

# DISEÑO DE UN LABORATORIO DE ROBÓTICA AUTÓNOMA

G. GLEZ. DE RIVERA, R. RIBALDA Y J. GARRIDO  
*Departamento de Ingeniería Informática. Escuela Politécnica Superior.  
Universidad Autónoma de Madrid. España*

*En este trabajo se presenta una nueva plataforma hardware/software destinada a la realización de prácticas de robótica autónoma que se está aplicando actualmente en titulaciones de Ingeniería Informática y de Telecomunicación, pero que es lo suficientemente abierta en su concepción y manejo para ser utilizada en otro tipo de estudios superiores. El diseño abierto de esta plataforma permite al alumno centrarse, bien en el desarrollo de algoritmos y pruebas de software, utilizando cualquier lenguaje y bajo cualquier SO o bien centrarse en el diseño e implementación de nuevos sistemas periféricos sensores o de control.*

## 1. Introducción

El presente artículo describe el desarrollo de un laboratorio para la realización de prácticas de robótica en asignaturas impartidas en las titulaciones de Ingeniería Informática e Ingeniería de Telecomunicación de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid, utilizando una nueva plataforma de desarrollo. Esta plataforma es la evolución de un proyecto original a lo largo de una serie de años. La asignatura nace en el curso 1999-2000 y el objetivo principal fue diseñar una tarjeta de control con la que manejar un sencillo robot. Debido al perfil de los estudiantes (hasta este curso sólo Ingeniería Informática) en el que las áreas de electrónica y diseño no están muy desarrolladas, eran pocos los que conseguían un diseño operativo. Para el curso siguiente se diseñó un sistema completo de desarrollo, denominado GP\_Bot [1], basado en un microcontrolador de Motorola, en el que cada grupo de alumnos contaba con una unidad. Esta novedad permitió, ahora sí, que todos los alumnos pudieran diseñar un robot, lo programaran y desarrollaran diferentes algoritmos en él. A lo largo de los diferentes cursos el diseño ha ido evolucionado y últimamente se había quedado algo pequeño, por lo que este curso 2005-2006 se ha introducido una nueva plataforma, bastante más completa y ambiciosa, denominada GdRBot [2], [3] y [4]. Así mismo será utilizada por alumnos de la titulación de Ingeniería de Telecomunicación, de reciente implantación en nuestra Escuela.

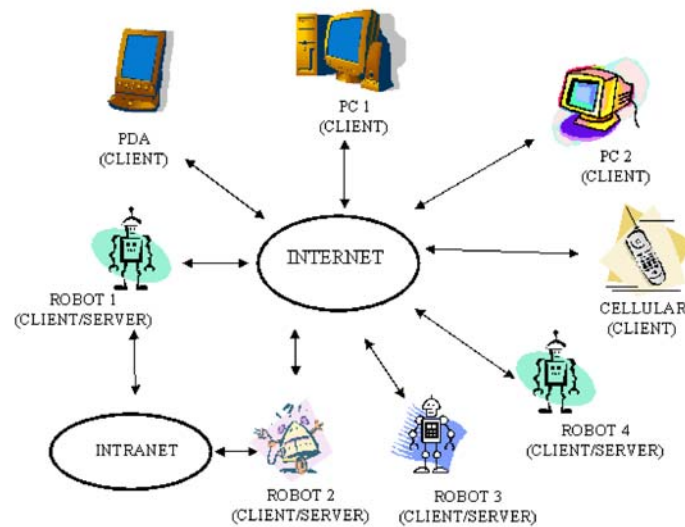
## 2. Descripción de la Plataforma GdRBot

La plataforma GdRBot, Fig. 1, consiste en una serie de robots genéricos que pueden ser controlados tanto de forma remota como local por un conjunto heterogéneo de dispositivos. Así mismo, permiten la adición de nuevos elementos sensores o actuadores a los robots de manera muy sencilla. El gran aporte de la plataforma es que el alumno puede crear algoritmos complejos y ejecutarlos en un robot, desde cualquier lenguaje de programación y prácticamente sin ninguna limitación, del mismo modo que programaría un simulador.

