

Herramientas hardware y software para el desarrollo de aplicaciones con Microcontroladores PIC bajo plataformas GNU/Linux

Juan González Gómez¹,
Andrés Prieto-Moreno Torres²

29-Enero-2004

¹Escuela Politécnica Superior, Universidad Autónoma de Madrid

²Ifara Tecnologías

Resumen

Los microcontroladores *PIC* están muy extendidos. Microchip, la empresa que los fabrica y distribuye, sólo ofrece herramientas de desarrollos para plataformas *Windows*. En este artículo describimos algunas de las herramientas libres que nos permiten trabajar con ellos desde *GNU/Linux* y presentamos la arquitectura del grabador que hemos desarrollado, así como el *software* creado para realizar la descarga. Toda la información está publicada en la web. Se pueden encontrar todos los detalles en los enlaces que se indican.

1. Introducción

Los **microcontroladores PIC**, de Microchip[1], se están popularizando y cada vez se utilizan en más desarrollos. Están muy extendidos en la **robótica**, usándose para gobernar tanto pequeños robots móviles[2, 3, 4] como prototipos de robots modulares reconfigurables[5]. Su reducido coste¹ 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[6], universidades[7, 8] o particulares[9, 10].

También se están empleando en **asignaturas teóricas** y **laboratorios** de diferentes universidades Españolas, como "Sistemas informáticos II" en la Universidad de Huelva[11], "Componentes Electrónicos" (Universidad de Oviedo)[12], "Informática Industrial" (Universidad de Sevilla)[13], "Diseño de Sistemas Basados en Microprocesador" (Universidad Palmas Gran Canarias)[14], "Laboratorio Sistemas Digitales" (Escuela Superior de Ingenieros de Bilbao)[15], "Diseño automático de sistemas digitales" (ETS Ingeniería informática. Universidad de Granada)[15] o "Laboratorio de Arquitectura de computadores" (Universidad Pontificia de Salamanca en Madrid)[17] entre muchas otras.

Existen incluso **implementaciones en VHDL**[18] y **Verilog**[19], lo que permite utilizar *cores* de **PICS en FPGAs**.

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.

Microchip ofrece gratis el programa **MPLAB**, un entorno que permite ensamblar y simular el *software* para el *PIC*, así como grabarlo en la *flash* (si se dispone de un grabador). Este programa **sólo está disponible para Windows**.

¹El 16F876A, de gama media, cuesta 14,5 euros, para 1 unidad. Precio tomado de Amidata (RS)

En este artículo analizamos las herramientas disponibles para poder trabajar con estos microcontroladores desde plataformas *GNU/Linux* y describimos el *hardware* y *software* desarrollado que nos ha permitido cerrar el ciclo completo de diseño (ensamblar, simular y grabar). Nos hemos centrado en la familia de microcontroladores **PIC16F8X** y **PIC16F87X**.

2. ¿Por qué plataforma Linux?

Al desarrollar sistemas basados en microcontroladores, lo importante es el sistema diseñado y no tanto las herramientas o plataformas empleadas para su diseño. El diseñador selecciona un microcontrolador por sus características, precio, etc, pero no tiene sentido que el fabricante nos oblique a usar una determinada plataforma. Para programar los *PICs* se utiliza el programa **MPLAB**, que se ejecuta sobre *Windows*, por lo que habrá que adquirir, además, una licencia de este sistema operativo. O dicho de otra manera, “*o eres usuario de Windows o no podrás utilizar los PICs*”. Y es precisamente esto lo que ha provocado que los autores de este artículo nos centrásemos en analizar las herramientas disponibles para Linux, de manera que el diseñador elija lo que más le convenga y no venga impuesto por el fabricante, cuya decisión de utilizar una plataforma u otra puede deberse a acuerdos comerciales y quizás no técnicos.

