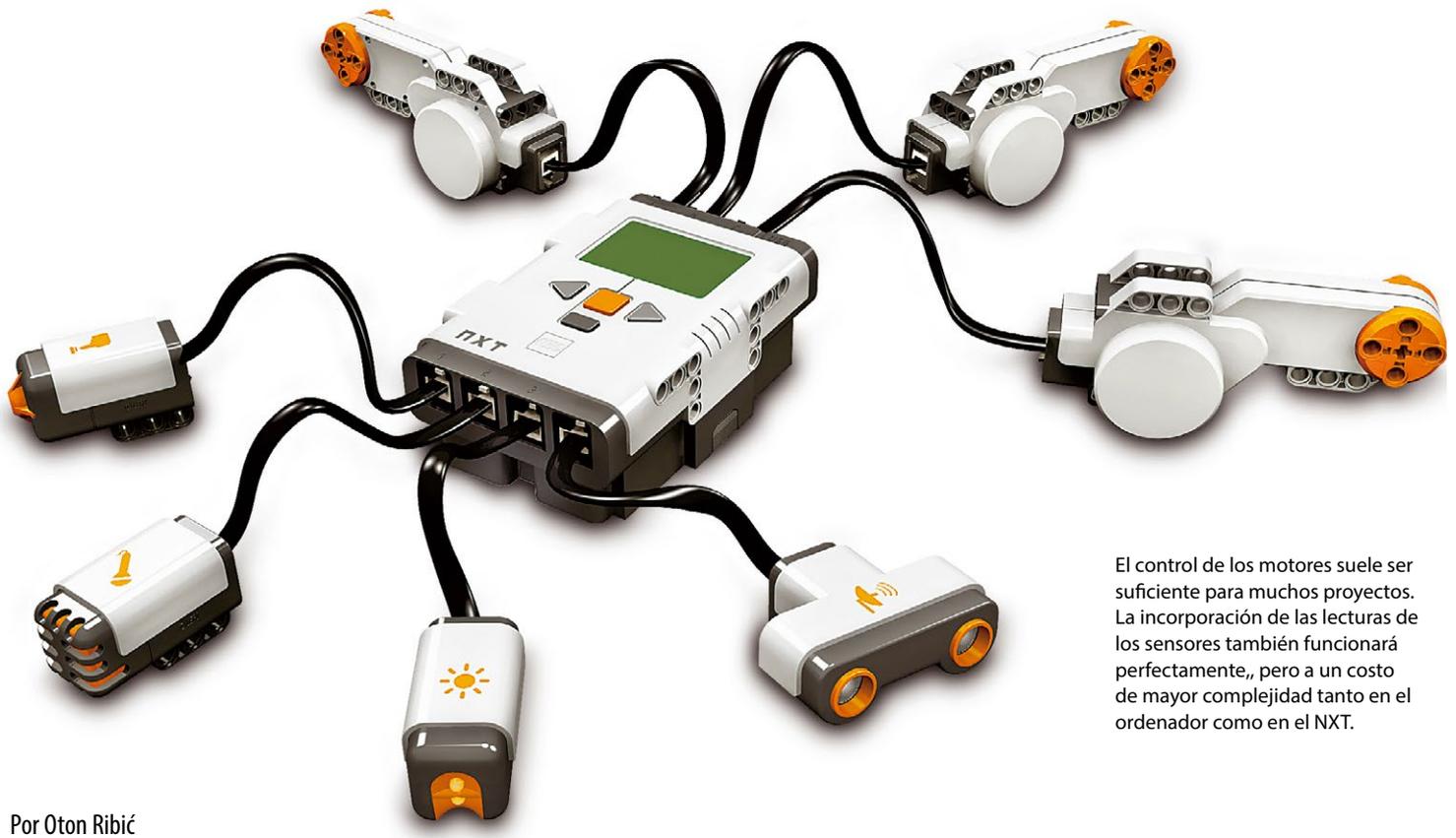


De vuelta al viejo NXT



El control de los motores suele ser suficiente para muchos proyectos. La incorporación de las lecturas de los sensores también funcionará perfectamente,, pero a un costo de mayor complejidad tanto en el ordenador como en el NXT.

Por Oton Ribić

Tras haber navegado por las complicadas aguas del control del EV3 vía Bluetooth en artículos anteriores, en este último episodio de la ya larga serie Mindstorms Control volveremos a la algo anticuada, pero aún muy vital, NXT. También soporta la comunicación Bluetooth, y si tu hardware lo permite, no hay dificultad en combinar NXT y EV3 Smart Bricks a la vez, o incluso múltiples de cada tipo simultáneamente.

