

Hardware Libre: la Tarjeta Skypic, una Entrenadora para Microcontroladores PIC

Juan González-Gómez¹ y Andrés Prieto-Moreno Torres²

¹ Escuela Politécnica Superior, Universidad Autónoma de Madrid,
juan.gonzalez@uam.es

² Ifara Tecnologías,
andres@ifara.com

Resumen Las ideas del *software* libre se están extendiendo a otros ámbitos. Uno de ellos es el *hardware*. En este artículo presentamos una tarjeta libre, para desarrollo de proyectos con microcontroladores PIC. Es *hardware* libre, por lo que cualquiera la puede utilizar, estudiar, fabricar, modificar, distribuir y redistribuir las modificaciones. Las aplicaciones principales son la robótica y la docencia, aunque se puede utilizar en cualquier proyecto donde se requiera un microcontrolador. Los diseños de *hardware* libre presentan muchas ventajas para la sociedad, siendo la mayor de ellas el aumento del conocimiento tecnológico: están ahí no sólo para ser usados, sino para que cualquiera pueda comprender su funcionamiento interno.

1. Introducción

El *software* libre ofrece al usuario cuatro libertades: libertad de uso, de estudio y modificación, de distribución, y de redistribución de las mejoras. Existen licencias que las garantizan y que dan una cobertura legal, como por ejemplo la GPL.

Estas ideas se pueden aplicar a otros campos, como el del diseño *hardware*, dando lugar a lo que se conoce como *hardware* libre o *hardware* abierto[1].

No existe una definición universalmente aceptada de *hardware* libre, ni tampoco está disponible una licencia similar a la GPL. El criterio que hemos seguido es el propuesto en[1], que considera que un diseño *hardware* es libre siempre y cuando se garantice que las mismas libertades del *software* libre se aplican a los “ficheros fuente” de los planos *hardware*: esquemático, PCB (*Printed Circuit Board*) y fichero *gerber* para fabricación industrial.

Los microcontroladores PIC, de Microchip, son muy populares y cada vez se utilizan en más desarrollos. Están muy extendidos en la robótica, usándose para gobernar tanto pequeños robots móviles[2,3] como prototipos de robots modulares reconfigurables [4]. Su reducido coste y los diferentes encapsulados en que se distribuyen, los hacen muy atractivos para la construcción de tarjetas entrenadoras, de tipo industrial o prototipos, realizadas por empresas[5], universidades[6,7] o particulares[8,9,10].

La tarjeta Skypic[11] es una entrenadora de propósito general, para trabajar con los microcontroladores PIC de 28 pines. Y es *hardware* libre en el sentido antes indicado: están disponibles todos los “ficheros fuente” de los planos y se conceden permisos para utilizarla, fabricarla, modificarla y distribuirla.

En este artículo se describen las características técnicas de esta tarjeta, sus aplicaciones principales y se discuten las repercusiones más importantes que tienen los diseños libres, tanto para las empresas como para la sociedad.

2. Características técnicas

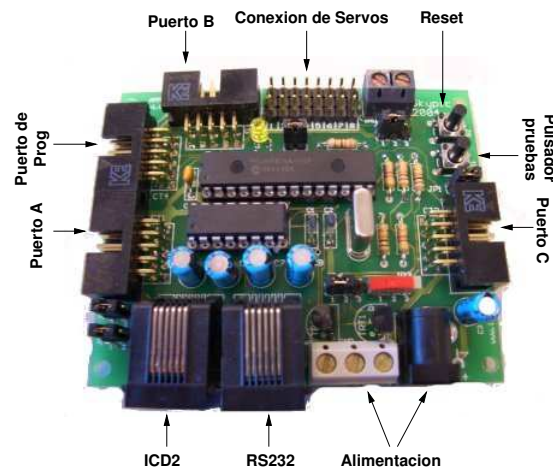


Figura 1. Descripción de los componentes de la Skypic

La tarjeta Skypic permite ejecutar programas en cualquier microcontrolador PIC de 28 pines, aunque se diseñó para el modelo PIC16F876. Contiene sólo la electrónica indispensable para que el microcontrolador pueda funcionar, por lo que tiene unas reducidas dimensiones: 80x65mm. A través de cuatro conectores acodados para cable plano, se tiene acceso a todos los pines del Pic.

Las características principales se muestran en el cuadro 1 y la disposición de los componentes en la figura 1.

