

# Tarjeta entrenadora para FPGA, basada en hardware abierto

Juan González, Pablo Haya, Sergio López-Buedo, Eduardo Boemo  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid

Laboratorio B-209, Tlf: 913482268  
juan@iearobotics.com, Pablo.Haya@ii.uam.es,  
Sergio.López-Buedo@ii.uam.es, Eduardo.Boemo@ii.uam.es

## Resumen

En este artículo se presenta la placa JPS, una tarjeta entrenadora para FPGAs de Xilinx, diseñada con una licencia abierta que permite a cualquiera disponer de los esquemas y planos relativos al PCB, modificarlos y/o distribuirlos, así como fabricar la placa. La tarjeta permite realizar diseños autónomos sobre algunos modelos de FPGAs Spartan y XC4000. Éstos se pueden cargar desde una EEPROM serie incluida en la placa, o trabajar en modo entrenador, cargando los diseños desde el PC.

## 1. Introducción

La existencia de lenguajes de descripción *hardware* (HDL) tales como VHDL, Verilog, Handel-C, etc hace que los diseños sean ficheros de texto, que contienen el “código fuente” del circuito. Así, el proceso de diseño de *hardware* cada vez se parece más al de *software*. Esto ha conducido a que el *hardware* se convierta en algo que se puede compartir, al estilo de lo que sucede con licencias abiertas tales como GPL[1]. Por ejemplo, han aparecido iniciativas como la de Open Cores[2], donde se publican diseños complejos bajo una licencia abierta que permite ser compartidos por el resto de las llamadas “comunidades de desarrollo”.

Dentro del contexto del *hardware* libre, parece necesario contar con una placa sobre la que probar los circuitos, y que ésta a su vez sea libre, pudiéndose compartir los esquemas, PCBs y ficheros de fabricación de manera que cualquiera la pueda fabricar, usar, modificar, y redistribuir las modificaciones.

Para aplicaciones de robótica, se utilizan placas microcontroladores, de 8 bits, tales como la GP-BOT[6, 5] basada en el microcontrolador 68hc08 o la CT6811[3], que usa el 68hc11. La tarjeta JPS también está orientada a la robótica y ha sido diseñada para que se puede utilizar junto las tarjetas mencionadas arriba, complementándolas o en su caso sustituyéndolas.

## 2. Introducción a las FPGAs

A la hora de implementar un sistema electrónico digital, el diseñador dispone de un conjunto amplio de tecnologías. Una de las más populares actualmente son los dispositivos de lógica programable (PALs, PLDs, FPGAs,...). Los dispositivos de lógica programable más versátiles son las FPGA[10, 11] (*Field Programmable Gate Array*). Internamente, una FPGA esta compuesta por un conjunto de bloques iguales (CLBs) dispuestos de forma regular. Cada bloque contiene pequeñas memorias RAM y *flips-flops* que se pueden configurar para realizar todo tipo de circuitos combinatoriales y secuenciales de pequeña escala. Los bloques se pueden interconectar entre sí mediante conexiones también configurables. La configuración de la FPGA se realiza mediante una comunicación serie denominada *bitstream*, que puede estar almacenado en una memoria externa (PROM, EEPROM, RAM...) o provenir de otro sistema (PC, microcontrolador, otra FPGA...).

La tecnología FPGA permite realizar diseños a medida, de bajo coste de desarrollo, incluso para la producción de pocas unidades. Estas características la hacen muy interesante para realizar prototipado rápido. Especialmente tiene un gran interés dentro del campo docente. En el caso de la tarjeta JPS, el estudiante puede definir su propio microcontrolador (arquitectura y juego de instrucciones) específicamente orientado a aplicaciones de Robótica. En unas pocas semanas puede tener su prototipo funcionando e interactuando con los motores, sensores, etc. Adicionalmente, puede utilizar la JPS para construir periféricos complejos como medidores de distancia por ultrasonido, conversores serie paralelo, unidades de PWM, bloques de transmisión de datos, etc. Sobre FPGAs, la mayoría de estas aplicaciones funcionan cerca un orden de magnitud más rápido que en un microcontrolador.