La plataforma está compuesta por dos elementos bien diferenciados: Cliente y Servidor. Ambos elementos se comunican vía XML-RPC [5]. Se ha elegido XML-RPC debido a que es un sistema de comunicación inter-procesos muy simple, fácil de depurar y está basado en estándares muy conocidos, como son HTTP y XML. En la actualidad XML-RPC posee más de 100 implementaciones oficiales para diferentes lenguajes de programación, desde Lisp hasta Java pasando por C. Así mismo XML-RPC está posicionándose de forma muy relevante en lo que se está empezando a denominar Web-2.0.

El servidor es básicamente un CGI colocado sobre un servidor Web cualquiera, todo esto ejecutándose en el robot. En la actualidad, las pruebas se han realizado con Apache [6] (un servidor Web complejo) y Boa (un servidor mucho más ligero). Mediante este CGI se accede de una forma estándar tanto a los sensores como a los actuadores del robot. Así mismo puede darnos información

sobre el sistema, estadísticas de uso, gestión de accesos... Por último, nos podrá dar información reflexiva, esto es, información acerca de las funciones existentes y los parámetros de las mismas. Este tipo de información es imprescindible si estamos frente a una red de robots heterogénea.



**Figura1.** Diagrama de bloques de la Plataforma propuesta

El cliente posee una implementación de XML-RPC que le permite comunicarse con el servidor. Puede correr fuera del robot o incluso dentro de él (para proporcionar información acerca de otros robots). El cliente no tiene por que ser un PC, puede correr en cualquier dispositivo con una pila TCP-IP y conectividad con el robot.

Puesto que todo funciona sobre TCP-IP, pueden ser usadas múltiples capas físicas en función de las necesidades específicas del proyecto. Destacan:

- Ethernet
- Bluetooth
- Puerto serie y paralelo
- USB
- GPRS
- UMTS
- Wifi

Por último, destacar que la propia arquitectura del sistema fuerza a una separación muy fuerte entre el cliente y el servidor, permitiendo una fácil movilidad de los mismos entre diferentes dispositivos de forma transparente.

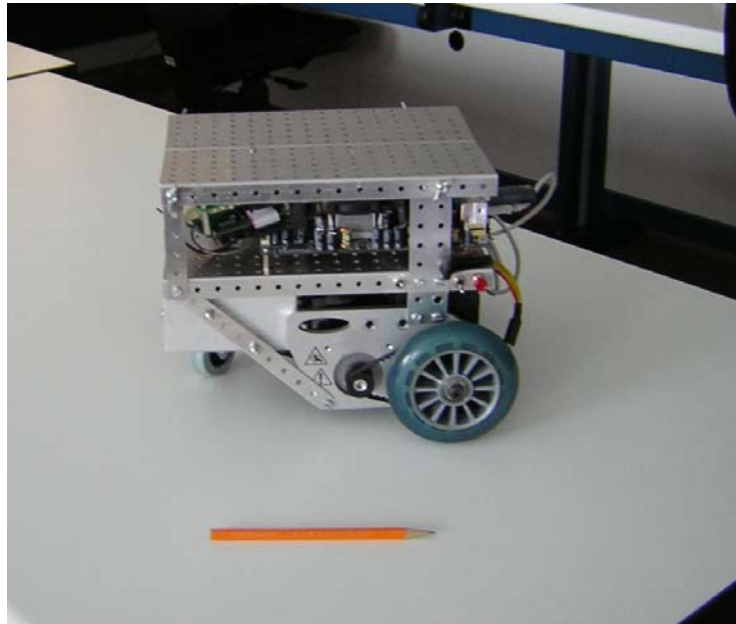
Puesto que los robots poseen una gran cantidad de elementos hardware, muchos de ellos complejos, se necesita un sistema operativo central capaz de operar con ellos. Para esta tarea se ha decidido usar Linux [7] como sistema operativo, puesto que soporta una gran cantidad de arquitecturas hardware y, al poseer el código fuente del mismo, se puede llegar a adaptar a otro tipo de arquitecturas no inicialmente soportadas. El hecho de estar trabajando siempre con el mismo sistema operativo ha permitido que el trabajo de implantación en diferentes arquitecturas haya sido mínimo.

### **3. Dispositivos Hardware actualmente integrados con la plataforma**

Al alumno se le brinda la posibilidad de usar distintos tipos de robots completamente heterogéneos en función de la tarea que desee realizar. Cada uno de ellos está basado en diferentes arquitecturas hardware y responden a necesidades también distintas.