Los PIC son microcontroladores de tipo RISC, de 8 bits, que incorporan periféricos muy diversos: temporizadores, unidades de comunicaciones serie síncronas y asíncronas, bus CAN, USB, conversores A/D, comparadores, etc, por lo que se adaptan a una gran variedad de aplicaciones. En el cuadro 2 se han resumido las características del modelo Pic16F876.

La programación del pic se realiza *in circuit*, por lo que no es necesario extraerlo del zócalo. Se puede emplear bien el grabador ICD2 de Microchip, o bien otra tarjeta Skypic con un *software* de grabación (ver apartado 4).

Tarjeta Skypic
Conector para grabar desde el ICD2 de Microchip
Conector telefónico para la conexión RS232 con el PC
Led y pulsador para realizar pruebas
Puerto para programación directa desde la tarjeta CT6811 u otra Skypic
Puertos A,B y C accesibles mediante conector acodado de 10 vías
Conexión directa de 8 servos compatibles Futaba

Cuadro 1. Características de la tarjeta Skypic

PIC16F876
Microprocesador Risc de 8 bits
Frecuencia de Reloj: 4Mhz
Temporizador de 8 bits y uno de 16 bits
Dos unidades de captura, comparación y PWM
Buses síncronos I2C y SSP
Unidad de comunicaciones serie asíncrona
8 canales A/D de 10 bits
Programación “ <i>in circuit</i> ” (ICSP)
Memoria <i>Flash</i> de 8Kb y SRAM de 368 bytes
Memoria <i>eprom</i> de 256 bytes

Cuadro 2. Características del microcontrolador PIC16F876

3. Planos

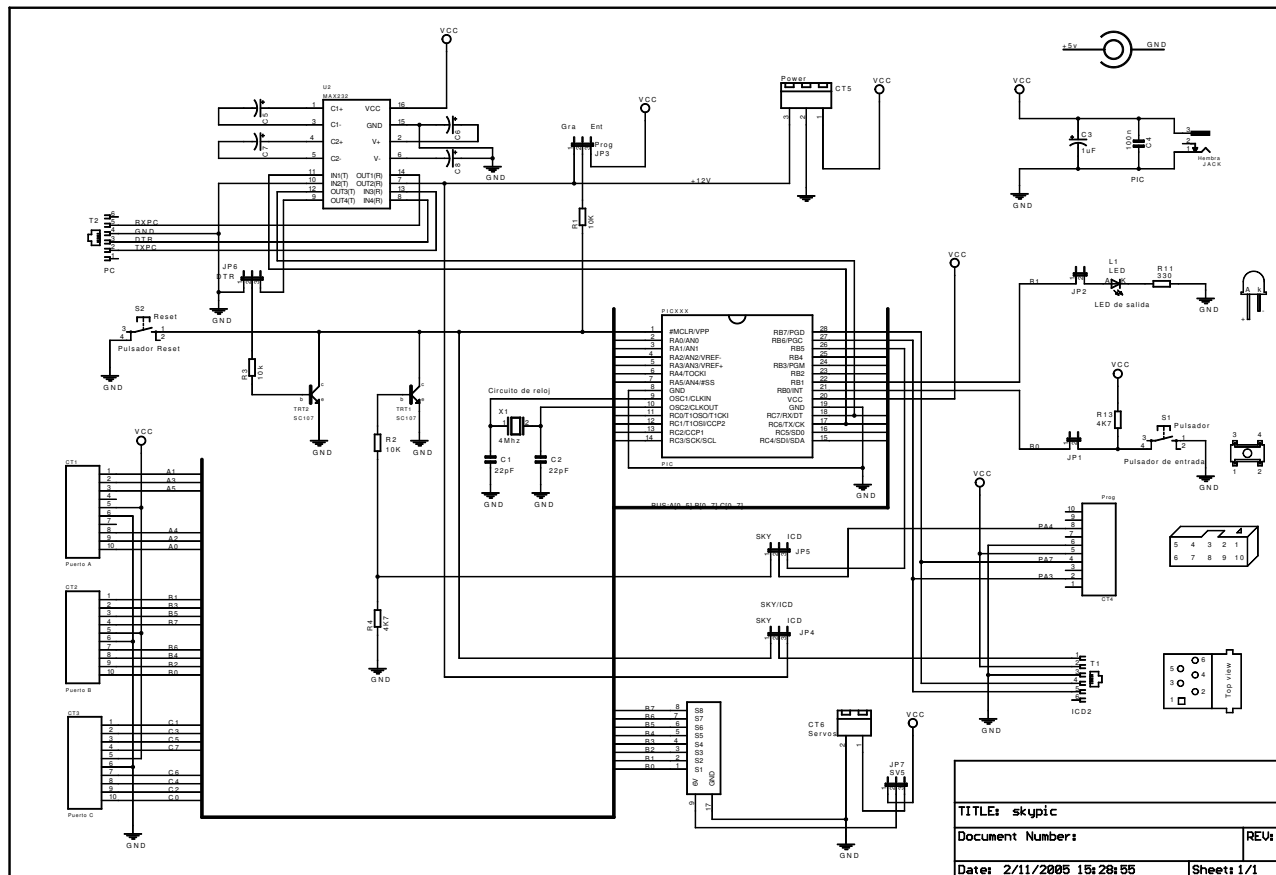
Siguiendo los criterios establecidos en [1], consideraremos que un *hardware* es libre si están disponibles los “ficheros fuentes” de los planos. Nos referimos a los ficheros originales que utiliza el diseñador para la especificación del *hardware* y que cualquier otra persona los pudiese abrir con la misma u otra herramienta de diseño. Una placa de la que sólo se dispongan sus planos en algún formato no editable, como por ejemplo PDF, no se consideraría libre.