### 3. Características de la tarjeta

La tarjeta JPS-XP84[4] (ver foto en la figura 1) es una tarjeta entrenadora para FPGAs, de propósito general, aunque está pensada para ser utilizada fundamentalmente en robótica. Sus dimensiones son reducidas (99x73mm) y dispone de conectores de expansión de 10 vías, compatibles con cables planos.



Figura 1: La tarjeta JPS, situada al lado de un disquete de 3 y 1/2”

Las FPGAs de la empresa Xilinx[7] que soporta la JPS se muestran en la tabla 1. Todas ellas tienen el encapsulado PLCC84, elegido por ser el de menor coste. Adicionalmente, es uno de los pocos encapsulados actuales que pueden ser soldados sin herramientas especiales para SMD.

Device	Logic cells	Total CLBs	Total FF
XC4003E	238	100	360
XC4005E	466	196	616
XC4010E	950	400	1120
XCS05	238	100	360
XCS10	466	196	616

Cuadro 1: FPGAs soportadas por la tarjeta JPS

Las características de la tarjeta son:

- 6 puertos de expansión (48 *pin*s), que dan acceso a las patas genéricas de entrada/salida de la FPGA.
- Zócalo para la inserción de un oscilador que genere la señal de reloj para los diseños síncronos.
- *LED* y pulsador de pruebas, conectados a unos terminales fijos de la FPGA, para hacer pruebas de funcionamiento, sin tener que conectar ningún circuito externo.
- *LED* de programación, conectado al terminal *done*, que indica si la FPGA está correctamente programada.

- Tres *switch*s de configuración genéricos, conectados a tres terminales fijos, para ser utilizados como entradas de configuración en los diseños realizados.
- Configuración del modo de carga de la FPGA: maestro o esclavo. El modo maestro permite cargar la FPGA desde la memoria EEPROM integrada en la propia tarjeta. El modo esclavo se utiliza para cargarla desde un sistema externo (PC o un microcontrolador)
- Programación *in circuit* de la EEPROM.
- Pulsador para hacer *reset* de la FPGA.
- Conector para usar el cable de descarga (*download*) estándar de Xilinx

### 4. Modos de funcionamiento

La tarjeta puede trabajar en dos modos de funcionamiento: modo entrenador y modo autónomo. En el modo entrenador, la tarjeta se conecta a un PC desde el que se reconfigura la FPGA, usando el cable de descarga de Xilinx o bien uno construido por el propio desarrollador. Una vez que la simulación del diseño cumple con las especificaciones, y el circuito se ha sintetizado correctamente, se conecta la placa y se descarga el *bitstream*, permitiendo comprobar físicamente su funcionamiento (realización de mediciones, conexión de otros dispositivos, etc).

Para su utilización en robótica, es muy útil el modo autónomo, en el que el diseño se graba en una memoria EEPROM serie y la FPGA se configura automáticamente cada vez que se conecta la tarjeta o se pulsa el botón de *reset*. Usando una FPGA del tipo XC10, el tiempo de reconfiguración es de 200ms.

Otra opción es configurar la tarjeta en modo entrenador (FPGA en modo esclavo) pero realizando la carga desde un sistema microcontrolador, en vez de desde la EEPROM serie. Esto abre la posibilidad de trabajar con sistemas reconfigurables o adaptables, que descarguen un determinado circuito según las necesidades concretas.

### 5. Diagrama de bloques

El diagrama de bloques se muestra en la figura 2. El bloque principal lo constituye la FPGA, a la que se añade una circuitería adicional, dividida en los siguientes bloques:

- **Circuito de reloj**, para la realización de diseños síncronos. La frecuencia del oscilador empleado depende de la aplicación de usuario.
- **Circuito de programación interno**, constituido por la memoria EEPROM serie y un multiplexor para que

los pines de la EEPROM sean accesibles bien desde la FPGA, para su carga, o bien desde los pines del puerto de control para su programación in circuit.