Las ventajas que se obtienen al utilizar *software* libre para trabajar con los *PICs*, en entornos docentes, son:

- **Multiplataforma.** El diseñador puede elegir entre una gran variedad de plataformas. Por ejemplo, la distribución **Debian** de Linux está disponible para micros *Alpha*, *Arm*, *HP PA-RISC*, *Intel x86*, *Intel IA-64*, *Motorola 680x0*, *MIPS*, *MIPS (DEC)*, *PowerPC*, *IBM S/390* y *SPARC*, en total **11 arquitecturas diferentes**.
- **Independencia del fabricante.** No estamos atados a los caprichos del fabricante. Si nos conviene utilizaremos su *software*, pero tenemos la opción de usar otras plataforma que no estén soportadas por él. Hoy puedo estar utilizando una máquina *Windows*, pero quizás mañana use una estación de trabajo. En cualquier caso, puedo seguir programando los *PICs*. Nosotros decidimos, no el fabricante.
- **Creación de un LIVE CDs.** Podemos incluir las herramientas libres de desarrollo para el *PIC* en una distribución *LIVE*², que puede servir como demostración, para impartir talleres y seminarios o para que los alumnos dispongan del mismo entorno de trabajo que en laboratorio y puedan hacer las prácticas sin tener que instalar nada en sus ordenadores.
- **Prácticas avanzadas.** Posibilidad de proponer prácticas para mejorar el propio entorno de trabajo o adaptarlo a las necesidades concretas de una asignatura. Tanto profesores como alumnos lo podrían realizar.
- **Libertad de elección.** La mayor ventaja, y un poco como resumen de todas, es la de que el diseñador tenga la **libertad para poder elegir la plataforma que más le guste**. Si utiliza *Windows*, entonces podrá usar el **MPLAB**. Pero si usa otro sistema operativo, también tendrá opción a utilizar estos microcontroladores.

3. Herramientas en GNU/Linux

3.1. Introducción

Para poder trabajar con los *PICs*, y en general con cualquier otro microcontrolador, se necesitan las siguientes herramientas: un **ensamblador**, con el que generar el fichero con el código ejecutable, un **simulador**, para poder ejecutar los programas paso a paso y detectar errores, y por último un **software de descarga**, que permita grabar el fichero ejecutable en la memoria no volátil del microcontrolador. Se precisa para ello de un elemento *hardware*: el **grabador**.

²Las distribuciones *LIVE* o *LIVE CDs*, son CDs autoarrancables, que contienen un sistema Linux totalmente configurado y que se ejecuta en la memoria RAM del PC, por lo que no necesitan instalación ni tampoco acceden al disco duro.

3.2. Ensamblador y simulador

La herramienta **gputils**[20] incluye el ensamblador *gpasm* y el enlazador *gplink*, además de otras utilidades. Es totalmente compatible con el ensamblador **mpasm** de Microchip, lo que permite pasar los ficheros fuente de un sistema a otro sin tener que modificarlos. Soporta las familias *PIC12xxx*, *PIC16xxx*, *PIC17xxx* y *PIC18xxx*. La versión que se ha evaluado es la 0.11.8.

Para la simulación se usa el programa **gpsim**[21], que permite la ejecución paso a paso, establecimiento de *breakpoints*, visualización de los contenidos de registros, la *RAM*, etc. Dispone de un interfaz gráfico, para *GTK* 1.2, aunque también se puede utilizar en modo comando. La última versión evaluada es la 0.20.14.

Ambas herramientas tienen licencia GPL y están disponibles en la distribución **Debian**, lo que hace muy fácil su instalación y su inclusión en un *Live CD*. Se puede encontrar un breve tutorial sobre ambas en [22].

3.3. Software de descarga: skypic-down

Para realizar la grabación de los PIC se ha desarrollado la aplicación **skypic-down**[23], en línea de comandos, la cual toma un fichero en formato *.hex* y lo graba en la *flash* de los *PICs* de las familias 16F87X ó 16F8X. El programa está escrito en C y tiene licencia GPL.

3.4. Otras herramientas

Existen otras herramientas que no se han evaluado, como son el **SDCC**[24], un compilador de C para microcontroladores de 8 bits, que empieza a soportar los *PIC* (version alfa), o el **Picdev**[25], un entorno *IDE* completo para desarrollar aplicaciones para el *PIC*. Incluye software para la grabación. Estas herramientas no están todavía disponibles en Debian.

4. Grabación

4.1. Introducción

Cuando se trabaja con microcontroladores, **lo más importante es disponer de las herramientas para poder grabar en ellos el software desarrollado**. Si el fabricante no proporciona estas herramientas (tanto un grabador como el software necesario) para la plataforma empleada por el diseñador, de nada sirve disponer de ensambladores o simuladores. Al final no podremos probar nuestra aplicación.