Uno de los problemas que aparecen es la falta de un estándar abierto para la especificación de estos planos. Existen herramientas profesionales muy potentes (propietarias) cada una con su propio formato. Puede surgir la duda de si un diseño *hardware* realizado con un programa propietario se puede considerar libre. En nuestro caso, basándonos en criterios prácticos, consideramos que será libre si se ofrecen las cuatro libertades del *software* libre, independientemente de la herramienta empleada (esto se discutió en [1]).

La Skypic se ha diseñado con el programa Eagle[12], que no es libre, pero es multiplataforma (existen versiones para Linux, Max y Windows) y hay disponibles versiones de evaluación, de duración ilimitada que se pueden descargar de la web (de hecho, el eagle está en el repositorio *non-free* de Debian). Todos los planos se pueden descargar de [11].

En la figura 2 se muestra el esquema completo.

Figura 2. Esquema de la tarjeta Skypic



4. Software

Para la programación de aplicaciones en la Skypic se puede utilizar cualquiera de las herramientas de desarrollo para Pícs. Microchip ofrece las aplicaciones “oficiales”, que son propietarias y sólo están disponibles para plataformas Windows.

La comunidad ha creado sus propias herramientas libres, compatibles con las de Microchip. Entre ellas destacan las utilidades de GNU: *gputils*[16], que incluye un ensamblador y un enlazador entre otras. Para la simulación se puede emplear *gpsim*[17] y para la programación en C el compilador *sdcc*[18]. También existe un entorno de programación *PikDev*[19], en el que está todo integrado (editor, ensamblador, enlazador, programador, etc).

La Skypic se ha diseñado para que la grabación *in-circuit* de los programas ejecutables en la memoria *flash* del PIC se pueda realizar de diferentes maneras:

- Entorno Mplab y programador ICD2, de Microchip
- Programa Icprog, para Windows, y cable de conexión paralelo para la Skypic [20].
- Entorno PikDev, para Linux, y cable de conexión paralelo para la Skypic

En [21] planteamos un método de grabación alternativo, a través del puerto serie estándar del PC, que permite utilizar la propia Skypic como si fuese un grabador, similar al ICD2 de Microchip (aunque sin la opción de depuración). Para ello son necesarias dos tarjetas Skypic. Una que funcionará como grabadora, con el servidor *picp*[22] en la *flash*, y la otra es la que se graba. La herramienta *skypic-down*[23], disponible para Linux bajo licencia GPL, permite escribir los programas en la *flash*.

Este sistema de grabación tiene la ventaja de ser totalmente independiente de las características del PC empleado. La temporización se realiza en el *pic* del grabador. El PC sólo tiene que enviar la información a través del puerto serie, a una velocidad estándar de 9600 baudios.

5. Aplicaciones

La Skypic es una tarjeta de propósito general, muy útil en la construcción de prototipos. Una de las aplicaciones para la que fue creada es la construcción de microbots, pequeños robots que la incorporan como “cerebro”, como por ejemplo el robot seguidor de línea Skytritt (figura 3), una variante del robot abierto Tritt[13].