### 3.1 Arquitectura EPIA

Como elemento principal de control, cada robot posee una placa base VIA EPIA modelo TC10000 a 1 GHz, con 512 Megabytes de RAM [10], suficiente como para ejecutar algoritmos complejos. En la parte de comunicaciones posee tarjeta Ethernet, Wifi, Bluetooth, Modem GSM y puede ampliarse con cualquier dispositivo soportado por Linux.



**Figura 2.** Plataforma Hardware basada en la Placa Base VIA-EPIA

En cuanto a los sensores utilizados, incluye cámara de vídeo, medidores de distancia por ultrasonidos y por infrarrojos y diversos pulsadores. La mayor parte de los sensores se controlan de manera distribuida a través de las tarjetas GP\_Bot [1], conectadas con un protocolo serie. Con esto se logra independizar al control central del procesado de cada sensor. Dicho control central tan sólo da las órdenes y recibe los resultados. Esto también permite incluir cualquier sensor con la única condición de cumplir con un flexible protocolo.

Por último en la parte de actuación, cuenta con dos ruedas motrices y una “loca”. Las ruedas motrices son movidas por sendos motores paso a paso, controlado por un procesador dedicado. Todo esto está montado sobre una plataforma de aluminio y alimentado por una batería de plomo de 12 voltios. Pese a la gran cantidad de dispositivos que posee el sistema, se puede montar de forma muy compacta en un tamaño de 20x30 cm aproximadamente, según se puede ver en la Fig. 2.

### 3.2 Arquitectura NSLU2

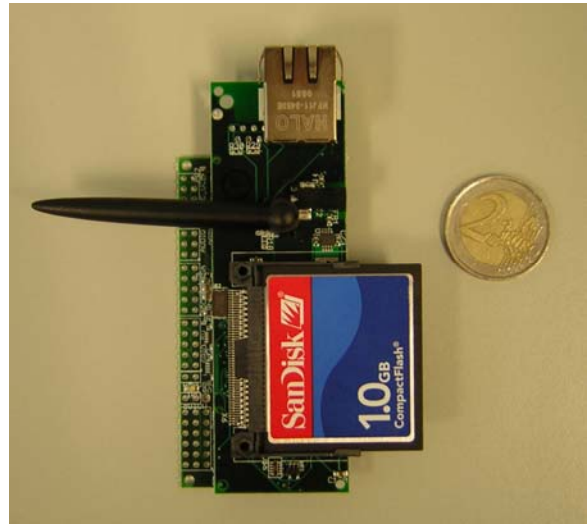
En este caso el elemento principal de control es un dispositivo de Linksys denominada NSLU2 [11], Fig. 3. Este dispositivo posee un micro ARM en su interior y en un principio se diseñó para compartir discos duros USBs por la red. Su precio es muy reducido (unos 80 Euros) y puede encontrarse en cualquier tienda de informática. Así mismo su diseño es muy compacto. De fábrica, viene con un pequeño Linux en su interior, dedicado específicamente a la tarea antes descrita. Sin embargo, se puede reprogramar para incluir una distribución de Linux completa, en la que alojar la plataforma GdRBot.

Es un dispositivo muy económico y compacto, con una potencia de procesado nada despreciable, equivalente incluso a un Pentium III. Como contrapartida, únicamente posee un puerto ethernet, y dos

puertos USBs, por lo que no es muy flexible a la hora de añadirle sensores, motorización, etc... Por suerte, el número de dispositivos con conectividad USB ha crecido de forma espectacular.



**Figura 3.** Plataforma basada en Arquitectura NSLU



**Figura 4.** Plataforma basada en la Arquitectura Gumstix

### 3.3 Arquitectura Gumstix

Esta es la arquitectura más novedosa y compacta, Fig. 4. El sistema de control es una tarjeta gumstix 400 connex bt [12], una potente micro-tarjeta con un microprocesador XScale de Intel a 400 Mhz y 64 MB de RAM [13]. Viene pre-instalada con Linux y tiene unas dimensiones de 80x20x6,3 mm.

Se le han incorporado dos tarjetas de expansión. La primera (netcf) permite su conectividad vía Ethernet, así como el uso de tarjetas Compact Flash. Recordar que existen en el mercado tarjetas Wifi en este formato. La segunda (audiostix) dispone de un microcontrolador Atmel con convertidores A/D, salidas PWM, salidas y entradas digitales, puertos serie, etc.

