

Hardware Copyleft como Herramienta para la Enseñanza de Procesamiento de Señales e Imágenes

Carlos Iván Camargo – Universidad Nacional de Colombia
cicamargoba@unal.edu.co

Abstract—El procesamiento digital de señales y de imágenes es un área que abarca un amplio rango de aplicaciones académicas y comerciales; es tema obligado en la mayoría de los centros educativos de formación superior y gracias al estado actual de la industria de los semiconductores es posible obtener dispositivos especializados en este tipo de procesamiento, lo que permite la creación de productos comercializables en esta área. Esto es de vital importancia en países en desarrollo como Colombia, donde la industria electrónica casi no existe y se depende por completo de productos provenientes de países más desarrollados, en especial de productos asiáticos. Este artículo presenta dos plataformas hardware *copyleft* que pueden ser estudiadas, reproducidas y modificadas incluso para fines comerciales; proporcionando una herramienta para el estudio de procesamiento de señales y de imágenes y un punto de partida para el desarrollo de empresas de base tecnológica que permitan disminuir los altos niveles de desempleo de los profesionales en áreas afines.

Index Terms—Sistemas Embebidos, Procesamiento de imagen, Linux, DSPs.

I. INTRODUCCIÓN

Estudios recientes [1], [2], [3], [4], [5] (que pueden ser aplicados a países de la región) revelan que la industria electrónica colombiana se encuentra muy atrasada en relación a los países industrializados; siendo una de las causas la utilización de metodologías de diseño y tecnologías obsoletas por parte de las empresas del sector, lo que se originó por la falta de profesionales con las habilidades necesarias para desarrollar productos comercializables que utilicen tecnología de punta, la fuerte dependencia hacia el mercado asiático, y la falta de políticas gubernamentales que protejan las industrias locales; esta situación ha eliminado por completo los departamentos de I+D en una gran número de empresas, reduciendo de esta forma la necesidad de personal relacionado con el diseño electrónico, lo que aumenta el nivel de desempleo en el sector y disminuye de forma considerable el nivel salarial y la calidad de vida de estos profesionales.

Sin embargo, es posible realizar una transferencia tecnológica exitosa en el diseño de sistemas digitales con el fin de actualizar a la industria electrónica, proporcionando herramientas para que puedan competir con los productos importados. Para dar solución a este problema se debe trabajar de forma conjunta entre la empresa, la academia y el gobierno; los centros de formación deben incentivar y asesorar a sus egresados en la creación de empresas de base tecnológica que utilicen conocimientos actualizados (ya que tratar de cambiar la mentalidad de los empresarios que llevan muchos años realizando importaciones es muy difícil); se deben actualizar los programas académicos para que los profesionales posean

las habilidades necesarias para concebir, diseñar, implementar y operar sistemas digitales. En [6] se muestra como se modificó el contenido del área de digitales en la Universidad Nacional de Colombia, pasando de un programa puramente académico que finalizaba en la simulación y utilizaba tecnologías propietarias obsoletas, a uno basado en herramientas abiertas [7], [8], [9] centrado en el diseño y en la utilización de metodologías de diseño y tecnologías modernas. Esta experiencia mostró la importancia de la implementación física, en la creación de habilidades que requiere la industria local para reducir el alto grado de dependencia tecnológica y en el cambio de mentalidad para pasar de ser un integrador que utiliza productos comerciales a un diseñador que crea sus propias soluciones.

A. Hardware Copyleft

Al ser inspirado en el movimiento Free Open Source Software (FOSS), los dispositivos *hardware copyleft* comparten la misma filosofía [10], y son su complemento perfecto. Los requisitos para que un dispositivo HW sea reproducible y modificable son: Disponibilidad de los esquemáticos y los archivos de la placa de circuito impreso en un formato que permita el uso de herramientas abiertas; la cadena de herramientas de compilación y depuración para desarrollo de aplicaciones; el código fuente de: el programa que inicializa la plataforma, la herramienta que carga dicho programa en la memoria no volátil, el sistema de archivos y aplicaciones; documentación completa que indique como fué diseñada, construida, como utilizarla, como desarrollar aplicaciones y tutoriales que expliquen el funcionamiento de los diferentes componentes. Adicionalmente, se debe contar con la posibilidad de fabricación y montaje, lo que constituye la principal diferencia entre el software y el hardware libre. Esto contrasta fuertemente con el movimiento de software libre, en donde no se requiere inversión de capital para modificar un proyecto existente. Por esta razón, pueden existir varios niveles de libertad, un proyecto que utilice componentes costosos y de difícil consecución limitará su alcance a un sector determinado.