- **Circuito de programación externa**, que permite descargar *bitstreams* desde el PC o desde un sistema externo.
- **Circuito de configuración**: jumper y switch para la configuración de los diferentes modos de trabajo. Mediante el jumper se pueden configurar el modo de trabajo: entrenador o autónomo. Mediante un conmutador se selecciona si la memoria EEPROM se conecta a la FPGA o al puerto de control para programarla desde un sistema externo, sin tener que sacarla del zócalo.
- **Circuito de pruebas**, constituido por un led y un pulsador conectados a los pines P68 y P69 de la FPGA, que permiten probar el correcto funcionamiento de la placa, configurando la FPGA con un diseño de pruebas que los use, como por ejemplo un puerta inversora entre ellos.
- **Puertos de expansión**. La placa incorpora 6 puertos de expansión, con 8 bits de datos, configurables para entrada o salida, y dos pines para la alimentación, de forma que los circuitos externos conectados se puedan alimentar directamente a través de los cables de bus. Están diseñados para ser compatibles con los conectores de la tarjeta GP-BOT[5] y CT6811[3].

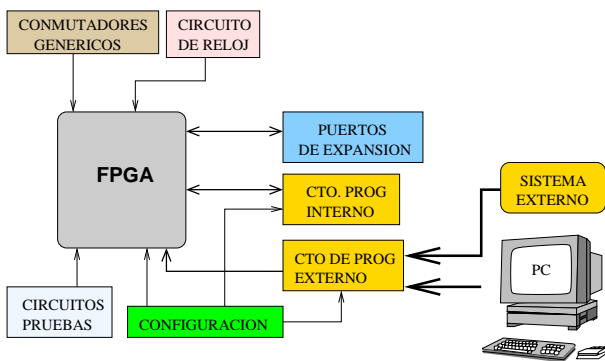


Figura 2: Diagrama de bloques de la tarjeta JPS

Todos los detalles se pueden ver en el esquema de la figura 3

## 6. Campos de aplicación

La tarjeta JPS se puede emplear para múltiples aplicaciones. Sin embargo, resulta especialmente útil en los siguientes campos:

- **Docencia**, en el campo de la electrónica y los lenguajes de descripción hardware (HDL). Los alumnos diseñan, simulan, sintetizan y finalmente descargan el diseño en la tarjeta entrenadora, pudiendo comprobar físicamente el correcto funcionamiento, tomando medidas y conectando circuitos externos. Por ser una placa libre y estar disponibles todos los esquemas y planos de fabricación, los alumnos también pueden ampliar su funcionalidad, diseñándose placas a su medida (realización de prácticas especiales, proyectos fin de carrera, trabajos de doctorado, etc). De la misma manera los profesores la pueden adaptar a las necesidades de una asignatura concreta.

