

CURSO DE ACTUALIZACIÓN

EN HERRAMIENTAS MODERNAS PARA DISEÑO y FABRICACIÓN DE DISPOSITIVOS DIGITALES

Carlos Iván Camargo Bareño

28 de septiembre de 2012

PROPUESTA INICIAL

Linux Como herramienta de Diseño para Sistemas Embebidos AUTHOR: C. Camargo
E-MAIL: carlos@unal.edu.co

Copyright ©2012 Universidad Nacional de Colombia.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, version 1.2, with no invariant sections, no front-cover texts, and no back-cover texts. A copy of the license is included in the end.

This document is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.

Índice general

1.1. Motivación	1
1.1.1. Tamaño del Mercado	2
1.1.2. Herramientas de Desarrollo	3
1.1.3. Plataformas de desarrollo copyleft hardware	5
1.2. Objetivos	11
1.2.1. Generales	11
1.2.2. Específicos	11
1.3. Metodología	12
1.4. Duración	12
2. CONTENIDO	14
3. Información General	16
3.1. Recursos	16
3.2. Requisitos	16
3.3. Material didáctico	16

1.1. Motivación

Los avances en el área de la electrónica están generando una revolución cuyas implicaciones sociales y económicas son difíciles de estimar. Los aspectos de esta revolución están asociados con la base tecnológica, la relación laboral, la estructura industrial e incluso hasta las relaciones internacionales. Los computadores embebidos se están convirtiendo en omnipresentes y son utilizados en un amplio espectro de aplicaciones, que van desde juegos y dispositivos móviles hasta aplicaciones automovilísticas, de aviación etc. Muchos de ellos toman la forma de artefactos que utilizamos en nuestra vida diaria (herramientas, vestuario, electrodomésticos, etc) pero mejorados con sensores, actuadores, procesadores y software embebido. Las industrias han visto como se muestran signos de recesión en los mercados tradicionales, por lo que la aparición de estos sistemas embebidos en chips y software nace como una buena alternativa económica. El análisis de los procesos de adopción de los dispositivos tecnológicos de hoy muestra que la incorporación en el mercado de nuevos dispositivos resulta en la alteración o la creación de nuevos hábitos.

Un aspecto importante del avance de la industria electrónica es el desarrollo de la industria de los semiconductores desde la creación del transistor en 1947. La evolución de la tecnología continúa en el presente, donde cada vez se producen dispositivos más económicos, veloces y densos. En la actualidad el mercado de los sistemas embebidos es el que reporta mayor cantidad de unidades vendidas, seguido de lejos, por el de los Computadores Personales y por último el de las Estaciones de trabajo (ver Fig 1.1). La gran difusión de los sistemas embebidos ha sido en gran parte debida a los desarrollos en el área de los procesadores, y a la disponibilidad de Sistemas Operativos (OS) como Linux, eCos y Windows CE.

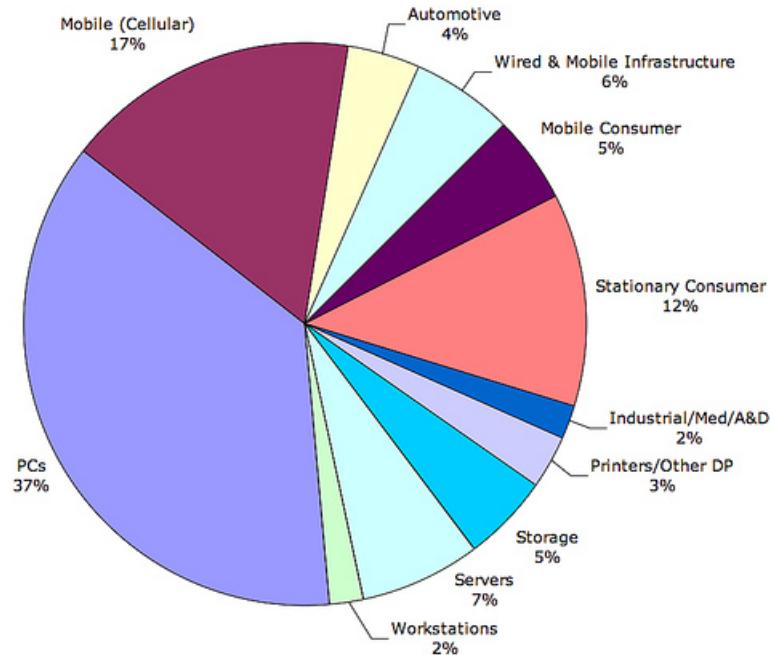


Figura 1.1: Mercado de Procesadores. Fuente: Venture Development Corporation (VDC)

1.1.1. Tamaño del Mercado

Como puede verse en la figura 1.1 existe una gran porción del mercado que requiere soluciones diferentes a las “tradicionales” (Software ejecutándose en PCs). En este tipo de aplicaciones se requiere un sistema con capacidad de computo que realice una función determinada de forma eficiente sujeta a restricciones de consumo de potencia, tamaño y costo. Siendo estos sistemas llamados sistemas embebidos. La figura 1.2 muestra algunas aplicaciones comerciales de estos dispositivos. A continuación se enumeran unas cifras que dan una idea de la magnitud del mercado de los Sistemas Embebidos:

Domestic	Business	Infrastructure
• PDAs / GPS / Games	• Communications	• Telecommunications
• Mobile phones	• Robots	• CCTV & security
• MP3 players / iPods	• Automation	• Tolling
• Cameras	• Photocopiers	• Information signs
• TV / Sound	• Medical devices	• Speed limit signs
• Smart appliances	• Instrumentation	• Signalling
• Security systems	• Diagnostic systems	• Navigation
• Computer peripherals	• Billboards	• Enforcement
• Climate controllers	• ATMs	• Ticketing
• Automobiles	• Vending	• Radar
		• Traffic Control

Figura 1.2: Algunas aplicaciones de los Sistemas Embebidos

- 385 millones de microprocesadores se construyeron en el 2000 para uso en computadores convencionales.
- 6.4 billones de microprocesadores se construyeron en el 2000 para uso en dispositivos Embebidos.
- El desarrollo de Software Embebido es una industria mundial multi-billionaria.
- El mercado de los sistemas operativos embebidos y herramientas de desarrollo fue de más de US\$1.3 billones en 2001 y continúa en aumento.
- En 2003 la mitad de los diseños incorporaron procesadores de 32 y 64 bits que utilizaban Linux.
- 98 % de los procesadores fabricados se encuentran en un Sistema Embebido.
- Se estiman un total de 16 billones de procesadores en el 2010, aprox 3 por persona.
- Los sectores con mayor crecimiento son las aplicaciones en automóviles y la electrónica de consumo.