El uso de hardware copyleft permite reducir de forma considerable la inversión necesaria para la creación de empresas de base tecnológica, ya que no es necesario pagar por costosas licencias software y es posible utilizar proyectos existentes como punto de partida de desarrollos propios ahorrando mucho tiempo y reduciendo de esta forma el *time to market*.

1) *ECB-BF532 y ECBOT Herramientas para la enseñanza de Procesamiento de Señales y de Imágenes*: En este artículo

presentamos dos plataformas *copyleft* hardware: ECB_BF532 basado en el DSP de Analog Devices BF532 y ECBOT [8] [9], compuesto por un procesador ARM9 y una FPGA de la familia Spartan3; estas computadoras en una sola tarjeta (SBC por sus siglas en inglés) permiten: el manejo a bajo nivel de cualquier sensor de imagen CMOS; implementar algoritmos de procesamiento de imagen y de señales en SW, en HW o en una combinación HW/SW y comparar su desempeño; la utilización de librerías abiertas como *opencv* y aplicaciones como *mplayer*, *octave* útiles en el desarrollo de este tipo de aplicaciones. Adicionalmente, la información necesaria para programarla, reproducirla y modificarla se encuentra disponible en el sitio del proyecto *linux EnCaja*¹.

II. PLATAFRORMA ECBOT

La SBC ECBOT [8] esta compuesta por un procesador ARM9 de 200MHz (AT91RM9200 de Atmel) y una FPGA de la familia Spartan3 de Xilinx, soporta la ejecución del sistema operativo Linux, posee 32 MBytes de memoria RAM, 2 Mbytes de memoria Flash serial y soporta la utilización de memorias micro SD de cualquier tamaño; proporciona 8 entradas análogas, 10 digitales, un puerto serie y un puerto USB host. En la figura 1(a) se muestra el diagrama de bloques y en la 1(b) una fotografía de esta plataforma.

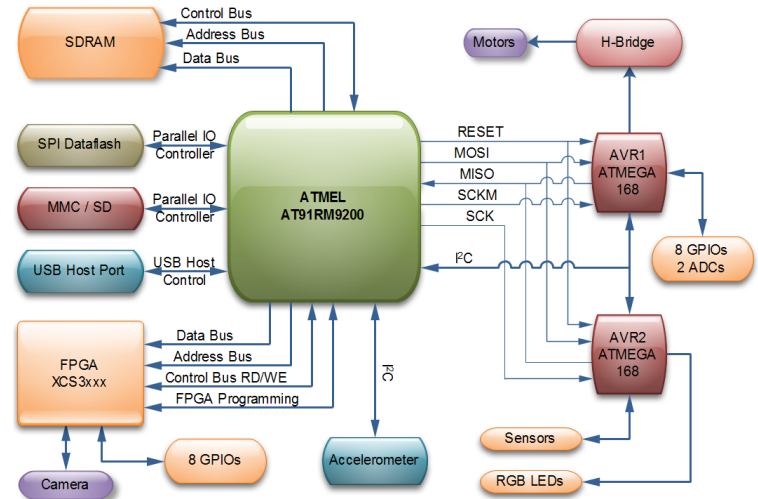
A. Sensor de Imagen

ECBOT permite la conexión de un gran rango de sensores de imagen a través de un conector dedicado para dicho fin; esta interfaz se conecta a uno de los 4 bancos de la FPGA y puede ser alimentado con un voltaje diferente, los otros 3 bancos de la FPGA se alimentan con 3.3V para poder comunicarse de forma adecuada con el procesador ARM9 y el cuarto banco recibe la alimentación de un regulador independiente del valor requerido por el sensor de imagen. En la figura 2 se muestra el esquemático de este conector, en el podemos encontrar las señales básicas para leer la información de un sensor de 8 bits (*D2C-D9C*, *vsync*, *hsync*, *pclk*, *mclk*); adicionalmente, se proporciona acceso al puerto I2C (*CSDL*, *CSDA*), ya que la mayoría de los sensores permite configurar su resolución, contraste, brillo, etc por este medio; y una fuente de alimentación independiente (*UF3*).