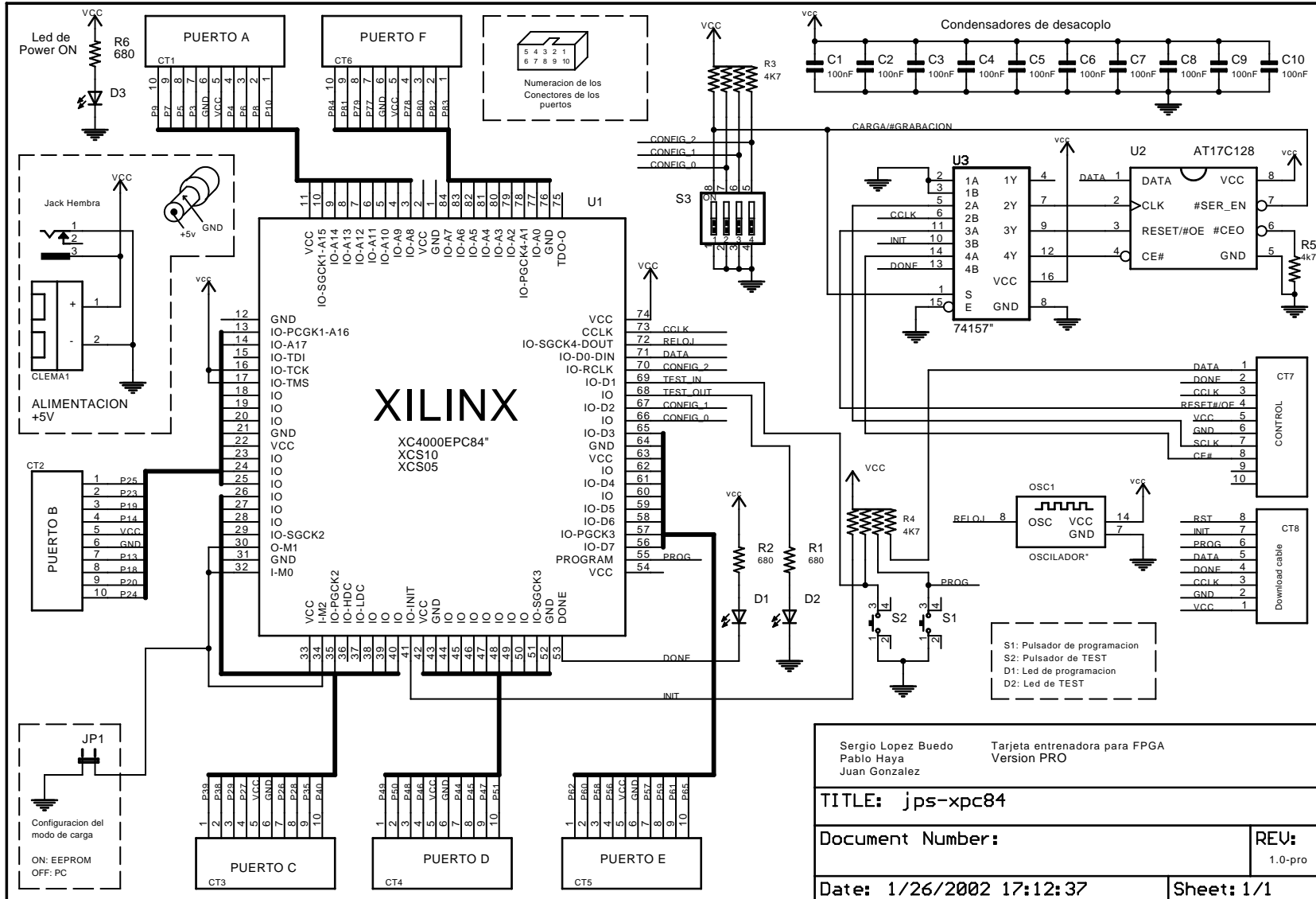
En el laboratorio de Estructura y Diseño de Circuitos Digitales[8] de la Universidad Autónoma de Madrid se están empleando 12 placas JPS para que los alumnos puedan sintetizar sus diseños en ellas y no se queden sólo en la fase de la simulación.

- **Conexión con microcontroladores**. Puesto que puede funcionar en modo autónomo, resulta muy útil para el desarrollo de periféricos para microcontroladores: controladores de sensores, coprocesadores para hacer ciertas operaciones más rápidamente, etc. Además es posible realizar la carga del bitstream desde el propio sistema microcontrolador, posibilitando el desarrollo de sistemas de hardware reconfigurables.
- **Robótica**. Posibilidad de diseñar CPUs específicas para aplicaciones de robótica así como controladores para periféricos: unidades de PWM para mover servos, temporizadores, controladores para sensores de ultrasonidos, de distancia, etc.

## 7. Hardware abierto

El software libre o de fuente abierta (Free Software, Open Source) se caracteriza porque permite a cualquiera copiar, modificar y distribuir las aplicaciones, siempre acompañadas de todos los ficheros fuentes (por ejemplo la licencia GPL[1]).

El hardware libre o abierto toma las mismas ideas del software, pero aplicadas a este campo. El que la placa JPS sea hardware abierto significa que se distribuye junto con todos sus esquemas y ficheros necesarios para su fabricación y se otorga permiso explícito para que cualquiera copie los esquemas, los modifique o fabrique la placa, manteniendo siempre esta libertad en cualquier diseño derivado.



