4%	9,7%	33,4%	0,7%	47,0%
	Tiempo	2,2%	0,7%	22,9%	0,2%	74,0%

Tabla 2. Porcentaje de número de veces y tiempo dedicado a cada tipo de ejecución ofrecida por el entorno.

La Tabla 3 muestra el porcentaje de las veces que se utiliza una u otra estrategia de visualización de datos en memoria. En este punto conviene comentar que existen otras dos opciones de visualización de información que están disponibles de forma continua para los alumnos: la visualización de los registros de la CPU y la presentación de información por el terminal del entorno.

Lenguaje	Tabla de Watches	Visualizando una posición de memoria concreta	Visualizando una zona de memoria
Ensamblador	26,6%	67,1%	6,3%
C	48,7%	50,4%	0,8%

Tabla 3. Estrategia utilizada por los alumnos para ver el valor de una variable en memoria.

De los datos presentados en las tablas Tabla 2 y Tabla 3 se pueden extraer las siguientes conclusiones:

- Tanto el porcentaje de tiempo como el porcentaje de veces que los alumnos utilizan la ejecución del programa sin ningún punto de parada es muy elevada. Siendo especialmente importante en el caso del programa desarrollado en C.
- En el caso de la programación en ensamblador, la estrategia de depuración combinada de “Puntos de Parada” y “Ejecución Paso a Paso” es la más importante llegando al 49,4% del tiempo dedicado a la depuración.
- En el caso de la programación en C el porcentaje mayoritario se lo lleva la “Ejecución Sin Punto de Parada” con un porcentaje muy elevado del tiempo 74,0%. Parece ser que el proceso de depuración en C ha consistido principalmente en este tipo de ejecución mostrando el valor de las variables mediante mensajes en el terminal del entorno.
- Finalmente comentar que la opción de visualización de memoria es muy interesante para ver errores en los tamaños de los datos (error muy frecuente entre los alumnos) y sin embargo no es una estrategia muy utilizada para la visualización de datos en tiempo de ejecución.

3.3 Valoración en las encuestas

En la Figura 8 se muestra un gráfico con la distribución de las calificaciones (entre 0 y 10) que los alumnos han dado al entorno EDColdFire en las encuestas para los cursos 04/05 y 05/06.

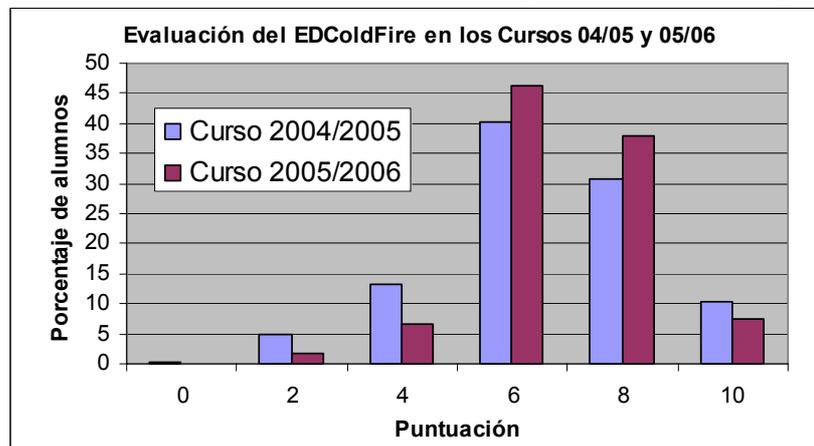


Figura 8. Resultados en las encuestas de la evaluación del EDColdFire por parte de los alumnos para los cursos 04/05 y 05/06.

Los alumnos valoraron en media con más de un 7 (sobre 10) su facilidad de uso. Como se puede ver la versión 3.0 del programa, utilizada en el curso 05/06, ofrece una mejor imagen a los alumnos. La nueva versión corrige algunos de los errores de las versiones anteriores y presenta un interfaz más atractivo.

4. Conclusiones

Esta comunicación presenta un nuevo entorno para el desarrollo de sistemas digitales con el microcontrolador MCF5272. En la descripción del entorno se detalla el conjunto de herramientas disponibles: edición de código, ensamblado, compilado, carga y depuración de programas.

En relación con la cantidad y variedad de proyectos desarrollados cabe comentar que la disponibilidad de herramientas software potentes como este entorno y de la plataforma DANTARES común a varios laboratorios (sin aprendizaje multiplataforma) permite rentabilizar el trabajo de los alumnos desarrollando proyectos más realistas y atractivos para ellos.