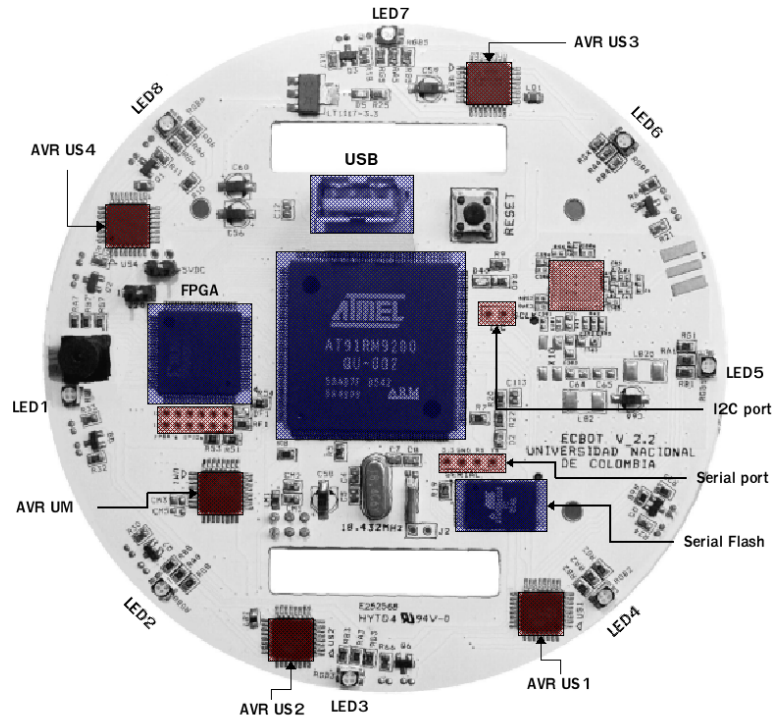
Hasta el momento se han probado con éxito 3 diferentes sensores de imagen en esta plataforma: El KAC-9628 de Kodak, el TCM8240 de Toshiba y el CAM3908.

B. Controlador del Sensor de Imagen

La comunicacion entre el sensor de imagen y el procesador se realiza a través de la FPGA, para esto se implementó un periférico dedicado que comparte el bus de direcciones, datos y control con el procesador; como puede verse en la figura 3 está compuesto por un módulo (*SYNC*) encargado de sincronizar las señales provenientes del procesador con el reloj de la FPGA; el módulo *write pulse generator* genera un pulso de duración de un ciclo de reloj de la FPGA por cada escritura del procesador, esto se hace con el fin de prevenir



(a) Diagrama de Bloques de ECBOT



(b) SBC ECBOT

Fig. 1. Arquitectura de la plataforma ECBOT

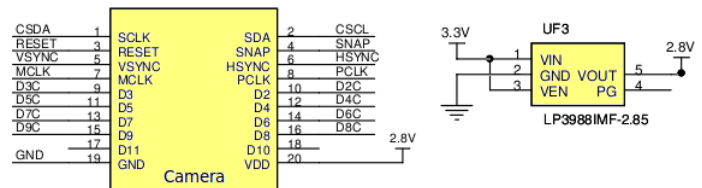


Fig. 2. ECBOT: Interfaz con el sensor de imagen

¹<http://wiki.linuxencaja.net/>

que el controlador ejecute una acción más de una vez debido a la mayor frecuencia de operación; el módulo *image capture* esta encargado de: recibir la información de la imagen y almacenarla en la memoria de doble puerto y generar una señal de interrupción indicando que la información de la imagen se encuentra disponible; y una memoria RAM de doble puerto que permite leer la información de la imagen y transferirla al procesador para su procesamiento.