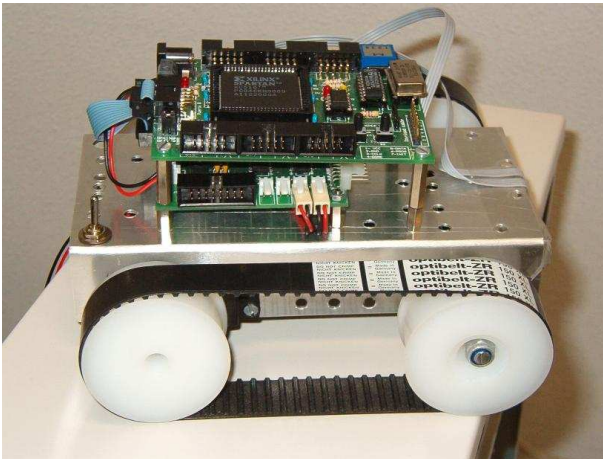


Figura 4: Robot de docencia, que lleva una CPU diseñada en VHDL y sintetizada en la placa JPS

## 8. Un ejemplo de uso: robot de docencia

En el laboratorio de Arquitectura e Ingeniería de Computadores los alumnos trabajan en el estudio y diseño de una CPU de docencia, de tipo RISC, de 16 bits, con una arquitectura Harvard y un total de 8 instrucciones. Esta CPU ha sido implementada en VHDL y sintetizada en una FPGA de tipo Spartan I (XCS10).

Una de las aplicaciones realizadas ha sido el robot de docencia, mostrado en la figura 4, programado para seguir una línea negra sobre un fondo blanco. Los resultados de sintetizar la CPU junto con la memoria que contiene el programa se muestran en la tabla 2.

Número de CLBs	147 (75 %)
Máxima frecuencia	12MHZ

Cuadro 2: Resultados de la síntesis de la CPU del robot de docencia

El programa del robot sigue líneas ocupa en total 56 bytes y la FPGA se encuentra al 75 % de ocupación. Otros ejemplos de aplicaciones se muestra en la tabla 3.

## 9. Ventajas e inconvenientes

Las ventajas de la tarjeta JPS frente a los tradicionales sistemas basados en microcontroladores son las que se derivan de la utilización de las FPGAs:

- **Hardware a la medida.** El diseñador no tiene que buscar los productos del mercado que mejor se adapten a sus diseños, sino que se los diseña a la medida de

Diseño	CLBs ocupados
<b>Algoritmo de criptografía RSA</b>	
Clave de 8 bits	96 (49 %)
Clave de 16 bits	164 (84 %)
<b>Algoritmo de compresión LZ</b>	122 (82 %)
<b>Reloj digital</b>	113 (58 %)

Cuadro 3: Otros diseños sintetizados

sus necesidades, o reutiliza o modifica los diseños ya existentes. En robótica, es muy común emplear un microcontrolador u otro en función de los periféricos que traiga integrados, y es muy común que sólo se utilicen unos pocos de ellos. En un sistema que incorporen FPGAs, es el diseñador el que implemente sólo los controladores necesarios. Se abren nuevas posibilidades de diseño, como por ejemplo el crear una CPU específica para una aplicación determinada junto con sus propios controladores de periféricos.