Es por ello de vital importancia **disponer de un grabador**. Y si los que existen no están soportado en *GNU/Linux*, habrá que diseñar uno nuevo, probarlo y liberarlo, para que otros los puedan utilizar. Es lo que hemos hecho en este trabajo.

4.2. Tipos de grabadores

En el mercado existen diferentes grabadores para los microcontroladores *PIC*. Los esquemas de muchos de ellos están disponibles para que los usuarios los puedan construir. Nos encontramos con tres tipos:

1. **Grabadores por el puerto paralelo**. Como por ejemplo el EPIC+[26]. En Estos programadores, el software de grabación tiene que acceder al puerto paralelo, a bajo nivel, para implementar el protocolo de grabación. No hay una forma estándar de hacerlo y depende del sistema operativo utilizado. El *software* no es portable. Además, existe el riesgo de que funcione correctamente en unos ordenadores, pero no en otros que vayan a otra velocidad.
2. **Grabadores por el puerto serie**. Por ejemplo el JDM[27]. Están más extendidos, son más pequeños y toman la alimentación del propio PC. Aunque se conectan por el puerto serie, la comunicación **NO sigue el estándar RS-232**, sino que se utilizan las señales auxiliares *CTS* y *DTR* para la transmisión de los datos y el reloj. El inconveniente es que **es muy dependiente del ordenador**, tanto del *software* como del propio equipo (En los portátiles normalmente no funciona).

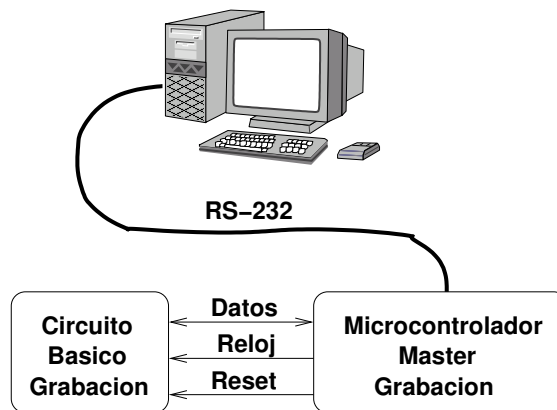


Figura 1: Diagrama de bloques del grabador

3. **Grabador autónomo.** Como por ejemplo el *ICD* de Microchip. El protocolo de grabación se implementa en un microcontrolador, dentro del grabador y NO en el PC. El *software* del PC envía los datos a grabador y éste gestiona el protocolo con el *PIC*.

El grabador construido es de tipo autónomo. El protocolo de grabación se implementa en un microcontrolador, que se comunica con el PC a través de comunicaciones serie RS-232 estándares, a 9600 baudios.

4.3. Protocolo de grabación

Los microcontroladores *PIC* se graban mediante el protocolo *ICSP* (*in circuit serial programming*), que permite el acceso a la memoria de programa, la *eprom* de datos y la palabra de configuración. Para realizar esta grabación, el *PIC* debe entrar en **modo monitor**.

Es un protocolo **serie síncrono**, con un hilo para los **datos** y otro para la señal de **relej**. La frecuencia máxima de transmisión de los bits es de 5MHz. Mediante el envío de comandos, se puede leer el contenido de la memoria de programa, almacenar nuevos valores, acceder a la palabra de configuración, lectura de la *eprom*, escritura en ella, etc. Se puede encontrar más información sobre la grabación de *PICs* en [28].

4.4. Diagrama de bloques

El diagrama de bloques del grabador diseñado se muestra en la figura 1. Está constituido por las siguientes partes:

- **Ordenador**, donde se ejecuta el software de grabación
- **Máster de grabación.** Dispositivo que se conecta al PC por el *RS-232* a 9600 baudios. Contiene el microcontrolador en el que se implementa el protocolo *ICSP*.
- **Circuito básico de grabación.** Placa con el *PIC* que queremos grabar. Puede ser desde una simple placa con un zócalo y unos pocos componentes, hasta la tarjeta entrenadora donde queremos probar nuestras aplicaciones. El circuito más sencillo se muestra en la figura 2. Obsérvese que son necesarias dos alimentaciones, una 5v y otra de 12v. Para entrar en modo monitor hay que aplicar una tensión de 12v en la pata MCLR.