El hecho de tener acceso total al programa nos ha permitido recopilar información de uso de las diferentes opciones por parte de los alumnos. Información que nos ha permitido obtener las siguientes conclusiones relacionadas con los hábitos de uso:

- Los alumnos no utilizan correctamente todas las posibilidades que ofrece el entorno para la depuración de errores de los programas. Este hecho nos lleva a considerar la necesidad de

invertir tiempo en la formación de las diferentes estrategias de depuración y las herramientas del entorno a utilizar en cada estrategia.

- La depuración en lenguaje C, al realizarse también a nivel de ensamblador, no resulta muy cómoda para los alumnos que prefieren sacar mensajes por pantalla para ver el valor de ciertas variables en tiempo de ejecución. Es necesario dotar el entorno de nuevas estrategias que faciliten la depuración de los programas en C.

Finalmente comentar que el entorno de desarrollo EDColdFire ha sido bastante bien valorado por los alumnos en las encuestas dándole una puntuación media superior a 7 (sobre 10) en los dos cursos académicos en los que se ha implantado.

Agradecimientos

Los autores quieren agradecer a todos los miembros del Departamento de Ingeniería Electrónica de la ETSIT-UPM, su gran esfuerzo y dedicación que permite crear un marco de trabajo con gran experiencia y conocimiento en sistemas electrónicos. Los autores quieren agradecer en especial las contribuciones de Luca Necchi y Javier Guillén.

Referencias

- [1] A Project-Based Learning approach for Electronic Curricula Design. Javier Macías-Guarasa, Juan Manuel Montero, Rubén San-Segundo, Álvaro Araujo, Octavio Nieto. Aceptado en IEEE Transactions on Education. (ISSN: 0018-9359) (Impact Factor en “JCR Science Edition 2004”: 0,526).
- [2] Towards the Acquisition of Soft and Systemic Skills: a Project Based Learning Experience in Massive Laboratories on Electronics. Rubén San-Segundo, Juan M. Montero, Javier Macías-Guarasa, Javier Ferreiros and Ricardo de Córdoba. International Conference on Engineering and Computer Education 2005 (ICECE'05). (ISBN: 84-6098149-5) Madrid (Spain), 13-16 de Noviembre, 2005.
- [3] Curriculum en electrónica centrado en el aprendizaje basado en proyectos. Álvaro Araujo, Rubén San-Segundo, Javier Macías-Guarasa, Juan Manuel Montero, Octavio Nieto-Taladriz. VII TAAE 2006.
- [4] ANTARES: Una Integración de Docencia Universitaria e I+D+I en una misma Plataforma. Á. Araujo, J.M. Montero, D. Fraga, J. I. Izpura y O. Nieto-Taladriz). VI TAAE 2004, pp. 49-61 (ISBN: 84-688-7339-X).
- [5] Desarrollo de Sistemas Digitales con el microcontrolador MFC5272. R. San-Segundo et al. FUNDETEL (U.P.M) 2005. (ISBN 84-7402-315-7).
- [6] Tools and Strategies for Improving PBL Laboratory Courses with a High Student-to-Faculty Ratio. J. Macías-Guarasa, R. San-Segundo, J.M. Montero, J. Ferreiros and R. Córdoba. IEEE Frontiers in Education. 2005 (FIE'2005), pp. F2C_7-F2C_12. ISSN: 0190-5848. Indianapolis (EE.UU.) 19-22 de octubre de 2005
- [7] Automatic Tools For Diagnosis And Feedback In A Project Based Learning Course. R. San-Segundo, J.M. Montero, J. Macías-Guarasa, R.Córdoba and J. Ferreiros. IEEE Frontiers in Education 2005 (FIE'2005), pp. S2D_17-S2D_22. ISSN: 0190-5848. Indianapolis (EE.UU.) 19-22 de octubre de 2005.
- [8] Methodology For The Analysis Of Instructors' Grading Discrepancies In A Laboratory Course. Juan M. Montero, Rubén San-Segundo, Javier Macías-Guarasa, Ricardo de Córdoba, Javier Ferreiros. Aceptado en International Journal of Engineering Education (ISSN: 0949-149X) 2006 (Impact Factor en “JCR Science Edition 2004”: 0,228).
- [9] ED68K: un entorno de desarrollo para el diseño de sistemas digitales basados en el MC68000. J.M. Freire, M.L. López Vallejo, J. Colás en 2º Simposio Internacional en informática educativa SIIE'2000.
- [10] Entorno de desarrollo EDColdFire V3.0. R. San-Segundo et al. Fundación Rogelio Segovia para el desarrollo de las Telecomunicaciones 2005. (ISBN 84-7402-330-0 Depósito Legal: M-47921-2005)