

Iniciación a la robótica con LEGO® MINDSTORMS, 11ª entrega

Movimientos básicos de un robot

por Koldo Olaskoaga

Este es el tercer artículo de la serie de retos a resolver con LEGO® MINDSTORMS. La idea responde a las dudas que me han planteado equipos de la FLL relacionadas con la navegación de los robots. Vamos a ver cómo crear un robot que se mueva por donde deseemos con unas propuestas de mejora al uso básico del bloque Mover.

Reto

Esta propuesta es muy básica, la típica para iniciarse con la programación de robots. Es un reto en el que únicamente vamos a utilizar las salidas de motores del robot. Se trata de hacer que el robot se mueva en un itinerario predeterminado que podemos definir como queramos. En mi caso lo que voy a hacer es colocar 3 latas en el suelo y vamos a ver cómo solucionar algunos problemas comunes.

El robot

Vamos a comenzar montando el robot. Antes de comenzar a programar con sensores conviene practicar con robots que sean capaces de moverse de memoria, es decir, del mismo modo que nos moveríamos por un espacio conocido con la limitación de no poder tocar nada, totalmente a ciegas. Esto quiere decir que no le vamos a pedir mucho, únicamente lo siguiente:

- Capacidad de desplazarse en línea recta y girar.
- A partir de esto, la primera decisión que habremos de tomar es qué sistema de locomoción utilizar.

Sistemas de locomoción

La elección de uno u otro sistema de locomoción condiciona tanto la complejidad de la construcción de un robot como su programación, pero no solo eso, también su maniobrabilidad. Sin pretender ser exhaustivo, veamos algunos de los sistemas.

Dirección diferencial

Este sistema se basa en conectar un motor a la rueda o ruedas de cada uno de los lados del robot. Si los dos motores giran a la misma velocidad el robot avanzará en línea recta, si giran a diferentes velocidades el robot girará a uno u otro lado. Incluso será capaz de girar sobre sí mismo. Dependiendo de la superficie sobre la que ha de moverse el robot, puede haber diferencias en el modo en que se plasma este sistema de dirección, veamos tres de ellos.

- Robot con dos ruedas

Esta primera construcción es la más básica, sencilla de montar y programar, aunque no se mueve bien por superficies irregulares o poco deslizantes. El robot de la figura se apoya en las dos ruedas y otros dos puntos que deslizarán mejor o

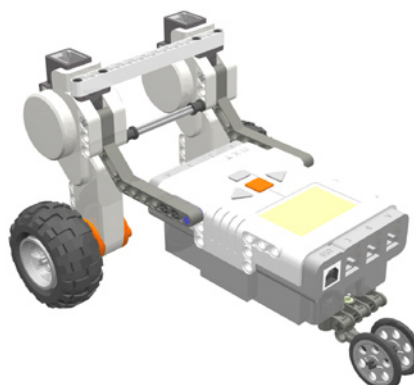
peor dependiendo de las características de la superficie sobre la que se mueva.



- Robot con dos ruedas y un tercer apoyo
- El apoyo delantero del sistema anterior se puede sustituir por un tercer apoyo como el de la imagen, aunque en este caso su montaje es más complejo.



- Robot con dos ruedas y rueda loca
- En este caso el robot también se apoya en tres puntos con una rueda que adapta su posición al giro del motor. En este caso el diseño del mecanismo de la rueda loca condiciona su eficiencia. Es posible también utilizar dos en lugar de una del mismo modo que lo hacen las sillas de ruedas y las sillas para niños.



- Robot con sistema de arrastre por orugas

Este sistema ofrece una base muy estable. Se trata de un robot sencillo de construir y relativamente estable. Su sistema de arrastre compatible con diferentes tipos de suelo, alfombra, terrazo, parquet... y permite al robot superar pequeños obstáculos. Es el que utiliza el modelo de inicio de LEGO® MINDSTORMS 2.0

Otros sistemas de locomoción

Otro sistema bien conocido es el que utilizan los coches que requerirá de dos motores, uno para el avance-retroceso y otro para el sistema de giro. Son robots estables pero de maniobrabilidad más reducida. Los robots con patas que merece la pena considerarlos aparte y corresponden a un nivel más avanzado.

El centro de gravedad

Pero hay un factor muy importante que siempre hay que tener en cuenta en cualquier diseño mecánico: la posición en que se encuentra el centro de gravedad. El centro de gravedad va a condicionar el comportamiento del robot, sobre todo en los arranques, paradas y giros. Un robot aparentemente estable puede balancearse o incluso volcar al arrancar si tiene el centro de gravedad muy retrasado. Puede volcar lateralmente al girar si lo tiene muy alto...

El centro de gravedad siempre se tiene que encontrar sobre el polígono que describen los apoyos del robot en el suelo (triángulo, cuadrilátero...), pero no solo eso, cuando el robot se pone en marcha, se detiene o gira, aparecen nuevas fuerzas que pueden hacer que el robot se balancee o vuelque, esta es la razón de que, como con los coches, convenga que el centro de gravedad esté lo más bajo posible.

