

Programando robots con software libre

Vicente Matellán Olivera

4 de noviembre de 2004

Programando sistemas

El software libre en general y GNU/Linux en particular son opciones habituales para la programación de aplicaciones en ordenadores personales o grandes sistemas, pero también son una plataforma de desarrollo muy popular para la programación de sistemas empotrados, en particular para la programación de robots. De hecho, en la comunidad de investigación en robótica el software libre se ha convertido en estándar.

Esto probablemente no sea muy importante para el usuario medio de informática a día de hoy, pero lo será a corto plazo por la aparición de robots en nuestras vidas. Hoy en día ya podemos comprar corta-césped autónomos, por ejemplo, el RL-500¹; carritos domésticos como el Cye² capaz por ejemplo de llevar una bandeja de una habitación a otra; mascotas robóticas como el famoso perrito AIBO³ del que Sony ha vendido varios centenares de miles de unidades; o la muy barata (por menos de 200 Euros) aspiradora robótica Roomba⁴, que ha sido regalo tecnológico de moda el año pasado en los EE.UU. etc. Sin embargo, ninguno de ellos ha alcanzado los miles de unidades vendidos por LEGO MindStorms, el juguete programable que permite hacer diversas construcciones y en el que me centraré en este artículo.

Como es lógico, todos estos robots están controlados por un ordenador, o al menos un microcontrolador, lo que los hace susceptibles de ser “programados”. Es aquí donde aparece la importancia del software libre. Una vez más queremos poder inspeccionar que hacen nuestros robots, que estarán conectados a la red, tendrán cámaras, etc. Querremos poder modificar sus programas, adaptarlos, etc. Es decir, nos interesa disponer de software libre también para nuestros robots. Un ejemplo claro de que el software libre está también incorporándose a este mundo es la gran cantidad de software libre desarrollado alrededor de los LEGO Mindstorms, como decía antes, el robot más difundido hasta la fecha.

LEGO Mindstorms, tu primer robot

LEGO Mindstorms es un “juguete” diseñado como un producto de consumo, lo que le dota de una gran calidad y además de una enorme flexibilidad. Permite construir una enorme variedad de robots, desde un brazo hasta un robot recolector, sin necesitar ninguna soldadura. Además, permite

¹<http://friendlyrobotics.com/>

²<http://www.personalrobots.com>

³<http://www.sony.com/aibo>

⁴<http://www.irobot.com>

una gran libertad a la hora de colocar los sensores y los motores de los robots. El “cerebro” de estos robots está basado en un microprocesador Hitachi H8/300 con 32 Kbytes de RAM. Dispone de tres puertos de salida para conectar motores y tres puertos de entrada para conectar sensores, además de un puerto de infra-rojos para comunicarse con el ordenador que sirve de plataforma de desarrollo. Los sensores disponibles son el de luz ambiente, de luz reflejada, de colisión, de temperatura y de rotación.

Los LEGO Mindstorms se venden como *kits* completos formados por más de 700 piezas tradicionales de LEGO, los tradicionales ladrillos de plástico. Cada *kit* además incluye dos motores, dos sensores de contacto, uno de luz y el bloque del micro-procesador que se denomina habitualmente “ladrillo” o RCX (con la memoria y los puertos de entrada/salida). Además, el *kit* incluye un entorno gráfico de desarrollo y el equipo necesario para descargar el software desde un ordenador personal al ladrillo.

El entorno para el desarrollo de programas incluido en el *kit* Mindstorms 1.5 es un lenguaje gráfico de programación pensado fundamentalmente para niños o adultos sin ninguna preparación informática. Se trata de un entorno muy limitado, y que además no es software libre. Afortunadamente, los LEGO Mindstorms han recibido mucha atención por parte de la comunidad de software libre, con lo cual han aparecido varias opciones diferentes para su programación. Entre ellas destacan dos, NQC y BrickOS, que se analizan en las siguientes secciones.