Sin embargo, a diferencia de otras industrias, no son necesarias grandes inversiones de dinero para generar aplicaciones comerciales, ya que como se puede ver en la Figura 1.3, donde cerca del 60 % de los desarrolladores de sistemas embebidos trabajan en compañías con menos de 10 desarrolladores.

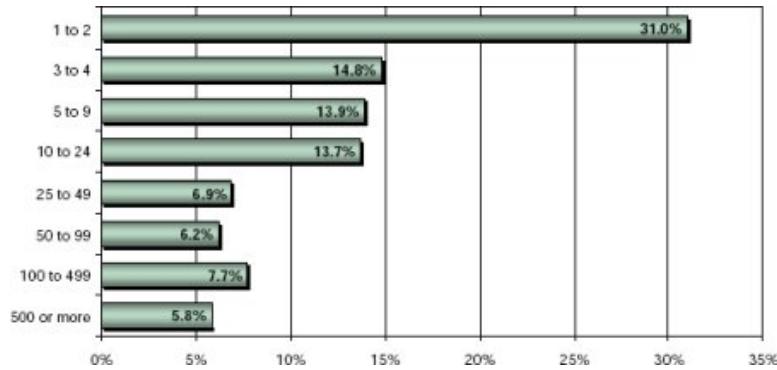


Figura 1.3: Número promedio de desarrolladores por compañía. Fuente: Venture Development Corp.

1.1.2. Herramientas de Desarrollo

Existe una gran variedad de herramientas Hardware y Software para el diseño de Sistemas Embebidos. En el componente hardware gracias a los avances de la industria de los semiconductores, y en particular el nivel de integración, existen dispositivos que integran una Unidad de Procesamiento Central con una gran cantidad de periféricos en un solo Circuito Integrado (SoC), que cubren una gran cantidad de áreas de aplicación, entre las cuales encontramos:

- Comunicaciones:
 - Controladores para comunicaciones I2C, SPI, USB, RS232.
 - Controladores para red ethernet y WiFi.
- Aplicaciones Multimedia:
 - Audio: Codecs de Sonido.
 - Video: Controladores para LCD, controladores de sensores de imagen.
- Almacenamiento:
 - Memorias volátiles: RAM, SDRAM.
 - Memorias no volátiles: Flash, Nand Flash, Compact Flash, Secure Digital.
- Procesamiento:
 - Unidades de Punto Flotante.
 - Encripción.

El componente software por su parte debe ser capáz de manejar estos periféricos, y proporcionar funciones que permitan realizar aplicaciones con un nivel de abstracción adecuado. Actualmente existen una gran cantidad de herramientas comerciales y de código abierto disponibles para ser usadas con Sistemas Embebidos, siendo las más populares las suministradas por la comunidad, tal como se puede observar en la Figura 1.4

En la Figura 1.5 se muestra una comparación de la utilización de los sistemas operativos más populares para el diseño de Sistemas Embebidos (comerciales y de libre distribución). Como puede verse el

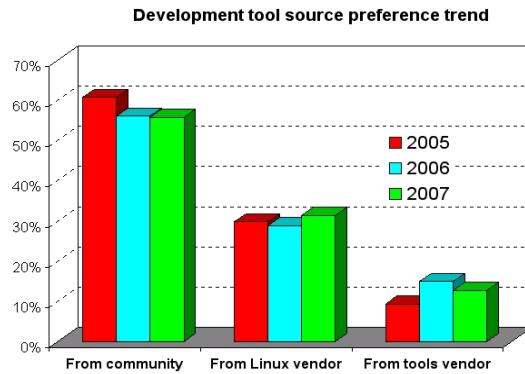


Figura 1.4: Preferencias de herramientas para Sistemas Embebidos

sistema operativo Linux es el más utilizado, seguido de lejos por Windows CE. Esto obedece al carácter libre de Linux, a la facilidad de realizar drivers (lo cual es muy importante en este tipo de dispositivos ya que pueden integrar periféricos diseñados especialmente para la aplicación) y al gran soporte existente a través de un gran número de listas de discusión y a la gran cantidad de aplicaciones y bibliotecas disponibles. Esta preferencia por Linux no solo se ve en pequeñas industrias, en la actualidad observamos grandes empresas como Motorola y Nokia que utilizan Linux como sistema Operativo para sus productos.

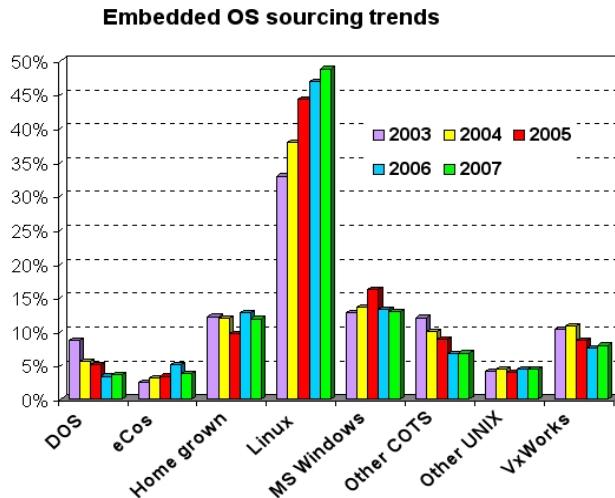


Figura 1.5: Preferencias de herramientas para Sistemas Embebidos

Como conclusión se puede decir que el mercado de los Sistemas Embebidos mueve una gran cantidad de dinero, proporciona una alternativa a las aplicaciones tradicionales, cubre una gran variedad de aplicaciones, lo cual hace ver que trabajar en el desarrollo de este tipo de sistemas sea atractivo desde el punto de vista económico. Por otro lado, se encuentran una gran variedad de herramientas Hardware y Software que facilitan su diseño; gracias al movimiento de Software libre y al bajo costo de los Dispositivos semiconductores. Así, es posible que pequeños grupos de trabajo desarrollen aplicaciones comerciales con pequeñas inversiones de capital.