4.5. Máster de grabación

El **Máster de grabación** hace accesibles los comandos del protocolo *ICSP* al *software* del PC, a través de un interfaz *RS-232*. El protocolo entre el PC y el *Máster* está descrito en [29]. Se puede implementar usando diferentes microcontroladores.

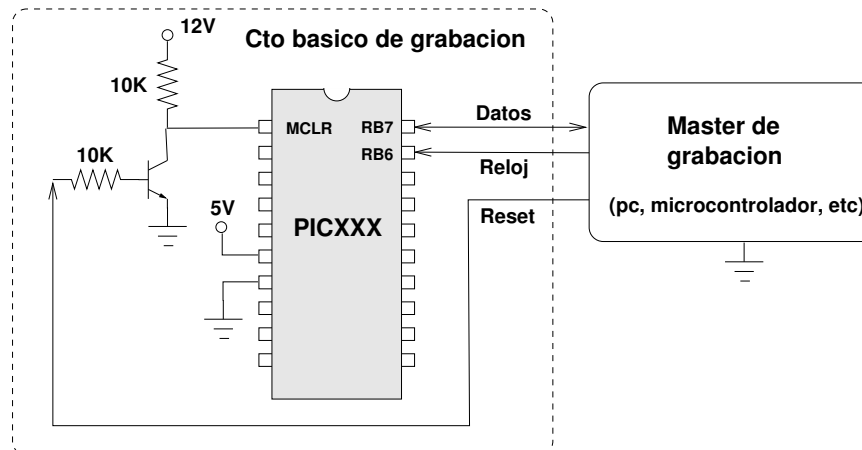


Figura 2: Un Circuito básico de programación

En la versión actual se implementó en un 6811, de motorola, usando la tarjeta CT6811[30] (ver figura 3). También se ha construido otro Máster con un PIC16F876A. Esto permite que los alumnos se construyan una tarjeta entrenadora para el PIC, que además les sirva como grabador. Se han hecho pruebas con algunos grupos del Laboratorio de Arquitectura[17].

4.6. Ventajas e inconvenientes

Las **ventajas** se deben a la utilización de comunicaciones serie estándar, a 9600 baudios. El grabador se puede utilizar desde cualquier ordenador o dispositivo que disponga de este tipo de comunicaciones: un PC, una PDA, otro microcontrolador, etc. El método de grabación también es independiente del sistema operativo empleado y del *software* de grabación.

La programación del *software* de grabación es sencilla, porque no requiere que el desarrollador tenga que conocer los detalles de bajo nivel de la grabación de los PICs. La aplicación debe leer el fichero ejecutable, construir las tramas necesarias y enviarlas por el puerto serie.

El principal **inconveniente** es la velocidad de grabación, que es más lenta que en otros grabadores. Está limitada por los 9600 baudios de las comunicaciones serie y por las tramas que se tienen que enviar y recibir. Por cada palabra a grabar (de 2 bytes), se intercambian 8 bytes (en el caso peor), por lo que para grabar un programa de 1K palabras, se intercambiarán 8192 Bytes, que a una velocidad de 9600 Baudios tardarán aproximadamente 7 segundos. Si añadimos verificación, el tiempo se duplicará: unos 14seg por K palabra.

5. El proyecto Stargate

El *Máster* de grabación tiene un microcontrolador en el que se ejecuta un programa servidor, que ofrece servicios al PC a través del RS-232. El servidor que permite grabar los PICs se denomina PICP[29]. No se trata de un servidor aislado, sino que forma parte del **proyecto Stargate**[31], donde se ha definido una arquitectura cliente-servidor para el acceso a servicios ofrecidos por microcontroladores, a través del RS-232. Según el *stargate* conectado, el cliente del PC accede a unos servicios u otros. Por ejemplo, posicionamiento de servos[32] para hacer prototipos de robots articulados, un servidor genérico[33] para el acceso a todo el mapa de memoria del microcontrolador, o el servidor PICP para grabar PICs[29].