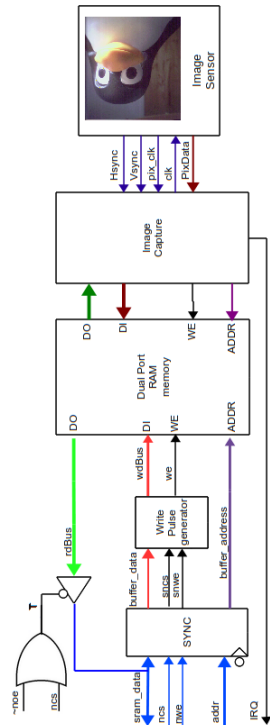


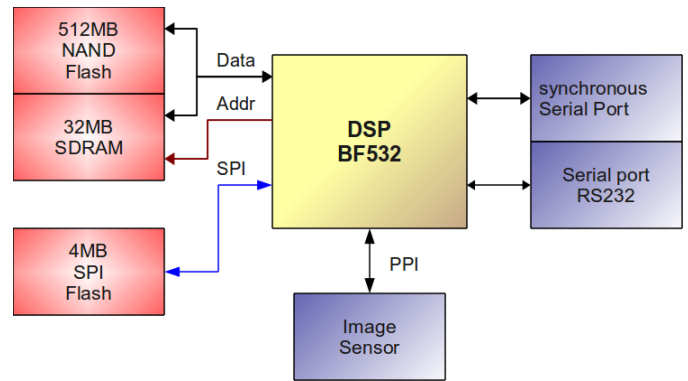
Fig. 3. Arquitectura del módulo encargado de controlar el sensor de imagen en la plataforma ECBOT

III. PLATAFORMA ECB.BF532

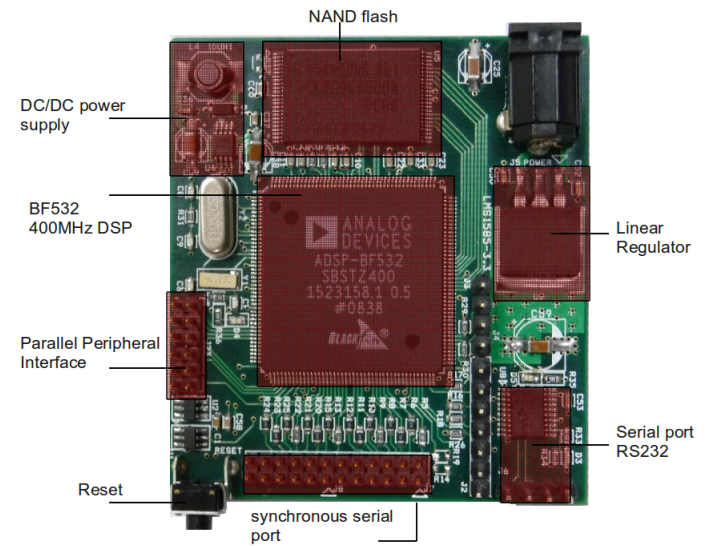
La plataforma *ECB.BF532* está basada en el DSP de Analog Devices *BF532* que corre a 400MHz, posee una memoria SDRAM de 32 Mbytes, una memoria NAND flash de 512 Mbytes utilizada para almacenar datos, una memoria SPI de 4MHZ encargada de almacenar el cargador de Linux (*u-boot*), la imagen del kernel de Linux y un sistema de archivos básico (que puede ser expandido en la memoria NAND) basado en la distribución *uClinux*. En la figura 4(a) se muestra su diagrama de bloques y en la 4(b) su disposición física.

A. Procesamiento de señales

ECB.BF532 permite la conexión de conversores Análogo/Digital y Codecs de Audio utilizando los puertos sincronicos seriales o la interfaz paralela *PPI*. El sistema operativo Linux, que se ejecuta en la SBC, permite el acceso a esta información y facilita el desarrollo de aplicaciones utilizando el language C; el esquemático mostrado en la figura 5 es parte de una aplicación de encriptamiento de voz, donde se utilizó el proyecto abierto OSS (Open Sound



(a) Diagrama de Bloques de ECB.BF532



(b) SBC ECB.BF532

Fig. 4. Arquitectura de la plataforma ECB.BF532

System) para controlar el codec de audio AD1836A; al existir soporte en Linux para este tipo de dispositivos, se simplifica su uso, centrando el esfuerzo en la aplicación final y no en detalles de bajo nivel, todo el manejo se realiza a través del archivo */dev/dsp*, asociado al controlador del dispositivo; dicho controlador proporciona funcionalidades básicas para abstraer el manejo del codec haciendo más fácil su uso.