1.1.3. Plataformas de desarrollo copyleft hardware

Un ejemplo de difusión en transferencia tecnológica se ha venido dando hace 10 años con la creación del portal Linuxencaja, creado por Andrés Calderon y Carlos Camargo. Este portal posee la información necesaria para reproducir, modificar las siguientes plataformas de desarrollo:

Plataformas de desarrollo

La primera plataforma desarrollada utiliza el SoC AT91R40008 de ATME. El diagrama de bloques de la plataforma y la foto del prototipo se muestra en la figura 1.6, posee componentes de montaje superficial siendo este el punto de partida para el estudio de técnicas de fabricación de placas de circuito impreso, y con la cual se realizaron aplicaciones *standalone* y con el sistema operativo eCos. Dentro del proceso de diseño de sistemas digitales, algunas de las tareas deben ser implementadas en hardware para cumplir con los requerimientos temporales; por este motivo es importante realizar implementaciones de tareas hardware en dispositivos lógicos programables (Programmable Logic Device (PLD)s). Para estudiar la forma de comunicar estas tareas hardware con Systen On a Chip (SoC) comerciales se diseñó un clon de la tarjeta comercial XPORT de la empresa norteamericana Charmedlabs, que permite conectarse con el procesador de la plataforma comercial ©GameBoy Advance de Nintendo [1] (ver figura 1.7).

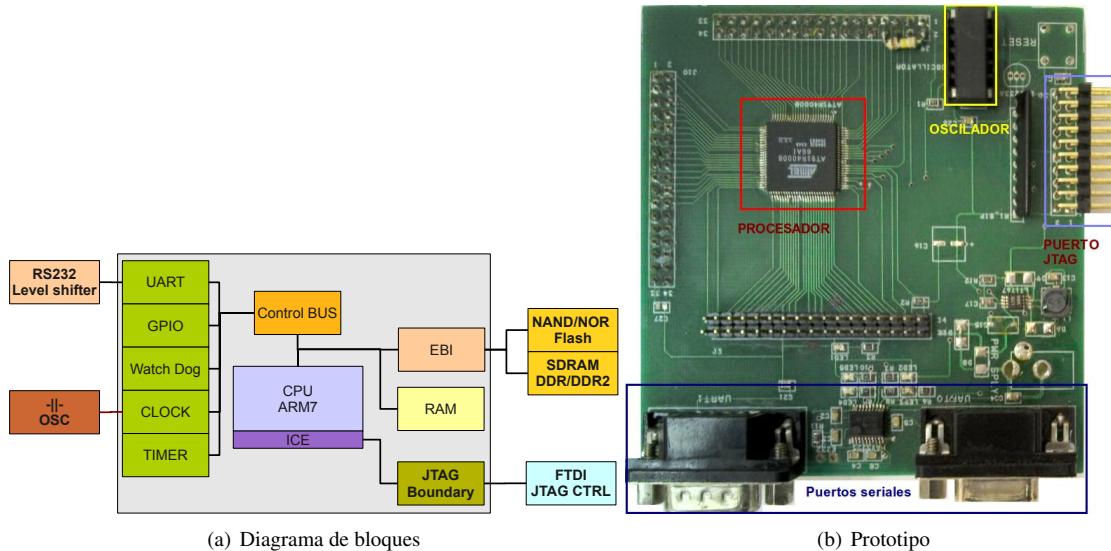


Figura 1.6: Plataforma de desarrollo ECB_ARM7

El uso de Linux como herramienta de desarrollo ha aumentado en los últimos años, llegando a ser el sistema operativo más utilizado por los desarrolladores; empresas como Nokia, Motorola, Google, Dlink, Hewlett Packard utilizan Linux para desarrollar el firmware de sus dispositivos; por este motivo, se diseñaron plataformas que soportan la ejecución de este sistema operativo. Se diseñó una familia de plataformas denominadas ECB_AT91 V1 (ver figura 1.8 [2] [3]), ECB_AT91 V2 (ver figura 1.9) y ECBOT (ver figura 1.10 [4] [5] [6]). Esta familia de plataformas se encuentran registradas en el kernel oficial de Linux, por lo tanto, su soporte queda garantizado en futuras versiones del kernel. ECB_AT91 V1 fue el punto de partida para el estudio de técnicas de fabricación de placas de circuito impreso para SoC que operan a velocidades mayores de 180MHz, y para aprender el proceso completo de adaptar el kernel de Linux a una plataforma nueva. En la figura 1.10 (a) se muestra el diagrama de bloques de estas tres arquitecturas que