Los servomecanismos del tipo Futaba 3003 o compatibles se conectan directamente, lo que es muy útil para el diseño y prueba de robots articulados. El programa *star-servo8*[14], para Linux, permite controlar hasta 8 servos desde el PC, pudiendo generar secuencias de movimiento para los robots articulados.

Otro ámbito de aplicación es el docente. Se puede utilizar en laboratorios de arquitectura de computadores, sistemas digitales o robótica. Los alumnos no sólo se limitarían a usar la placa (libertad 0), sino que también la pueden



Figura 3. El microbot Skytritt, que utiliza la tarjeta Skypic como “cerebro”

estudiar (libertad 1) o modificar (libertad 2) para adaptarla a sus propios diseños. También se puede tomar de ejemplo para aprender a diseñar placas industriales, obteniendo la información sobre su tecnología de fabricación: anchura de las pistas, diámetro de los pads, planos de masa, serigrafías, etc.

Sin embargo, la mayor utilidad está en el diseño de prototipos. Al ser *hardware* libre, no es necesario “reinventar la rueda”, diseñando el sistema desde cero. Es mejor opción partir de algo que ya funciona y adaptarlo a tus necesidades, ahorrando tiempo y reduciendo costes.

Esto es lo que se ha hecho en el proyecto *Chronojump*[15], en el que se está diseñando un sistema *software* y *hardware* para la medición de los tiempos de vuelo de diferentes saltos realizados por deportistas, para conocer su estado de forma, la eficacia de un entrenamiento, la evolución del deportista, etc. Para la creación del prototipo *hardware* se está empleando una Skypic, que más adelante se modificará para adaptarla a las necesidades concretas del proyecto. Si no fuese *hardware* libre las dos únicas opciones que se tendrían serían: 1) diseñar el *hardware* desde cero, 2) Adaptar el proyecto al *hardware* existente en el mercado.

6. ¿Por qué hardware libre?

Un diseño libre lo puede fabricar, distribuir y vender cualquier empresa o persona. Esto cambia las reglas del juego. Analicemos algunas de las ventajas:

- **Adaptar, no reinventar.** En el mundo *hardware*, siempre se diseña desde cero, reinventándose constantemente la rueda. El conocimiento sólo está disponible en el seno de las empresas fabricantes y sólo está accesible para sus

ingenieros, que además tienen que firmar contratos de confidencialidad. El *hardware* libre permite que cualquiera pueda abordar proyectos complejos, adaptando el *hardware* existente. Hay ahorro de tiempo y dinero.

- **Mayor perdurabilidad en el tiempo.** Los diseños libres son independientes del fabricante. Si la empresa suministradora decide abandonar la fabricación, se puede buscar otra, contratar a alguien para que lo haga o bien hacerlo nosotros mismos. Deja de tener sentido la frase “ya no se fabrica ese *hardware*”.
- **Monopolios no, competencia sí.** El *hardware* libre garantiza la no existencia de monopolios. Cualquier empresa o particular puede fabricarlo (siempre que tenga los conocimientos y la experiencia).
- **Aumento del conocimiento tecnológico.** Se introduce una nueva dimensión en el conocimiento. Los aparatos electrónicos no sólo estarán para ser usados, sino también para ser comprendidos y modificados. El efecto “lo uso pero no lo entiendo” desaparece.

Los beneficios para la sociedad son claros, sin embargo surge la pregunta: “¿Por qué una empresa va a “regalar” su conocimiento “gratis” para que los competidores se aprovechen de él?”.

Los autores no tenemos la respuesta y sólo el tiempo podrá decir si el modelo basado en el conocimiento libre triunfará. El *hardware* libre está en pañales y hoy en día, la respuesta de una empresa a la pregunta anterior sería un rotundo no. La misma respuesta que hubiesen dado las empresas de *software* hace diez años. Actualmente, ya existen empresas que están liberando código. ¿Podemos esperar un comportamiento similar a largo plazo con el *hardware* libre?

Al día de hoy, las ventajas que obtendría una empresa por liberar *hardware* podrían ser: prestigio y reconocimiento de la comunidad, publicidad, aumento de la motivación de los diseñadores, captación de desarrolladores que se involucren en nuevas versiones del *hardware*, aparición de *software* libre alrededor de ese *hardware*, etc.