Creo que el LEGO Mindstorms constituye uno de los primeros intentos de difundir el mundo de la robótica a escala doméstica. De hecho se pueden considerar equivalentes en cuanto a difusión, a lo que fue la familia de los ZX de Sinclair en su día en el mundo de los ordenadores. Ya habían existido computadores que se vendían en forma de *kit*, como por ejemplo el MIT Altair 8800 (basado en el intel 8080). Igualmente han existido otros robots que han tenido difusión comercial, como por ejemplo el RugWarrior que todavía se comercializa, y que está basado en el chip 68HC11 de Motorola. Otro ejemplo es la tarjeta Handyboard⁵, diseñada específicamente para la construcción de robots, o la tarjeta española de Mibrobótica⁶ que usa el mismo chip. De igual forma, han existido y siguen apareciendo, muchos otros robots “en *kit*” basados en este y en otros chips. De hecho, se pueden comprar colecciones de este tipo en los kioscos.

Todos estos robots comparten con los primeros ordenadores el “espíritu de garaje”, es decir, el programador tiene que montar su hardware y luego hacer sus programas. Este espíritu es atractivo para muchos de nosotros, pero a la vez hace que muchos potenciales usuarios queden fuera del mundo de robótica por la complejidad y dedicación que implica. En general tener que usar un soldador, un polímetro, etc. son condiciones inaceptables para el gran público.

Por supuesto, existen robots móviles programables que se venden como productos comerciales finales. Estos robots generalmente están destinados a los centros de investigación y su precio es tan elevado (casi siempre por encima de los 10.000 Euros) que los hacen inalcanzables para el gran público. Ejemplos típicos de estos robots son los de Real World Interface⁷ con robots como el B21, Magellan o la familia ATRV; o los de ActiveMedia Robotics⁸ con productos como el Pioneer, Peoplebot, o Amigobot. En esta categoría de robots para centros de investigación también hay algunos más pequeños y asequibles (alrededor de los 3.000 euros) como los suizos de K-Team⁹ (Khepera y Koala) o el australiano Eye-Bot¹⁰, pero aún así prohibitivos para el consumidor medio.

⁵<http://el.www.media.mit.edu/projects/handy-board/>

⁶<http://www.microbotica.es>

⁷<http://www.irobot.com/>

⁸<http://www.activrobots.com/>

⁹<http://www.k-team.com>

¹⁰<http://www.ee.uwa.edu.au/~braunl/eyebot/>

El siguiente estadio tras los garajes en la evolución del mercado de los robots dirigido al gran público es el paso del *kit* de auto-montaje, al producto final, el equivalente al del ZX en el mundo de los ordenadores. En este paso el caso de los LEGO Mindstorms puede resultar engañoso, de hecho es un producto comercial que precisamente está pensado para que los usuarios “construyan” sus modelos. Sin embargo, el aspecto electrónico del producto es el de producto final, es decir, no hay que soldar ningún elemento, simplemente comprar unas pilas y encenderlo.

De la misma forma, la parte de programación del robot es un producto final, se entrega un entorno de programación completamente visual muy orientado a los niños, que hace que cualquier usuario, sin ninguna formación informática pueda programar fácilmente el robot. Sin embargo, se trata de un **software propietario**.

Programando el LEGO Mindstorms

¿Cuál es el motivo de la gran cantidad de software desarrollado para el LEGO Mindstorms? El factor principal ha sido la permisividad de LEGO. Bien porque su mercado es la venta de sus “piezas”, es decir, del hardware, o bien porque creen que no hay más remedio que permitir que los *hackers* modifiquen tus productos sin son susceptibles de hacerlo; la realidad es que LEGO ha consentido la ingeniería inversa necesaria para que se pueda desarrollar software libre para su plataforma. Los entornos de programación más extendidos son NQC y BrickOs.

Algo parecido sucedió con el perro-máscota de Sony. Un *hacker* publicó las llamadas al sistema de su sistema operativo propietario (OPEN-R), lo cual fue inicialmente perseguido por Sony hasta que decidió publicar un entorno de desarrollo: openr.aibo.com. En este caso Sony no ha liberado su sistema operativo, pero si se está constituyendo una comunidad de desarrolladores alrededor de sus robots, con lo que puede que en algún momento surga una alternativa libre.