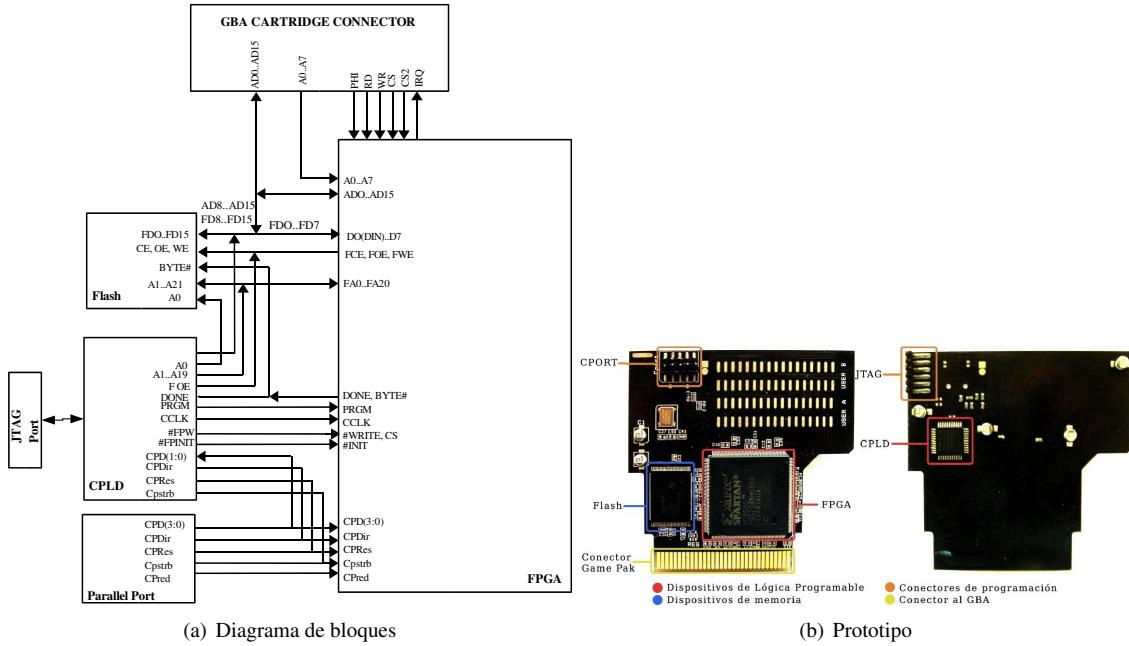


Figura 1.7: Plataforma de desarrollo UNAL_UIS_XPORT

poseen el SoC de 180MHz de Atmel AT91RM9200. Debido a que este SoC no posee conversores análogos - digitales internos fue necesario incluir un micro-controlador de 8 bits que se comunica con el SoC vía Inter-Integrated Circuit (I2C), el SoC realiza la programación del microcontrolador emulando un puerto paralelo en sus pines de entrada/salida de propósito general (General Purpose Input/Output (GPIO)s) que van conectados a los pines del puerto In System Programming (ISP) del microcontrolador.

La plataforma ECB_AT91 V2 es el resultado del interés de utilizar dispositivos lógicos programables y SoC comerciales para crear periféricos dedicados y la creación de nuevos drivers de Linux que los controlaran, su arquitectura es idéntica a la de la plataforma ECB_AT91 V1 pero el bus de datos, dirección y control del SoC se conectan a la Field Programmable Gate Array (FPGA) para permitir la comunicación con los periféricos implementados en ella; la FPGA es configurada por el SoC utilizando unos GPIOs que van conectados al puerto Joint Test Action Group (JTAG) de la FPGA, lo que elimina la necesidad de cables adicionales de conexión, permitiendo su re-programación total de forma remota.

La plataforma ECBOT fue diseñada con el propósito de iniciar el estudio en robótica móvil y procesamiento de imagen; para esto, se utilizó una arquitectura similar a la de la plataforma ECB_AT91_V2 adicionando una conexión dedicada a un sensor de imagen y circuitos especializados para el control de sensores (6 sensores de choque, 1 sensor de imagen, movimiento de los motores basado en BEMF¹), actuadores (8 Light-Emitting Diode (LED)s RGB, 2 motores DC) y un convertidor DC/DC que garantiza el máximo uso de las baterías. Como parte de este trabajo, se realizó la adaptación de proyectos libres que facilitan la investigación en robótica como *player/stage*² y *openblocks*³. En la actualidad ECBOT está siendo usado por los grupos de robótica de los departamentos de Sistemas y Mecatrónica de la Universidad Nacional de Colombia.