B. Procesamiento de Imagen

De forma similar, Linux permite el control de bajo nivel de sensores de imagen proporcionando un controlador (*/dev/video0*) que realiza una abstracción del manejo de bajo nivel del sensor de imagen, (este controlador hace parte del proyecto *video4linux*) permitiendo obtener información y configurar el sensor de forma fácil. A diferencia de la SBC *ECBOT* la información de la imagen es almacenada directamente en la memoria del DSP, lo que hace más rápido el proceso de captura.

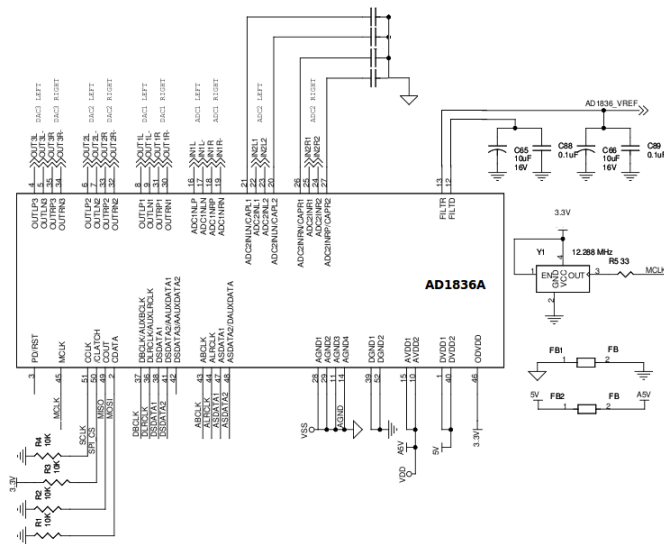


Fig. 5. Conexión del CODEC de audio AD1836A

IV. IMÁGENES CAPTURADAS Y EJEMPLO DE EJECUCIÓN DE ALGORITMOS

En la figura 6 se muestran algunas imágenes capturadas con los sensores de imágenes mencionados anteriormente, los cuales envían la información sin formato (*raw*) y un programa que se ejecuta en las plataformas las convierte al formato deseado (*PNG en este caso*). En las dos plataformas se obtiene la misma calidad de imagen, ya que se utilizan los mismos sensores.

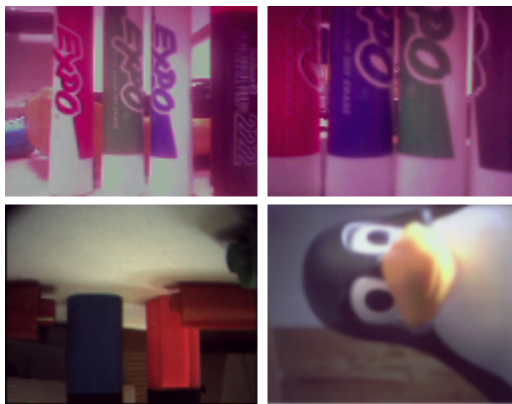


Fig. 6. Imágenes capturadas en las SBC ECB.BF532 y ECBOT

Se ejecutó un algoritmo sencillo para detección de colores basado en vecindades para probar la capacidad de las plataformas, este algoritmo tiene como entrada una imagen similar a la que se muestra en la figura 6 y su función es detectar un determinado grupo de colores (*blobs*) que reflejan el estado interno de un robot móvil, el resultado de la ejecución de este algoritmo se puede ver en la figura 7

V. CONCLUSIONES Y TRABAJO FUTURO

Se presentaron dos plataformas abiertas para aplicaciones en procesamiento de señales e imágenes; que permiten ser