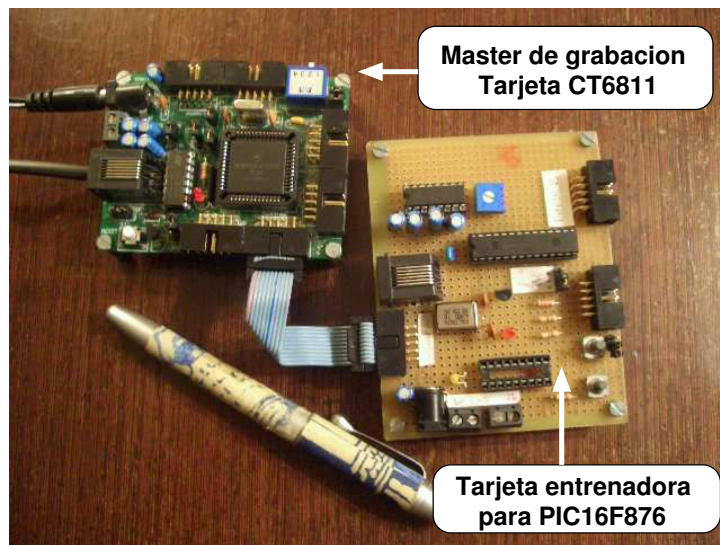


Figura 3: La tarjeta CT6811, usada como máster de grabación

6. Conclusiones

Los microcontroladores **PIC** están en auge, utilizándose en proyectos industriales, de investigación y para docencia. No tiene sentido que el diseñador sólo pueda utilizar una única plataforma para desarrollar aplicaciones con ellos. Parece más lógico que el diseñador utilice la que más le convenga, o la que normalmente use. Para ello, **hemos diseñado un grabador** para los microcontroladores **PIC** de la familia *16F87X* y *16F8x*, que se puede implementar usando cualquier microcontrolador que disponga de comunicaciones RS-232. Hemos definido un protocolo para la grabación y lo hemos implementado en una **tarjeta CT6811** y en una tarjeta prototipo con un **PIC16F876A**.

Se ha programado un cliente en Linux, que se comunica con el grabador y que permite cargar programas ejecutables en la flash. El *software* está bajo licencia GPL y toda la información sobre el grabador se ha liberado.

Además se han evaluado las herramientas **gputils** y **gpsim**, para ensamblar y simular en máquinas *GNU/Linux*. Tanto el grabador como el *software*, se han probado durante la impartición de la asignatura “Laboratorio de Arquitectura de computadores” en la UPSAM y podemos concluir que **es totalmente viable plantearse la docencia práctica sobre los microcontroladores PIC utilizando entornos GNU/Linux**.

7. Trabajo futuro

Dada la viabilidad de desarrollar aplicaciones para los *PIC* desde *GNU/Linux*, estamos trabajando en una tarjeta entrenadora, basada en hardware libre, para los *PIC*, a la que llamamos **Skypic**. Esta tarjeta sirve tanto como entrenadora, como grabadora y se podrá usar desde entornos *GNU/Linux* o *Windows*, el diseñador elige.

El software de descarga, **skypic-down**, se va a mejorar, añadiéndose la autodetección del *PIC* empleado y construyendo una interfaz gráfica, para *GTK2.0*. Con todo ello pretendemos disponer de nuestras propias herramientas libres para impartir docencia, talleres y cursos con el *PIC*.

Referencias

- [1] Empresa Microchip, fabricante de los microcontroladores PIC. [En línea] <http://www.microchip.com/>