¹Back Electro Magnetic Force

²<http://playerstage.sourceforge.net/>

³<http://dspace.mit.edu/handle/1721.1/41550>

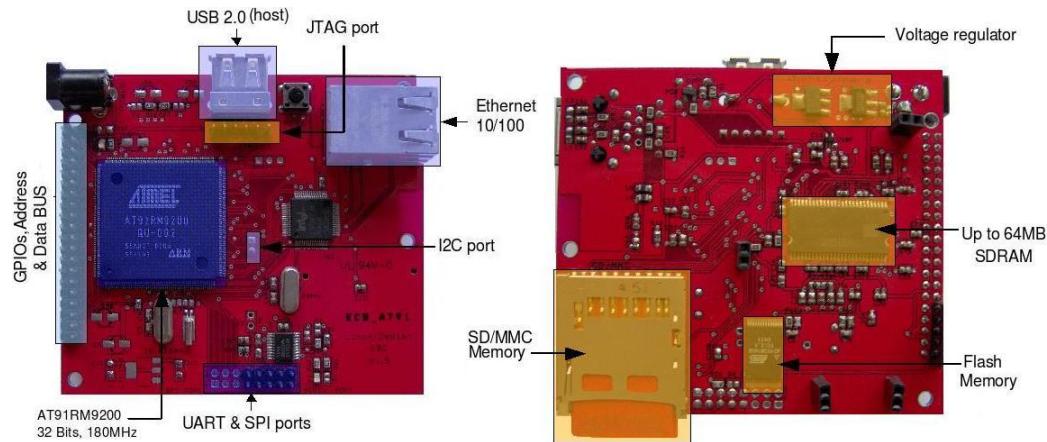


Figura 1.8: Plataforma de desarrollo ECB_AT91 V1

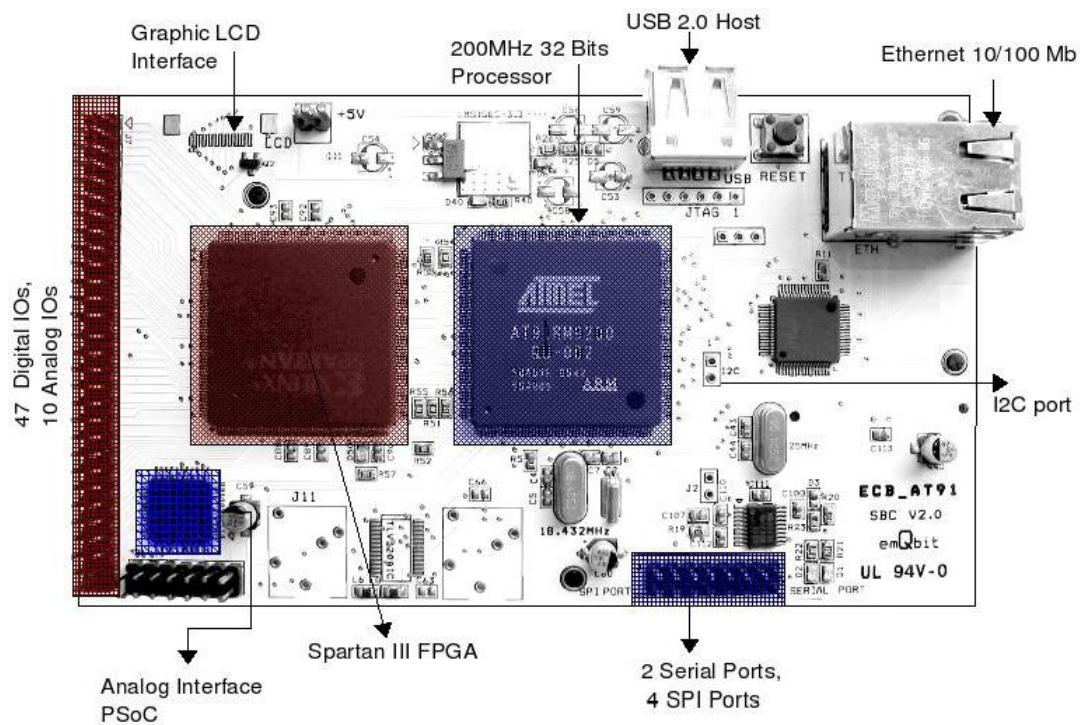


Figura 1.9: Plataforma de desarrollo ECB_AT91 V2

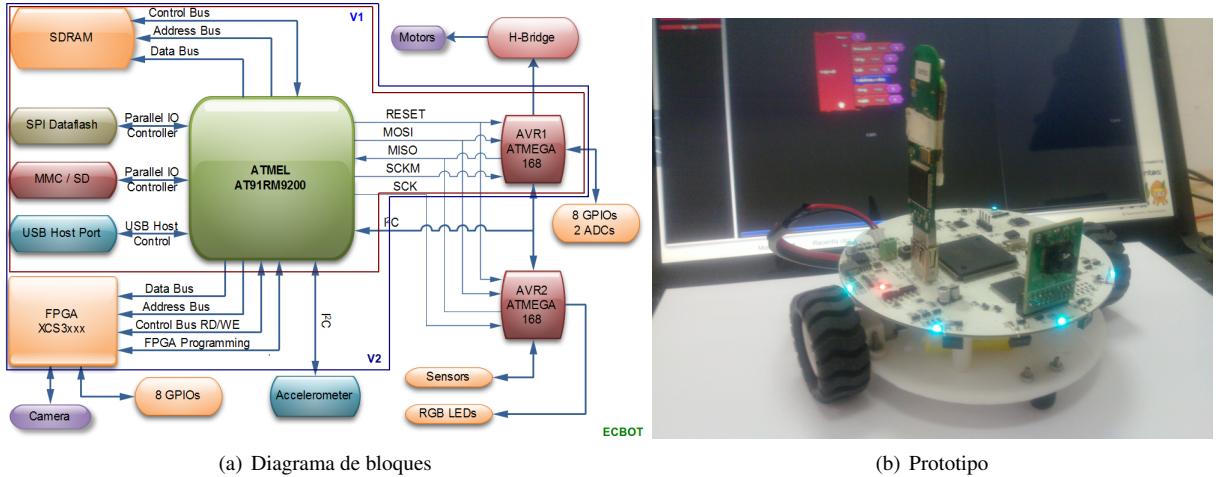


Figura 1.10: Plataforma de desarrollo ECBOT

Los SoC utilizados en las plataformas anteriores no pueden ser utilizados en aplicaciones de procesamiento de señales o de imágenes en tiempo real (a menos que se implemente algún periférico en la FPGA), ya que su arquitectura no está diseñada para este fin. Los Digital Signal Processor (DSP)s son procesadores especializados para ejecutar las operaciones más comunes en procesamiento digital de señales. Pensando en este tipo de aplicaciones se diseñó la plataforma ECB_BF532 [7] que utiliza el DSP BF532 de la familia Blackfin de Analog Devices. Esta plataforma es utilizada en los cursos de procesamiento digital de señales del Departamento de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Colombia.

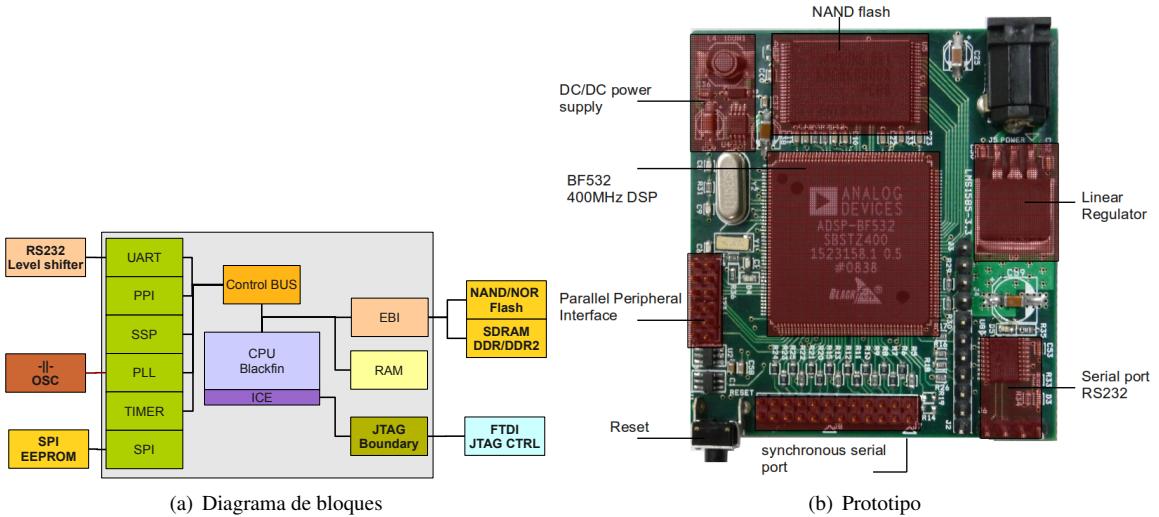


Figura 1.11: Plataforma de desarrollo ECB_BF532

La plataforma SIE[8], permite el desarrollo de aplicaciones gráficas y esta compuesta por un procesador MIPS fabricado por la compañía China Ingenic y por una FPGA (Spartan 3E de Xilinx) lo que posibilita la implementación de tareas hardware y la creación de controladores específicos en el sistema operativo Linux. En la actualidad esta pataforma esta siendo utilizada en tres de los cuatro cursos de la línea

de electrónica digital en el Departamento de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Colombia.