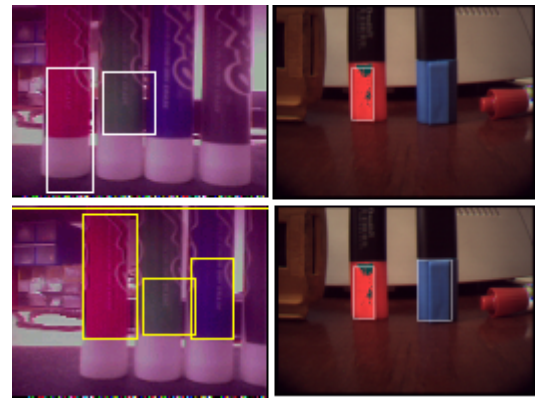


Fig. 7. Aplicación de un algoritmo de detección de *blobs*

estudiadas, programadas, reproducidas y modificadas, proporcionando un punto de partida para el desarrollo de aplicaciones comerciales y ser utilizadas en cursos relacionados con el diseño digital y procesamiento de imágenes de señales. A diferencia de plataformas existentes, ECBOT y ECB.BF532 permiten el estudio total de su arquitectura, desde el componente físico (esquemático, placa de circuito impreso) hasta su programación; eliminando el problema de realizar diseños a alto nivel sin tener en cuenta la arquitectura del dispositivo.

El hardware *copyleft* representa un ahorro en tiempo y dinero en el desarrollo de aplicaciones comerciales, ya que proporciona diseños funcionales y conocimiento sobre su programación y funcionamiento. Por otro lado, la utilización de aplicaciones abiertas que hacen parte del proyecto FOSS, permite su programación de forma fácil abstrayendo el manejo tedioso a bajo nivel del hardware, permitiendo su uso a diferentes niveles de formación. La unión hardware *copyleft* y software libre permite reducir de forma considerable la inversión necesaria para el desarrollo de productos tecnológicos comerciales; el objetivo principal de estos movimientos es la difusión de conocimientos a todos los interesados, convirtiéndolos en una herramienta poderosa en la transferencia de tecnología y de conocimientos.

El desarrollo de plataformas hardware es de vital importancia en países en vía de desarrollo ya que posibilita la creación de habilidades necesarias para la creación de empresas de base tecnológica, lo que puede generar una demanda creciente en productos, bienes y servicios relacionados con la manufactura de dispositivos digitales.

Estas plataformas serán utilizadas en los cursos de DSP y procesamiento de imágenes en el Departamento de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Colombia sede Bogotá.

VI. BIBLIOGRAFÍA

REFERENCES

- [1] C. Camargo. Metodología Para la Transferencia Tecnológica en la Industria Electrónica Basada en Software Libre y Hardware Copyleft. *XVII Workshop de Iberchip, Bogotá Colombia*, February 2011.
- [2] M. Tovar and R. Rodríguez. PROSPECTIVA Y VIGILANCIA TECNOLÓGICA DE LA ELECTRÓNICA EN COLOMBIA. Master's thesis, Universidad Nacional de Colombia, 2007.

- [3] D Zuluaga, S Campos, M Tovar, R Rodríguez, J Sánchez, A Aguilera, L Landínez, and J Medina. Informe de Vigilancia Tecnológica: Aplicaciones de la Electrónica en el Sector Agrícola. Technical report, COLCIENCIAS, 2007.
- [4] M. Duque and A. Gauthier. Formación de Ingenieros para la Innovación y el Desarrollo Tecnológico en Colombia. *Revista de la Facultad de Minas - Universidad Nacional de Colombia*, December 1999.
- [5] Héctor Martínez. Apropiación de conocimiento en Colombia. El caso de los contratos de importación de tecnología. *Revista Cuadernos de Economía*, 2004.
- [6] C. Camargo. Hardware copyleft como Herramienta para la Enseñanza de Sistemas Embebidos. *Simposio Argentino de Sistemas Embebidos*, 2011.
- [7] C. Camargo. SIE: Plataforma Hardware copyleft para la Enseñanza de Sistemas Digitales. *XVII Workshop de Iberchip, Bogotá Colombia*, February 2011.
- [8] C. Camargo. ECBOT: Arquitectura Abierta para Robots Móviles. *IEEE Colombian Workshop on Circuits and Systems*, 2007.
- [9] C. Camargo. ECBOT y ECB_AT91 Plataformas Abiertas para el Diseño de Sistemas Embebidos y Co-diseño HW-SW. *VIII Jornadas de Computación Reconfigurable y Aplicaciones*, 2008.
- [10] R. Stallman. Philosophy of the GNU project. URL: <http://www.gnu.org/philosophy/>, 2007.