7. El Futuro del hardware libre

Un campo especialmente activo es el del *hardware* reconfigurable[1], aquel que se puede describir utilizando un lenguaje de descripción *hardware*, como VHDL o Verilog entre otros. Este *hardware* tiene la características de ser similar al *software*. Los diseños quedan perfectamente definidos mediante unos ficheros fuentes que se pueden compilar, simular, sintetizar y descargar en una FPGA. La licencia GPL se puede aplicar.

Existen muchos diseños libres de este tipo y están surgiendo comunidades muy importantes, como OpenCores[24]. También existen proyectos de gran envergadura como es el procesador *Sparc* LEON[25], diseñado por la agencia espacial europea (ESA) y distribuido bajo licencia GPL.

El futuro del *hardware* estático, el discutido en este artículo, es más incierto. Hasta ahora, no existían herramientas profesionales libres comparables a las

propietarias. Los pocos diseños libres que había, estaban diseñados cada uno con una aplicación diferente, lo que dificultaba su compartición. Sin embargo, hace poco han liberado Kicad[26], muy similar al Eagle. Es una herramienta muy prometedora que puede conseguir que su formato se convierta en un estándar para los planos electrónicos.

El impulso del *hardware* libre lo deben dar los miembros de la comunidad: los propios usuarios. No podemos esperar por el momento que las empresas liberen su *hardware*. Sin duda será un proceso lento, pero tenemos como referencia la evolución del *software* libre.

8. Conclusiones y trabajo futuro

En este artículo hemos presentado la tarjeta Skypic, una entrenadora para microcontroladores Pic, que tiene una licencia de *hardware* libre. Los “ficheros fuentes” de los planos están disponibles y se conceden permisos para su uso, estudio, fabricación, modificación y redistribución de las mejoras.

Las aplicaciones principales son la robótica y la docencia. También resulta muy útil en la realización de proyectos que requieran el uso de un microprocesador. En vez de diseñar el sistema desde cero, se puede usar la Skypic y adaptarla a las necesidades concretas. Un ejemplo es el proyecto Chronojump.

El *hardware* libre ofrece una serie de ventajas a la sociedad, siendo quizás la más importante el aumento del “patrimonio tecnológico”, en el sentido de aportar no sólo un nuevo instrumento, sino el conocimiento de cómo está realizado.

Actualmente, los diseños libres (de tipo estático) son escasos y no existe todavía la suficiente masa crítica. No obstante, la idea tiene el potencial suficiente como para que se vaya extendiendo.

Una sociedad tecnológica como la nuestra, no debe caracterizarse sólo por la existencia de gran cantidad de “aparatos” o “cajas negras” que todo el mundo usa pero que pocos conocen sus detalles internos. El *hardware* libre es otra pieza más hacia la sociedad del conocimiento libre que muchos deseamos.

Como trabajo futuro evaluaremos la herramienta libre de diseño electrónico Kicad y la utilizaremos para rehacer los planos de la Skypic, consiguiendo así una placa que además de ser libre, esté diseñada íntegramente con *software* libre.

Agradecimientos

Queremos agradecer a la empresa Ifara Tecnologías la financiación de la primera tirada de PCBs de la Skypic. Valoramos muy positivamente el que haya empresas que estén dispuestas a dar una oportunidad a las nuevas ideas.

Referencias

1. I. González, J. González, F. Gómez-Arribas (2003), “Hardware libre: clasificación y desarrollo de hardware reconfigurable en entornos GNU/Linux”. VI Congreso de Hispalinux. [En línea]