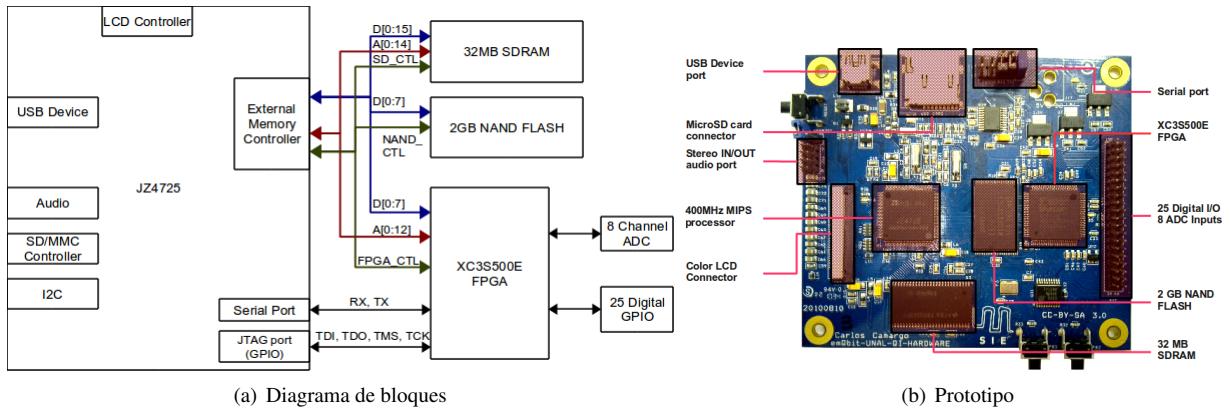


Figura 1.12: Plataforma de desarrollo SIE

La plataforma STAMP (se encuentra en proceso de pruebas al momento de escribir este documento) es el resultado del interés por el estudio del sistema operativo Android. Debido a los requerimientos de memoria de este sistema operativo, las plataformas anteriormente diseñadas no pueden ejecutar aplicaciones Android. Por este motivo se utilizó el SoC de Freescale iMX233 que incluye el controlador de memorias Double Data Rate (DDR) (las que son más económicas y de mayor densidad que las Synchronous dynamic random-access memory (SDRAM) utilizadas en las plataformas existentes); adicionalmente, este SoC permite inicializar desde una memoria Secure Digital (SD), lo que elimina la necesidad de utilizar memorias NAND, reduciendo de forma considerable el costo. En la figura 1.13 se puede observar el diagrama de bloques y una foto del prototipo de esta plataforma.

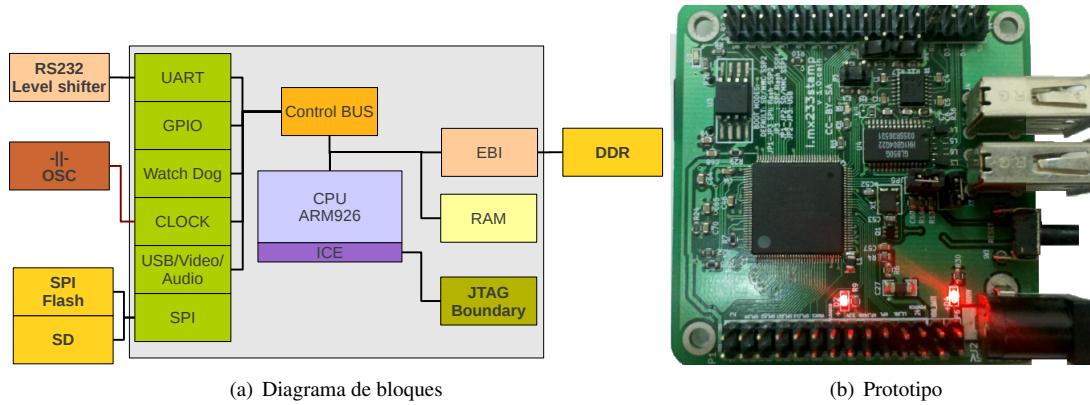


Figura 1.13: Plataforma de desarrollo STAMP

Técnicas de fabricación

Durante el proceso de concepción, diseño, implementación y operación de estas plataformas se trabajó con diferentes proveedores de bienes y servicios nacionales y extranjeros, y se utilizaron diferentes

técnicas para su construcción y montaje. En la tabla 1.1 se resumen las principales características de cada plataforma: Central processing unit (CPU) utilizada, número de capas del Printed Circuit Board (PCB), tipo de montaje: manual o automático, cantidad de unidades producidas, sistema operativo (Operating System (OS)) estudiado y adaptado totalmente a la plataforma y finalmente las universidades que utilizan estas plataformas.

Cuadro 1.1: Características de las plataformas de desarrollo concebidas, diseñadas e implementadas

Plataforma	CPU	Capas	Montaje	Cant.	OS	Usuario
ECB_ARM7	ARM7,33M	2	local Manual.	2	eCos	UN
UN_UIS_XPORT	ARM7,50M	2	local Manual.	2	eCos	UN, UIS
ECB_AT91_V1	ARM920,180M	2	local Manual/Autom.	100	Linux	UN, UIS, ULA, ENAP, UDFJC, USTA
ECB_AT91_V2	ARM920 180M	4	local Manual.	30	Linux	UN, UIS, ULA, ENAP, UDFJC
ECBOT	ARM920 180M	4	local Manual.	20	Linux	UN, UIS
ECB_BF532	Blackfin 400M	4	local Manual.	5	uCLinux	UN
SIE	MIPS32 300M	2	externo Autom.	80	Linux	UN, UIS, ULA, ECI
STAMP	ARM926 454M	2	local Manual.	2	Android	UN

Proceso de diseño y fabricación

La figura 1.14 muestra los pasos que se siguieron en la elaboración de las plataformas de desarrollo; esta metodología puede ser utilizada para la realización de cualquier nivel de producción; el montaje de prototipos puede realizarse de forma manual o automática, dependiendo de los recursos económicos, de la disponibilidad de equipo y de personal especializado.