- **Acortamiento del ciclo de diseño.** El modelo de diseño hardware basado en HDL contiene muchas de las ventajas del diseño software. El circuito es ahora un fichero de texto, que se puede editar, simular, modificar y finalmente sintetizar. Se pueden crear repositorios hardware, con colecciones de diseños ya probados: controladores de VGA, UARTs, temporizadores, CPUs, etc. El diseñador puede crear prototipos muy rápidamente, probarlos, medirlos y modificarlos.
- **Flexibilidad.** Con el mismo hardware físico, conseguimos tener hardware con comportamientos diferentes. En la misma placa JPS, ahora podemos tener sintetizada una CPU, y más adelante podemos probar una UART.
- **Posibilidad de pasar algoritmos al hardware:** En los diseños mixtos microcontrolador/FPGA se puede optar por implementar una solución hardware frente a uno puramente software, consiguiéndose una mayor velocidad. Por ejemplo la implementación de algoritmos de cifrado. En un microcontrolador de 8 bits esto sería un proceso lento, pero se convierte en viable si se añade un hardware que haga el cifrado.
- **Diseños hardware libres.** Posibilidad de realizar diseños hardware libres que se compartan dentro de la comunidad hardware y que cualquier diseñador pueda utilizarlos, modificarlos y distribuir las modificaciones. Esto es especialmente útil en el campo de la docencia y la investigación. Esto se potencia si el hardware físico en el que se prueban los diseños es también libre, como es el caso de la placa JPS. Desaparece la dependencia con el fabricante de la placa y cada Universidad o diseñador puede fabricarse las placas que considere necesarias.

Este tipo de arquitecturas presenta también una serie de inconvenientes:

- **Mayor precio.** Una FPGA en comparación con un microcontrolador es mucho más cara. En concreto, la FPGA utilizada en el robot de docencia cuesta 24 euros frente a los 5 euros (precios en el 2003) que costaría típicamente un microcontrolador de 8 bits en el que se puede implementar fácilmente el mismo algoritmo de seguir la línea.
- **Entornos de desarrollo propietarios, con licencias altas.** Para realizar la síntesis del hardware hay que utilizar las herramientas del fabricante de las FPGA, que son caras y al día de hoy no existen alternativas libres.

- [7] Empresa Xilinx, fabricante de FPGAs. <http://www.xilinx.com>
- [8] Estructura y Diseño de Circuitos Digitales. [www.ii.uam.es/~ivan/temario.htm](http://www.ii.uam.es/~ivan/temario.htm)
- [9] Sven E. Wahlstrom. Programmable Logic Arrays – Cheaper by the Millions Electronics, 40(25), pp. 90-95, December 11, 1967.
- [10] Oldfield J. and Dorf R. , "Field-Programmable Gate Arrays. Reconfigurable Logic for Rapid Prototyping and Implementation of Digital Systems", John Wiley & Son. 1995.
- [11] Trimberger S., "Field-Programmable Gate Arrays Technology", ", Boston: Kluwer Academic Publishers, 1995.

## 10. Conclusiones

Se ha diseñado una placa pequeña, sencilla y libre, para que los estudiantes puedan construir prototipos utilizando FPGA's. En la Universidad Autónoma de Madrid se está empleando en el laboratorio de Estructura y Diseño de Circuitos Digitales y en el de Arquitectura e Ingeniería de Computadores. Con esta placa se ha construido un prototipo de un robot seguidor de línea (robot de docencia) que en vez de un microcontrolador clásico utiliza una CPU diseñada en VHDL y sintetizada en una FPGA.

## Referencias

- [1] Licencia pública GNU. Traducción al castellano. <http://gugs.sindominio.net/gnu-gpl/gples.html>
- [2] OpenCores. Repositorio de diseños hardware, descritos en lenguajes HDL, para ser sintetizados en FPGAs. <http://www.opencores.org>.
- [3] Tarjeta CT6811, basada en el microcontrolador 68hc11. <http://www.iearobotics.com/proyectos/ct6811/ct6811.html>
- [4] Tarjeta JPS. <http://www.iearobotics.com/personal/juan/doctorado/jps-xpc84/jps-xpc84.html>
- [5] Tarjeta GP-BOT, basada en el microcontrolador 68hc908GP32. [http://www.ii.uam.es/~gdrivera/robotica/gp\\_bot/gp\\_bot.htm](http://www.ii.uam.es/~gdrivera/robotica/gp_bot/gp_bot.htm)
- [6] G. González de Rivera, S. López-Buedo, I. González, C. Venegas, J. Garrido y E. Boemo, "GP\_BOT: Plataforma Hardware para la enseñanza de Robótica en Ingeniería Informática". TAAE'02. Pp 67-70. Universidad de las Palmas de Gran Canaria, España. Febrero de 2002.