En primer lugar, debemos enfatizar que el NXT es algo muy diferente al EV3 en lo que se refiere al control inalámbrico. Mientras que el EV3 nos permitió compilar mensajes para hacer que “obedeciera órdenes” directamente, en el caso del NXT es más sencillo comunicarse con un programa dedicado que se ejecuta en él, el cual escucha los datos entrantes y ejecuta los comandos en consecuencia. Esto hace las cosas más complejas, ya que el programa tiene que estar ejecutándose en el ladrillo, pero hace la comunicación en sí misma un poco más fácil. Por supuesto, uno puede decidir renunciar a toda la idea y flashear el NXT con un firmware para permitir el control directo, pero eso es un enfoque bastante complejo. Como se ha dicho antes, en estas series nos centraremos sólo en el firmware de serie.

Programar la escucha y la ejecución en el lado del dispositivo

Por lo tanto, el primer paso será configurar un programa dentro del dispositivo. Puedes optar por uno de propósito general, o por uno que haga específicamente lo que necesitas para el proyecto en cuestión. Aquí vamos a ir por la versión de uso general, y puedes fácilmente adaptarlo más para las diferentes necesidades basadas en estos principios.

Al controlar los motores, hay tres valores esenciales que debemos transmitir al Smart Brick de NXT para cada movimiento: el motor a utilizar, la cantidad a girar y la potencia a aplicar mientras se gira. En teoría, se podría argumentar que la dirección es el cuarto parámetro, pero podemos incluirlo en el número que especifica la cantidad a girar, usando un número negativo o positivo.

Suponiendo que cargamos estos valores en el orden mencionado, el esquema de un programa sería algo parecido a lo siguiente:

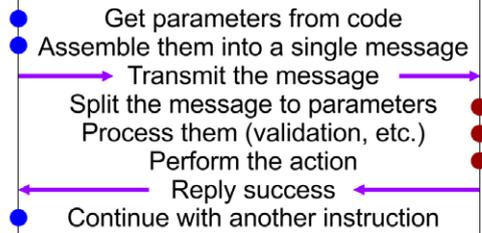
- Escuchar el Bluetooth hasta que los mensajes estén disponibles
- Cargar un “trío” de valores desde Bluetooth— Motor, Cantidad y Potencia
- Asignar el valor de Motor al parámetro correspondiente de la instrucción de

Motor en el NXT-G

- Comprueba si la cantidad está por debajo de cero, y asignar el resultado al parámetro de dirección del motor NXT-G
- Tome el valor absoluto de la cantidad y asignarlo al parámetro de ángulo de la instrucción en NXT-G
- Asignar el valor de Potencia que es un porcentaje al parámetro correspondiente de la instrucción NXT-G
- Realiza la rotación en sí misma y esperar hasta que termine (o no—si quieres usar un parámetro especial para eso!)
- Si quieres ser capaz de esperar a que la instrucción termine, envía un mensaje de vuelta vía Bluetooth
- Repetir desde el principio a menos que se reciba una instrucción de terminación, lo cual es opcional

Para darle una forma más comprensible, este es un programa NXT que escucha el Bluetooth, y cuando recibe valores, por ejemplo 2,-720,50, gira el segundo (B) motor dos veces en dirección contraria a la mitad de la potencia. Luego continúa escuchando un nuevo trío de números.

Esto es todo lo que se necesita para que el mando a distancia funcione desde el lado del



Una secuencia simplificada entre el ordenador y el NXT, comunicando vía Bluetooth.

dispositivo; sólo tienes que conectar tu ordenador al NXT Smart Brick a través de Bluetooth como se describe en los primeros artículos de esta serie, e iniciar el programa recién creado.

El control desde el ordenador

Ahora tenemos que pasar al lado del ordenador y configurar un programa que transmita los tres valores necesarios para realizar una instrucción de motor a la NXT, y opcionalmente esperar el reconocimiento.

Esta vez, el ensamblaje de los mensajes es mucho más sencillo que en el caso del EV3. Básicamente, tendremos que codificar cada uno de los parámetros del número de entrada en un código de 4 bytes, que representa un número de punto flotante. Revise la misma técnica comentada anteriormente en el caso del EV3, pero como recordatorio rápido, para el lenguaje Python por ejemplo, puede utilizar "struct.pack('f', number)", que devuelve los 4 bytes necesarios. Luego, agregue un byte cero al final, y la siguiente secuencia de bytes al comienzo: 5, 0, 0, 19, 10, 0, 1. Este es un encabezado que le dice al Ladrillo NXT lo que realmente se está comunicando.

