

# ¡Toma el control de tu ladrillo MINDSTORMS™!

Por Oton Ribic

Aunque no hay duda de que el método original de LEGO® de programar MINDSTORMS a través de «bloque» es bastante inteligente y divertido, muchos constructores que desean hacer proyectos más complejos se dan cuenta de que puede llegar a ser poco práctico, con estructuras que abarcan cientos o incluso miles de módulos. En una situación así, controlar ladrillos programables (pBricks) con un lenguaje de programación estándar es más sencillo.

Se han propuesto muchas soluciones para este fin, incluyendo un firmware adaptado y lenguajes de programación completos, lo que resulta en proyectos interesantes e impresionantes. Sirven bien, reciben actualizaciones regulares y son un buen comienzo. Sin embargo, para aquellos que quieren controlar cosas a un nivel más técnico, o que quieren comprender cómo funcionan los sistemas de 'ordenación' de los pBricks, revelaremos algunos detalles en esta mini serie de artículos. También supondremos que el usuario no quiere dejar de usar el firmware por defecto del NXT y EV3.

No olvides que para la gran mayoría de propósitos, quedarse con el entorno de programación original de MINDSTORMS o con una versión adaptada, sirve perfectamente; ir más allá es simplemente una cuestión de curiosidad, no de necesidad. Por lo tanto, lo que sigue puede servir como curiosidad tanto como para aplicaciones prácticas. Si te interesa eso último, merece la pena mencionar que todo esto se puede hacer de forma bastante sencilla en cualquier lenguaje de uso común (el autor lo hizo en Python y otro fan de LEGO® en C++).

## Disecionando la comunicación

Comencemos por investigar los métodos que el pBrick emplea para comunicarse con el mundo exterior. Aunque se puede usar un cable USB, es más común y práctico usar Bluetooth, y hay algunas otras opciones que son más exóticas. Pero en todos los casos, podemos comenzar por decir que todas las comunicaciones se efectúan en conjuntos de paquetes digitales (literalmente colecciones de bytes) que se envían y reciben desde estos aparatos a través de cualquiera de estos métodos. El objetivo de cualquier cosa empleada para controlar sistemas MINDSTORMS es poder preparar, enviar, esperar, recibir e interpretar los mensajes contenidos en estos paquetes.

Estos mensajes contienen toda la información que se intercambia entre el pBrick y su controlador - en la mayoría de los casos un ordenador. A veces envían datos numéricos, o instrucciones y ocasionalmente pide al receptor que envíe datos de vuelta (por ejemplo comprobando el valor de un sensor). Y también existe un sistema para contestar a estos mensajes con confirmaciones de su correcta recepción, lo cual es esencial si queremos que el sistema sea fiable.

## NXT vs EV3

El concepto general de intercambiar mensajes con paquetes de datos aplica tanto al NXT como al EV3. Sin embargo, difieren en un segmento, que se volverá más importante cuando llegemos a la construcción de mensajes. A saber: el EV3 con su firmware de fábrica permite que el dispositivo controlador mande las instrucciones de manera directa, independientemente del estado del pBrick, siempre que al menos esté encendido, y los ejecuta. Cualquier 'bloque' que podamos programar en el software oficial se puede envolver en un mensaje para que el EV3 lo ejecute.

Por otro lado, el NXT con su firmware de fábrica no lo permite de forma tan sencilla, al menos no sin usar algunos trucos que caen fuera del ámbito de estos artículos. Pero lo que el NXT sí admite es ejecutar un programa preparado en NXT-G estándar para recibir mensajes por Bluetooth e interpretarlos de la manera requerida. Por ejemplo, interpretar correctamente un mensaje



*Bluetooth es una manera práctica y sencilla de conectar ordenadores u otros dispositivos a un ladrillo programable. Dispositivos más antiguos suelen necesitar un adaptador para habilitarlo, pero está incluido en la mayoría de equipos modernos.*

que declara que son instrucciones para girar un motor, especificando el motor, la cantidad de giro la dirección y, finalmente, la potencia. Por supuesto, estos mensajes se tienen que construir cuidadosamente para que se interpreten correctamente, aunque no es tan difícil como suena mientras uses algunos de los comandos más comunes.

Basta decir por ahora que estos dos paquetes de MINDSTORMS son diferentes en ese aspecto. El Ev3 es más flexible y permite más control, pero también es más complejo. El NXT es más sencillo, pero menos potente. Aun así, para el caso común de girar un motor en un ángulo determinado, funciona perfectamente.

Hasta aquí la introducción - en el siguiente artículo veremos cómo construir mensajes sencillos y mandarlos a los pBricks.  
#



*Algo más antiguo, el NXT es más sencillo que el EV3, pero soporta el control de la mayoría de funciones con el firmware por defecto.*



*El Ev3 es más nuevo y permite un control más profundo del pBrick, pero también es más complejo.*