El conjunto no supera los 200 Euros y proporciona un sistema espectacularmente pequeño a la vez que potente y flexible. Gracias al microcontrolador se puede controlar casi cualquier tipo de sensor.

## 4. Desarrollo del curso

Para superar el laboratorio hay que realizar una serie de prácticas. Se propone un itinerario básico, con el que se trabajan todos los elementos, aunque no muy en profundidad. Los alumnos pueden proponer otros itinerarios en función de sus preferencias, de manera que profundicen más en algunos temas. El itinerario básico propuesto es el siguiente:

- *Práctica 0. Introducción a la plataforma GdrBot.* Sobre un robot básico, se propone al alumno controlarlo desde un lenguaje de alto nivel como Java, o C#. El alumno aprenderá a realizar llamadas XML-RPC y trabajará en profundidad con la iteración de distintas redes.
- *Práctica 1. Adición de un nuevo sensor, el sistema de desarrollo GPBot.* Se propone al alumno la incorporación de un nuevo sensor a través del sistema de desarrollo GPBot. El alumno aprenderá a programar un microcontrolador.
- *Práctica 2. Integración del nuevo sensor con la plataforma.* Una vez el alumno dispone de un sensor, este se integra dentro del robot, para lo cual elaborará un CGI accesible desde web. El alumno aprenderá a programar funciones XML-RPC y la creación de CGIs.

- *Práctica 3. Trabajo colaborativo.* Se propone la creación de un programa en el que interaccionen varios robots para la realización de una tarea usando el sensor creado en la práctica anterior. El alumno verá la complejidad de la creación de dichos algoritmos.

## 5. Conclusiones

Mediante el uso de la plataforma GdRBot presentada, el alumno que curse el laboratorio de robótica tendrá la suficiente base teórica y práctica para afrontar problemas complejos con robots, y poder desarrollarlos desde cualquier sistema operativo o lenguaje de programación. Debido al diseño de la plataforma, completamente terminada pero abierta, el alumno puede centrarse bien en el desarrollo, pruebas de software y algoritmos o bien en diseños hardware y de control.

## Referencias

- [1] 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*. Tecnologías Aplicadas a la Enseñanza de la Electrónica (TAAE'02). Pg 67-70 Univ. de las Palmas de Gran Canaria, España 2002
- [2] G.Glez. de Rivera, R. Ribalda, K. Koroutchev, J.Colas and J.Garrido. *Hardware Independent Architecture for Autonomous Collaborative Agents*. Proceedings of 2nd International Conference on Informatics in Control, Automation & Robotics (ICINCO-2005), Pp: 459-462. Barcelona, Spain. September 2005.
- [3] G.Glez. de Rivera, R. Ribalda, J.Colas and J.Garrido. *A Generic Software Platform for Controlling Collaborative Robotic System using XML-RPC*. Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2005). Pp 1336-41. IEEE CAT. N°: 05TH8801C. ISBN: 0-7803-9047-4. Monterrey (USA). July 2005
- [4] G. Glez. de Rivera, R. Ribalda, J. Colás y J. Garrido. *Plataforma genérica para desarrollos basados en agentes móviles*. Actas de Ubiquitous Computing & Ambient Intelligence (UCAmI-05). Pp: 363-369. Granada. Septiembre 2005
- [5] <http://www.xmlrpc.com>
- [6] Apache Web Server Project, <http://httpd.apache.org>
- [7] <http://www.linux.org>
- [8] Debian GNU/Linux, <http://www.debian.org>
- [9] GNU C Library, <http://www.gnu.org/software/libc>
- [10] Via Epia Motherboards, <http://www.via.com.tw/en/products/mainboards>
- [11] Linksys NSLU2, <http://www1.linksys.com/products/product.asp?prid=640>
- [12] Gumstix Home Page, <http://www.gumstix.com>
- [13] Intel XScale Technology, <http://www.intel.com/design/intelxscale>

### Otras referencias consultadas

- Building and Testing gcc/glibc cross toolchains, <http://www.kegel.com/crosstool>
- Alexander Wolfe. Tiny Linux Computer Has High Hopes For Robotics Apps. TechWeb News. 2005
- Vlado Handziski, Andreas Kopke, Andreas Willig, Adam Wolisz. TWIST: A Scalable and Reconfigurable Wireless Sensor Network Testbed for Indoor Deployments. TKN Technical Reports Series. 2005