Mi elección

He montado el robot que se puede ver en la primera figura, sencillo y rápido de montar. No hace falta nada más para experimentar con los motores.

A programar

Vamos a empezar con un programa muy básico para ir mejorándolo.

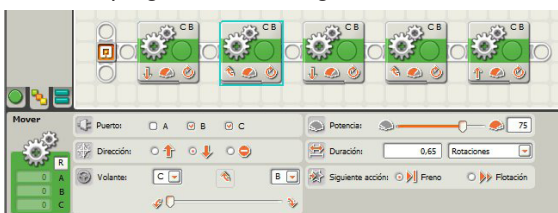
Reto básico nº 1

Trazamos una línea recta y colocamos una lata a la distancia de ella que se quiera. El robot arranca junto a la línea para rodear la lata y volver.

El algoritmo para este caso es muy sencillo:

1. avanza recto
2. gira rodeando la lata
3. avanza recto

Cuando lo convertimos en un programa en NXT-G, tras avanzar recto hay que girar y aunque podría hacerse de alguna otra manera, vamos a hacerlo de la siguiente: giro en ángulo recto a la derecha, avance para librar la lata y nuevo giro a la derecha en ángulo recto. De este modo puede hacerse un giro más cerrado que haciéndolo en un solo movimiento. El programa será el siguiente:



Para fijar cuánto han de girar los motores, NXT-G permite hacerlo por tiempo, grados de giro del motor (con precisión de 1 grado) y rotaciones. Además de estas tres opciones tiene una cuarta, ilimitado, muy útil para trabajar con sensores. En este programa se ha utilizado la opción de rotaciones en el panel de control del bloque Mover. Es una opción que generalmente da más precisión, ya que con la opción tiempo el rendimiento puede bajar si el nivel de las baterías desciende. Se ha determinado la magnitud del giro necesario por tanteo.

Mejora nº 1

Si bien desarrolla la tarea asignada, el movimiento no es fluido, sino que cuando la ejecución pasa de un bloque al siguiente los motores se paran y arrancan otra vez. Por defecto el bloque Mover tiene seleccionada la casilla Freno (Brake) en su panel de control, lo que hace que finalice su ejecución bloqueando los motores y poniéndolos en marcha otra vez con el siguiente bloque.

Se puede mejorar algo su comportamiento seleccionando la casilla Flotación (Coast) en lugar de la de Freno. De este modo al final de la ejecución del bloque, en lugar de bloquear los motores lo que hace es dejar de alimentarlos, de tal modo que incluso podría seguir moviéndose gracias a su inercia.



Reto

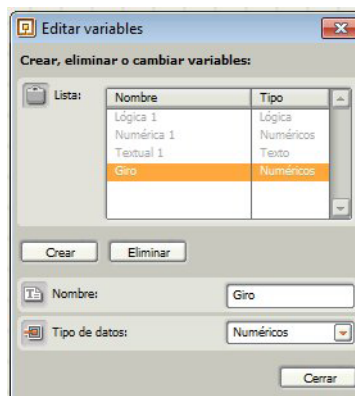
Otro reto interesante es colocar varias latas en el suelo y hacer un slalom. Queda como ejercicio y se trata de enlazar los bloques Mover necesarios (tramos rectos y giros) de tal modo que resuelva el reto.

Mejora nº 1: Mejorar la precisión en los giros

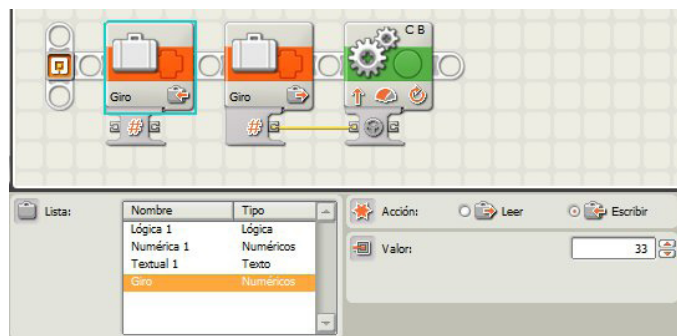
El control deslizante del panel del bloque Mover ofrece 10 posiciones diferentes para giros a la izquierda y 10 para giros a la derecha. Esto puede ser suficiente en muchas ocasiones, pero puede darse el caso de que en una posición se pase y en la anterior no llegue.

El giro también puede establecerse con valores numéricos entre 0 y 100 en un sentido y entre 0 y -100 en el otro. Hay que recordar que con el valor 100 supone que el robot girará sobre sí mismo en un sentido y con -100 en el otro.

Para dar la entrada del valor deberemos utilizar una variable en la que almacenaremos la magnitud del giro deseado. Una vez que tengamos abierto un nuevo programa iremos a Edición > Definir nueva variable. Crearemos una nueva llamada Giro tipo Numérica (ha de almacenar un valor numérico).