Debería terminar con una secuencia de 12 bytes para cada número codificado. Por ejemplo, codificando el número 720 debería dar 5, 0, 0, 19, 10, 0, 1, 0, 0, 52, 68, 0. Finalmente, sólo hay que ensamblar el mensaje final poniendo los tres códigos de 12 bytes uno tras otro (concatenándolos) en el orden requerido. El mensaje de 36 bytes que especifica el motor, la cantidad de giro y la potencia puede ser enviado al NXT.

Cómo abordar la sincronización

El procedimiento descrito arriba debería hacer que tu NXT gire según las instrucciones. Sin embargo, el siguiente paso en el lado del ordenador es recibir una notificación una vez que el NXT haya completado su tarea, para que pueda proceder con la siguiente. Anteriormente en el programa NXT puede que hayas creado un mensaje de respuesta para el ordenador, y ahora es el ordenador el que tiene que

reaccionar ante él.

Independientemente del lenguaje de ordenador que hayas decidido utilizar, sigue disparando la secuencia de bytes 5, 0, 0, 19, 10, 0, 1 hacia el NXT, que comprueba la "bandeja de entrada" del dispositivo por si hay mensajes en espera y los transmite si los encuentra. Repite el proceso, y busca cualquier mensaje cuando llegue. Normalmente, una simple respuesta del dispositivo podría ser suficiente, pero se puede dar un paso más y hacer que el NXT envíe un valor específico, por ejemplo una cadena "DONE" o "Acknowledged", para estar doblemente seguro. Luego, en el lado del ordenador, deberías descifrar el mensaje recibido del NXT y comprobar si la cadena especificada está contenida en él. Si es así, todo está bien; si no, sabes que algo en el lado del NXT salió mal.



Nada impide conectarse y controlar múltiples ladrillos NXT y EV3 simultáneamente. Pero ten en cuenta que tendrás que configurar los puertos serie para cada uno de estos Smart Brick individualmente.

Ampliar más

Este ejemplo como tal debería darte una idea general de cómo funciona la comunicación, cómo el NXT interpreta el mensaje, actúa sobre él, e informa cuando termina. Para los proyectos básicos en los que se requiere un movimiento controlado, esto debería ser más que suficiente. Pero si necesitas más, vamos a sumergirnos en varios puntos más que pueden ser útiles o descubrir algunos atajos para ti.

Puede ser sensato ocuparse al menos de la validación mínima de datos en el lado de NXT. Si el parámetro de potencia de motor es demasiado bajo para incluso girar un motor, o si está

bloqueado para girar hasta el punto requerido, la instrucción esencialmente nunca terminará, y todo tu sistema de ajuste fino se atascará. Por lo tanto, puede que quieras añadir un umbral, o expandir tu programa NXT para reportar un mensaje especial (leído por el ordenador) si se ha excedido un determinado período de tiempo de espera, y la instrucción ha sido abortada.

Si los movimientos de tus motores parecen ser correctos pero crees que no se mueven los motores correctos, revisa si el motor A está realmente identificado con el número 1, porque en muchos idiomas puedes especificarlo como cero, lo cual no funcionará.

Este esquema de comunicación revela cómo debería funcionar la expansión de la funcionalidad. Puede expandir ambos lados para comunicar cuatro parámetros, entre los cuales el primero sería el tipo de instrucción (motor, sensor de lectura o cualquier otra cosa), y los tres restantes servirían como parámetros para ese mismo tipo de instrucción. Esto requiere un programa mucho más complejo en el lado de NXT, pero es factible.

O puedes probar con una solución intermedia, y para tu proyecto específico establecer una regla, por ejemplo, si el número de motor especificado es 10, en lugar de girar el inexistente décimo motor, el NXT lo interpretará realmente como leer el sensor de distancia y enviarlo de vuelta en el valor de respuesta—que por supuesto debería ser igualmente preparado y analizado en el lado del ordenador.

En cualquier caso, independientemente de si optaste por el EV3, el NXT o ambos después de todo, no intentaremos engañarte para que pienses que no tendrás que experimentar un poco. Pero será divertido, y podemos atestiguar por nuestra propia experiencia que realmente no es tan difícil una vez que comprendes los conceptos básicos. En ese momento, la sensación de poder controlar múltiples motores y sensores a través de un programa que se ejecuta en tu ordenador revelará todo un nuevo horizonte de posibilidades. ¡Estamos deseando ver tus proyectos robóticos avanzados!