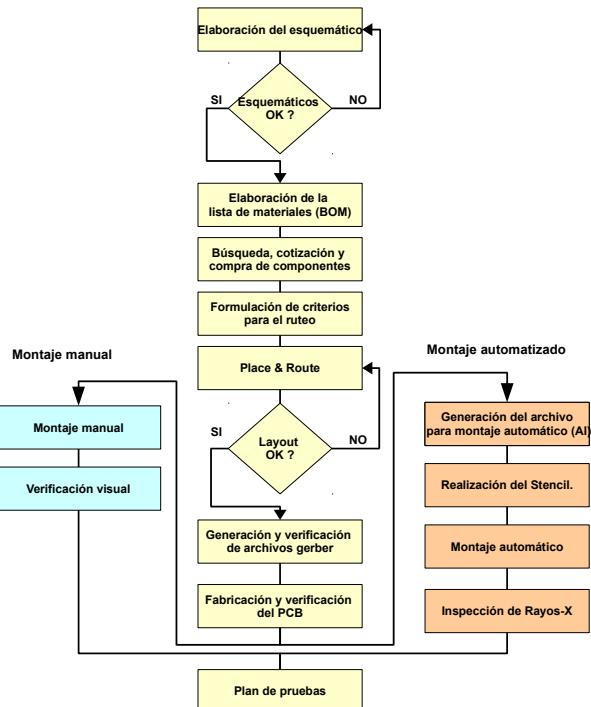


Figura 1.14: Flujo de diseño para las placas de circuito impreso

En esta etapa se adquirieron las siguientes habilidades:

- Concepción, diseño, implementación y operación de tarjetas electrónicas que utilizan SoC de 32 bits.
 - Técnicas de diseño de placas de circuito impreso.
 - Técnicas de montaje manual y automático.
 - Metodología para el diseño e implementación de tarjetas electrónicas.
 - Diseño de protocolo de pruebas.
- Adaptación del sistema operativo Linux a una nueva plataforma.

1.2. Objetivos

1.2.1. Generales

El objetivo principal de este curso es el estudio de Metodologías de diseño que permitan utilizar los avances tecnológicos actuales para dar soluciones a los siguientes problemas:

- Dependencia hacia proveedores externos: La mayoría de las aplicaciones son creadas por empresas externas que no suministran ninguna información sobre su funcionamiento, ni suministran la forma de modificar el funcionamiento inicial. Como consecuencia de esto los módulos existentes o no funcionan o funcionan de forma incorrecta.
- Desactualización tecnológica y baja calidad de los circuitos digitales: Una gran cantidad de módulos utilizan procesadores de 8 bits con limitada capacidad de comunicaciones lo que impide conocer su estado de operación de forma remota; esto, debido al escaso poder de computo y a las herramientas de bajo nivel utilizadas. Por otro lado, los circuitos impresos utilizados son de muy baja calidad (de una capa, sin capas de silkscreen o antisolder) y no cumplen con las normas mínimas para ser utilizados durante largos períodos de tiempo.
- Sobre-dimensionamiento: Se utiliza una gran cantidad de computadores personales para realizar tareas que podrían ser implementadas por un sistema embebido.
- Disminución de la producción local: Los diseñadores nacionales no diseñan sus placas de circuito impreso y dependen de productos provenientes de Asia; los escasos productos nacionales son realizados sin cumplir normas internacionales y con técnicas de fabricación rudimentarias.

1.2.2. Específicos

- Entender el flujo de diseño de sistemas digitales.
- Aprender a utilizar herramientas abiertas para realizar el flujo de diseño completo desde la concepción hasta la implementación física.
- Entender la arquitectura de los sistemas digitales modernos.
- Implementación de tareas hardware y software en dispositivos semiconductores modernos.
- Estudio de los requerimientos hardware y software para la ejecución de Linux en un sistema Embebido con limitaciones de recursos (Velocidad de procesamiento, memoria).
- Realizar drivers de Linux que permitan la interfaz con dispositivos Hardware externos.

- Aprender a crear una imagen del kernel a la medida para una aplicación embebida.
- Crear Sistemas de archivos optimizados para una aplicación.
- Aprender el uso de librerías gráficas para implementación de aplicaciones.
- Apropiar técnicas de diseño de circuitos impresos cumpliendo normas de calidad.
- Entender el proceso de fabricación y montaje de placas de circuito impreso.
- Entender cuales son los puntos a tener en cuenta en el montaje a gran escala de placas de circuito impreso.

1.3. Metodología

Debido a la naturaleza del tema, el curso tendrá un componente teórico en el que se darán las bases conceptuales del tópico a tratar, posteriormente se realizarán laboratorios prácticos con el fin de afianzar los conocimientos. Las prácticas se realizarán con la tarjeta de desarrollo stamp (ver Figura 1.15).

1.4. Duración

El curso tendrá una duración de 100 horas.