- <http://www.learobotics.com/personal/juan/publicaciones/art4/index.html>.
(consulta 10-Feb-2005).
2. Organización Hispabot. "KIT de microrrobot velocista para talleres de introducción a los microrrobots". Seminario Hispabot 2003, Alcalá de Henares, Madrid, Mayo 2003.
 3. D. Amor, J.M. Castro, J. Donate, A. Gutierrez, I. Navarro. "PI y TC-EPI: Una nueva iniciativa de hardware abierto orientado a la formación de la robótica móvil". Seminario Hispabot 2003, Alcalá de Henares, Madrid, Mayo 2003.
 4. Módulos G1v4 del robot Polybot, en el PARC (2002). [En línea]
<http://www2.parc.com/spl/projects/modrobots/chain/polybot/g1v4.html> (consulta 10-Feb-2005).
 5. Tarjeta Micropic Trainer, desarrollada por la empresa Microsystems Engineering. [En línea]
<http://www.pacalaconcurso.com/modules.php?name=News&file=article&sid=6>
(consulta 18-Feb-2005).
 6. I. Bravo, O. Durán, M.A. García, J.L. Lázaro, E. Martín, M. Marrón, "Programación en placa de dispositivos PIC". Seminario Hispabot 2003, Alcalá de Henares, Madrid, Mayo 2003.
 7. Tarjeta entrenadoras PICMIN y PICUPSAM, prototipos empleados para la docencia de la asignatura "Laboratorio de Arquitectura de computadores" en la UPSAM. [En línea]
<http://www.learobotics.com/personal/andres/proyectos/picmin/pic1.html>.
(consulta 18-Feb-2005).
 8. Placas Fast-pic y Power-pic, diseñadas por Víctor Apéstigue Palacio. [En línea]
<http://www.terra.es/personal5/cuchomen1/Micros/micros.htm> (consulta: 18-Feb-2005).
 9. Alejandro Alonso Puig, "Diseño de un servomotor controlado por bus I2C mediante microcontrolador PIC de gama media". [En línea]
<http://www.mundobot.com/tecnic/Svd01/Svd01.htm> (consulta: 18-Feb-2005).
 10. Tarjetas X-PIC y X-BOT. Alberto Calvo torrijos y Daniel Álvarez Sánchez. [En línea]
<http://www.x-pic-system.tk/> (consulta 18-Feb-2005).
 11. Tarjeta entrenadora Skypic. [En línea]
<http://www.learobotics.com/proyectos/skypic/skypic.html>. (consulta: 10-Feb-2005).
 12. Programa de diseño electrónico Eagle, de la empresa CadSoft. [En línea]
<http://www.cadsoft.de/> (consulta: 18-Feb-2005).
 13. Microbot Tritt. [En línea] <http://www.learobotics.com/proyectos/tritt/tritt.html>
(consulta: 10-Feb-2005).
 14. Programa de manejo de servos desde el PC: star-servos8. [En línea]
<http://www.learobotics.com/proyectos/stargate/clientes/star-servos8/star-servos8.html> (consulta: 17/feb/2005).
 15. Proyecto Chronojump. Albergado en Software-libre.org. [En línea]
<http://chronojump.software-libre.org/> (consulta: 18-Feb-2005).
 16. GPUTILS. Utilidades GNU para los PIC. [En línea] <http://gputils.sourceforge.net/>
(consulta: 18-Feb-2005).
 17. GPSIM. Simulador GNU para los PIC. [En línea]
<http://www.dattalo.com/gnupic/gpsim.html> (consulta: 18-Feb-2005).
 18. Programa SDCC (Small Device C Compiler). Compilador de C para diversor microcontroladores de 8 bits, entre ellos los PIC. [En línea] <http://sdcc.sourceforge.net/>
(consulta: 18-Feb-2005).

19. Entorno de desarrollo PiKdev. [En línea] <http://pikdev.free.fr/> (consulta: 18-Feb-2005).
20. Ricardo Gómez González. “Manual para grabar la SKY-PIC desde el puerto paralelo con el ICPROG”. [En línea]. http://www.iearobotics.com/proyectos/skypic/docs/conf_icprog.html (consulta 17-Feb-2005).
21. Juan González, Andrés Prieto-Moreno, ”Herramientas hardware y software para el desarrollo de aplicaciones con Microcontroladores PIC bajo plataformas GNU/Linux”, III Jornadas de Software Libre, Universidad Pontificia de Salamanca en Madrid. Mayo 2004. [En línea] <http://www.iearobotics.com/personal/juan/publicaciones/art5/index.html> (consulta: 18-Feb-2005)
22. Servidor de grabación para microcontroladores pic. [En línea] <http://www.iearobotics.com/proyectos/stargate/servidores/sg-picp/sg-picp.html> (consulta: 17-Feb-2005).
23. Skypic-down. Cliente de grabación de microcontroladores pics, para Linux [En línea]. <http://www.iearobotics.com/personal/juan/proyectos/skypic-down/skypic.html> (consulta: 17-Feb-2005).
24. OpenCores. Cores libres. [En línea] <http://www.opencores.org/> (consulta: 18-Feb-2005).
25. Procesador Sparc LEON2. [En línea] <http://www.gaisler.com/products/leon2/leon.html> (consulta: 18-Feb-2005).
26. Programa profesional y libre de diseño electrónico: KICAD. [En línea] http://www.lis.inpg.fr/realise_au_lis/kicad/ (consulta: 18-Feb-2005).