En la siguiente imagen puede verse un pequeño fragmento de código que muestra cómo establecer como almacenar un valor en la variable previamente definida y utilizarla para establecer la magnitud de giro. Se puede cambiar el valor de una variable siempre que se desee.



El primer bloque guarda en la variable Giro el valor deseado. El segundo lee el contenido de la variable y vía el cable amarillo, asigna dicho valor a la toma de datos Volante.

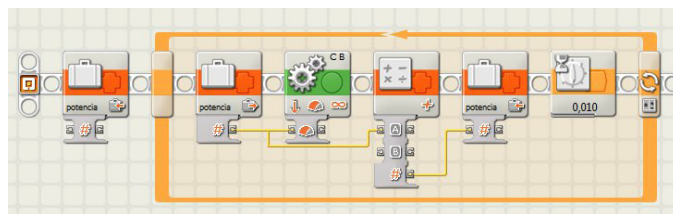
Mejora nº 2: Arranque y parada en rampa

Si bien conviene que el centro de gravedad esté bajo, en ocasiones la necesidad a cubrir puede impedir que esté tan bajo como se desea. No es el caso del robot que estamos utilizando, pero en dichos casos, los arranques y paradas pueden hacer que vuelque (del mismo modo que puede suceder con una motocicleta o con una bicicleta al frenar en seco).

Un modo de evitar esta situación es utilizando un arranque en rampa, es decir, en lugar de meter toda la potencia a la vez, hacerlo progresivamente hasta llegar al valor deseado. El bloque motor tiene esta opción incluida, pero dado que el bloque Mover no la tiene tendremos que hacerlo de otro modo. Veamos el algoritmo para que el robot arranque acelerando progresivamente, avance durante dos segundos y frene de la misma manera.

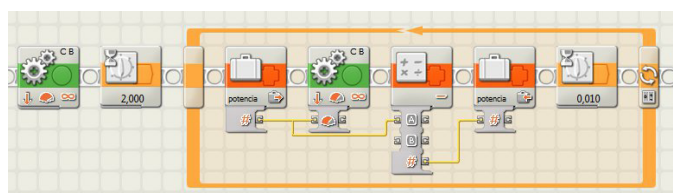
1. Definir una variable que guarde el valor de la potencia a asignar al bloque Mover.
2. Dar el valor inicial 0
3. Repetir lo siguiente 100 veces (para llegar a potencia 100)
 - a. Asignar el valor almacenado a la toma de datos Volante del bloque Mover
 - b. Sumar una unidad a la variable
 - c. Esperar 0,01 s (así dado que se va a repetir 100 veces pasará de 0 a 100 en 1s).
4. Avanzar durante 2 segundos a potencia 100
5. Repetir lo siguiente 100 veces (para bajar a 0)
 - a. Asignar el valor almacenado a la toma de datos Volante del bloque Mover (el último era 100)
 - b. Restar una unidad a la variable
 - c. Esperar 0,01 s (así dado que se va a repetir 100 veces pasará de 100 a 0 en 1s).

Una vez que tengamos abierto un nuevo programa hemos de crear la nueva variable. Le he puesto como nombre potencia y será del tipo Numérica (ha de almacenar un valor numérico). En la siguiente imagen pueden verse los pasos 2 y 3 del algoritmo.



Se asigna el valor 0 por medio del Bloque Variable y se abre un bucle. En el panel de control del bucle se selecciona la opción Control > Contar hasta 100 veces. En su interior la secuencia de bloques comienza asignando el valor contenido en la variable al bloque Mover, le suma una unidad y espera una centésima antes de volver a empezar.

El resto del programa:



Para avanzar utilizaremos el bloque Mover con potencia 100 y la opción duración en ilimitado, y a continuación un bloque Esperar con la opción Tiempo 2s. Si en lugar de hacerlo así utilizásemos solo el bloque Mover con la opción duración 2s, no podríamos controlar la frenada de este modo. El resto es similar al bucle inicial, solo que en este caso se resta una unidad en lugar de sumar.

Propuesta 1

Ahora se trata de practicar, intenta hacer esto último sin utilizar una variable. Para ello fíjate lo que sucede cuando en el panel de control de la estructura Bucle (Loop) hacemos clic sobre la casilla de la parte inferior Contador.

Propuesta 2

Si bien la potencia de los motores puede fijarse entre 0 y 100, el robot no comenzará a moverse con potencia 1 sino que necesitará un mínimo para ponerse en marcha, algo que dependerá de su peso y diseño mecánico. Si el robot utiliza diferentes accesorios para diferentes tareas, tal y como se hace en la FLL, esa potencia mínima puede variar. Desarrolla un programa que permita conocerla, se puede utilizar el display para mostrarla.
#



Lrobotikas.net

Robótica Educativa y Recreativa