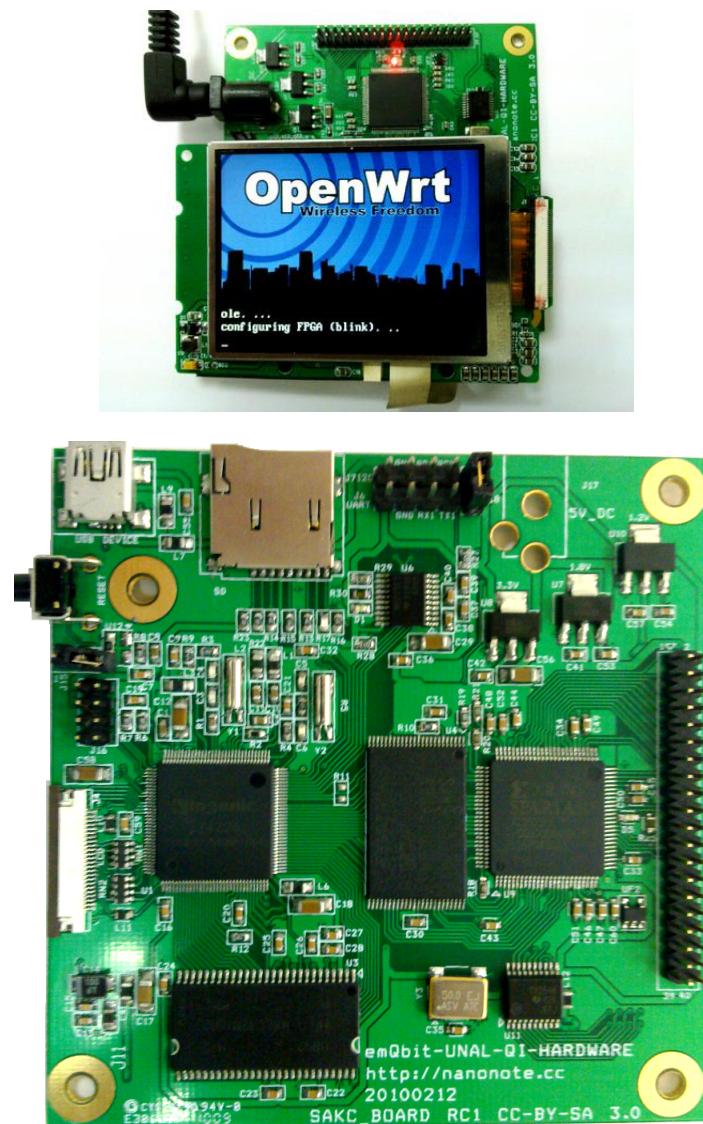


Figura 1.15: Plataforma de Desarrollo stamp

Capítulo 2

CONTENIDO

■ Sistema Operativo Linux

- El sistema Operativo Linux
- Operaciones básicas en Linux
 - Instalación de aplicaciones
 - Variables de entorno
 - Comandos básicos.

■ Sistemas Embebidos – Herramientas GNU

- ¿Qué es un sistema Embebido?
- Flujo de diseño de un Sistema Embebido.
 - Tareas Hardware
 - Tareas Software

■ Máquinas de Estado Algorítmicas (ASM)

- Implementación de tareas hardware usando máquinas de estado algorítmicas.
 - Multiplicador binario implementado en una ASM.
 - Divisor binario implementado en una ASM.
- Dispositivos lógicos programables.
- Lenguajes de descripción de hardware verilog y VHDL.
- **Práctica I: Ejemplos de implementación**

■ Sistemas sobre silicio (SoC)

- Implementación de tareas software usando SoCs.
 - Arquitectura del Procesador MICO32.
 - Arquitectura del SoC LM32.
 - Programación del SoC LM32.
 - ◊ Interfaz con los periféricos.
 - ◊ Herramientas GNU

- **Práctica II: Ejemplos de implementación**

- **Aplicaciones utilizando SoC comerciales**

- Plataforma de Desarrollo stamp.
 - Arquitectura del procesador iMX233.
 - Tarjeta de Desarrollo STAMP
 - Implementación del Programa: Hello World
 - Implementación del Programa: Blinker
- **Práctica III: Ejemplos de implementación sobre el iMX233 utilizando herramientas GNU**

- **Detalles del Sistema Operativo Linux**

- Kernel overview
- Requerimientos HW/SW
- Estructura del código fuente.
- Proceso de inicio de linux
- Drivers de Dispositivos
- Módulos del Kernel
- **Práctica IV: Configuración, compilación e inicio de un kernel “a la medida” utilizando la interfaz de red para descargarlo a la memoria SDRAM**

- **Sistema de Archivos y Módulos del Kernel**

- Estructura de un sistema de archivos.
- Archivos Importantes: Archivos de configuración, archivos de dispositivos.
- Sistemas de archivos disponibles: Debian, Buildroot, Openembedded.
- ¿Qué es un módulo?
- Diferencias entre el espacio de usuario y el espacio de kernel.
- Estructura de un módulo.
- **Práctica IV: Configuración, compilación de 2 sistemas de archivos: Debian y Buildroot, utilizando NFS**
- **Práctica V: Creación de un Módulo Sencillo**

- **Diseño, fabricación y montaje de placas de circuito impreso**

- XXXXX
- XXXXX
- XXXXX
- XXXXX
- XXXXX

Capítulo 3

Información General

3.1. Recursos

Para la realización del curso es necesario contar con la siguiente infraestructura:

- Un PC por cada dos asistentes, con las siguientes especificaciones mínimas:
 - Sistema operativo linux (Ubuntu preferiblemente).
 - Memoria RAM 512MB.
 - Procesador de 800 MHz.
 - Espacio libre en disco 2 GB.
 - Puerto serie.
- Tarjeta de Desarrollo stamp.

3.2. Requisitos

El curso esta dirigido a profesionales en las áreas de Electrónica Digital y Ciencias de la Computación, o cualquier persona con conocimiento básico de Sistemas Digitales, programación en C y/o C++ y arquitectura de computadores.

3.3. Material didáctico

A los asistentes al curso se les suministrará el siguiente material:

- Plataforma de Desarrollo stamp.
- Copia en CD de todas las presentaciones.
- Software de Desarrollo de libre distribución.
- Código fuente necesario para realizar las prácticas.

Bibliografía

- [1] S. Banguero and M. Erazo. *Plataforma de Desarrollo para Sistemas Embebidos Basada en el GameBoy Advance*. Universidad Industrial de Santander, 2007.
- [2] C. Camargo. ECB_AT91 y ECBOT Plataformas Abiertas para el desarrollo de Sistemas Embebidos. URL: <http://wiki.emqbit.com/free-ecb-at91>.
- [3] C. Camargo. First Colombian Linux SBC runs Debian. URL: <http://www.linuxfordevices.com/c/a/News/First-Colombian-Linux-SBC-runs-Debian/>, 2006.
- [4] C. Camargo. ECBOT: Arquitectura Abierta para Robots Móviles. *VII conferencia Iberoamericana en Sistemas, Cibernética e Informática*, 2008.
- [5] 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, Madrid España*, September 2008.
- [6] C. Camargo. ECBOT: Arquitectura Abierta para Robots Móviles. *IEEE Colombian Workshop on Circuits and Systems*, 2007.
- [7] C. Camargo. PLATAFORMAS ABIERTAS HARDWARE/SOFTWARE PARA APLICACIONES EN ROBOTICA. *V Congreso Internacional de Ingeniería Mecánica y III de Ingeniería Mecatrónica (por publicar)*, 2011.
- [8] C. Camargo. SIE: Plataforma Hardware copyleft para la Enseñanza de Sistemas Digitales. *XVII Workshop de Iberchip, Bogotá Colombia*, February 2011.