- [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] A. Barroso, P. Tinoco, S. Salamanca, E. Romero. “Realización de Estructura Maestro/Esclavo con MicroncontroladoresPIC para el Control en Tiempo Mínimo de un Microbot Rastreador”. Seminario Hispabot 2003, Alcalá de Henares, Madrid, Mayo 2003.
- [5] Módulos G1v4 del robot Polybot, en el PARC. [En línea] <http://www2.parc.com/spl/projects/modrobots/chain/polybot/g1v4.html>
- [6] Tarjeta Micropic Trainer, desarrollada por la empresa Microsystems Engineering. [En línea] http://www.pacalaconcurso.com/archivos/paginas/micropic_trainer.htm
- [7] 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.
- [8] 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.iearobotics.com/personal/andres/proyectos/picmin/pic1.html>
- [9] Placas Fast-pic y Power-pic, diseñadas por Víctor Apéstigue Palacio. [En línea] <http://www.terra.es/personal5/cuchomen1/Micros/micros.htm>
- [10] Alejandro Alonso Puig, “Diseño de un servomotor controlado por bus I2C mediante microcontrolador PIC de gama media”. [En línea] <http://www.mundobot.com/tecnica/Svd01/Svd01.htm>
- [11] “Sistemas informáticos II”, en la Universidad de Huelva. [En línea] http://www.uhu.es/manuel_sanchez/docencia/tii234/tii234.html
- [12] “Componentes Electrónicos”, Universidad de Oviedo. [En línea] <http://directo.uniovi.es/catalogo/FichaAsignatura.asp?Asignatura=8693>
- [13] “Informática Industrial”. Universidad de Sevilla. [En línea] <http://icaro.eii.us.es/asignaturas/ii/index.php>
- [14] “Diseño de Sistemas Basados en Microprocesador”. Universidad Palmas gran Canarias. [En línea] <http://www.dis.ulpgc.es/basesdis/pdas/publicapda-cuerpo.asp?idasignatura=326>
- [15] “Laboratorio Sistemas Digitales”. Escuela superior de ingenieros de Bilbao. [En línea] <http://det.bi.ehu.es/asignaturas/teleco/lisistdig.html>
- [16] “Diseño automático de sistemas digitales”. ETS Ingeniería informática. Universidad de Granada. [En línea] http://www-etsi2.ugr.es/planes/ingenieria/quinto/disenio_automatgico_de_sistemas_digitales.phtml
- [17] “Laboratorio de Arquitectura de computadores”. Facultad de informática. Universidad Pontifica de Salamanca en Madrid. [En línea]: <http://www.elalejandre.net/Arquitectura/ArquitecturaP.htm>
- [18] Juan Antonio Zafra Costa. “Implementación en VHDL de los microcontroladores PIC-16/17”. Proyecto Fin de Carrera. ESI Telecomunicación. Universidad de Sevilla. Junio del 2000.
- [19] Core del PIC16F84 en verilog. John Clayton. [En línea] <http://www.opencores.org/projects/risc16f84/>
- [20] GPUTILS. Utilidades GNU para los PIC. [En línea] <http://gputils.sourceforge.net/>
- [21] GPSIM. Simulador GNU para los PIC. [En línea] <http://www.dattalo.com/gnupic/gpsim.html>

- [22] Juan González. “Charla/taller en el Chat, sobre Microcontroladores PIC y Linux [I]”. Octubre 2003. [En línea] <http://www.iearobotics.com/proyectos/charlas-irc/pic-linux/log-26-oct-2003.html>.
- [23] Programa skypic-down, para la grabación de Microcontroladores PIC. [En línea] <http://www.iearobotics.com/personal/juan/proyectos/skypic-down/skypic.html>
- [24] Programa SDCC (Small Device C Compiler). Compilador de C para diversos microcontroladores de 8 bits, entre ellos los PIC. [En línea] <http://sdcc.sourceforge.net/>
- [25] Entorno Pikdev para desarrollos basados en los microcontroladores PIC. [En línea] <http://pikdev.free.fr/>
- [26] Programador para PICs del tipo EPIC+. [En línea] <http://www.users.globalnet.co.uk/~jchap/tvpropie.htm>
- [27] Programador para PICs del tipo JDM. [En línea] <http://www.k9spud.com/jdm/>
- [28] Juan González, “Cuaderno técnico 4: Grabación de microcontroladores PIC”. [En línea] <http://www.iearobotics.com/proyectos/cuadernos/ct4/ct4.html>
- [29] Juan González, “Servidor PICP”. [En línea] <http://www.iearobotics.com/proyectos/stargate/servidores/sg-picp/sg-picp.html>
- [30] Tarjeta entrenadora CT6811. [En línea] <http://www.iearobotics.com/proyectos/ct6811/ct6811.html>
- [31] Juan González. “Proyecto Stargate”. Arquitectura cliente-servidor para el acceso a servicios ofrecidos por microcontroladores, a través del RS-232. [En línea] <http://www.iearobotics.com/proyectos/stargate/index.html>
- [32] Juan González, “Servidor SERVOS8”. Servidor para el posicionamiento de servos. [En línea] <http://www.iearobotics.com/proyectos/stargate/servidores/sg-servos8/sg-servos8.html>
- [33] Juan González, “Servidor Genérico”. Servidor para el acceso al mapa de memoria de microcontroladores de hasta 64KB. [En línea] <http://www.iearobotics.com/proyectos/stargate/servidores/sg-generic/sg-generic.html>