Pero volvamos a las alternativas para programar el LEGO Mindstorms. La más extendida es NQC, cuyo acrónimo NQC significa *Not Quite C*, que traducido es algo así como “Un C no completo”. Es decir, se trata de un lenguaje sencillo con una sintaxis muy similar a C que se puede usar para programar el ladrillo. Es, desde luego, la alternativa más sencilla al lenguaje de programación basado en iconos arrastrables que proporciona LEGO.

NQC es software libre, distribuido bajo la Licencia MPL (*Mozilla Public License*). Sin embargo, usa el sistema operativo original de LEGO, lo que hace que no se trate de un entorno libre “completo”. A pesar de estas limitaciones, NQC es un lenguaje muy popular entre los usuarios de baja formación informática, debido sobre todo a su simplicidad, pero también a la abundante documentación y a la existencia de herramientas como el *RCX Command Center* para MS-Windows(TM) que facilitan el desarrollo y la descarga de programas. En el caso de GNU/Linux, se dispone de un compilador que produce código directamente descargable en el robot, pudiendo utilizarse como editor cualquiera que edite texto ASCII.

Sin embargo, si se quieren realizar programas que requieren cierto almacenamiento de datos, por ejemplo crear un mapa, o simplemente que sean más grandes, NQC no es la opción más adecuada y hay que utilizar un lenguaje de programación real (C, C++, Forth, Ada, etc.) lo cual obliga a cambiar el sistema operativo del ladrillo y en consecuencia a usar BrickOS.

BrickOS¹¹ es un sistema operativo libre para el LEGO Mindstorms, diseñado e implementado fundamentalmente por Markus Noga, y que inicialmente se llamó LegOS¹². Comparado con el

¹¹<http://www.lugnet.com/robotics/rcx/legos/>

¹²<http://www.noga.de/legOS/>

sistema operativo original de LEGO, ofrece muchas ventajas además de mejores prestaciones y mayor flexibilidad y lo que es más importante desde mi punto de vista: se trata de software libre.

BrickOS es únicamente el sistema operativo, lo que quiere decir que se necesita el soporte de un compilador capaz de generar código para el H8 a partir del lenguaje de programación que deseemos y para el que existan las librerías adecuadas de BrickOS. El entorno de programación bajo GNU/Linux incluye el compilador de C de GNU (*gcc*) compilado como cruzado para el Hitachi H8, para lo que hace falta usar las *binutils*. La distribución para GNU/Linux de LegOS incluye varias herramientas que permiten descargar el código de forma dinámica, descargar el *firmaware* o sistema operativo, así como varios ejemplos. Existen paquetes Debian, y de otras distribuciones, de forma que es muy sencillo de instalar.

Alrededor del sistema operativo BrickOS se han desarrollado múltiples herramientas auxiliares, como por ejemplo simuladores que hacen más fácil la depuración al permitir ejecutar programas en la propia plataforma de desarrollo usando un depurador tradicional de GNU/Linux como por ejemplo *gdb*. Ejemplos de estas herramientas son *emulegos*¹³, que permite simular las entradas y salidas; *LegoSim*¹⁴, otro simulador; etc. Existen también herramientas que no son para programadores, como por ejemplo *LDraw*¹⁵ que permite crear escenas virtuales de piezas LEGO y *renderizarlas*, haciendo incluso animaciones,

En resumen, el software libre no sólo está colonizando los ordenadores personales con GNU/Linux, Firefox u OpenOffice, está entrando con mucha fuerza en todos los entornos programables, y los robots no son una excepción. Pronto podemos encontrarnos al robotito de Bilo¹⁶, de la genial tira cómica de Javier Malonda en nuestra mesa ejecutando software libre...

©2004 Vicente Matellán Olivera. vmo@barrapunto.com

Se otorga permiso para copiar y distribuir este documento completo en cualquier medio si se hace de forma literal y se mantiene esta nota.

¹³<http://emulegos.sourceforge.net/>

¹⁴<http://moss.csc.ncsu.edu/~mueller/legosim/>

¹⁵<http://www.ldraw.org/>

¹⁶<http://tira.escomposlinux.org>