



ANEXO 1. El entorno
de programación “Enchanting”

ANEXO 1.

El entorno de programación “Enchanting”

Enchanting¹ es una pasarela que permite programar el robot LEGO Mindstorms NXT desde la versión BYOB del lenguaje SCRATCH. Añade algunas instrucciones más específicas para el control de un robot “real”, por lo que mejora a BYOB como lenguaje “robótico”. Enchanting crea una traducción a Java de los programas escritos en BYOB, que puede ser ejecutada por el microprocesador NXT de LEGO si previamente hemos sustituido su firmware² NXT original de LEGO por el firmware leJOS (JAVA para LEGO Minstorms).

1. EL ENTORNO ENCHANTING

Es semejante al entorno BYOB, salvo en unos pequeños, pero esenciales, detalles:

- La ejecución del programa la realiza siempre el robot LEGO y no se ejecuta en el “escenario” de la ventana del entorno Enchanting, donde sólo aparece el disfraz asignado al robot.

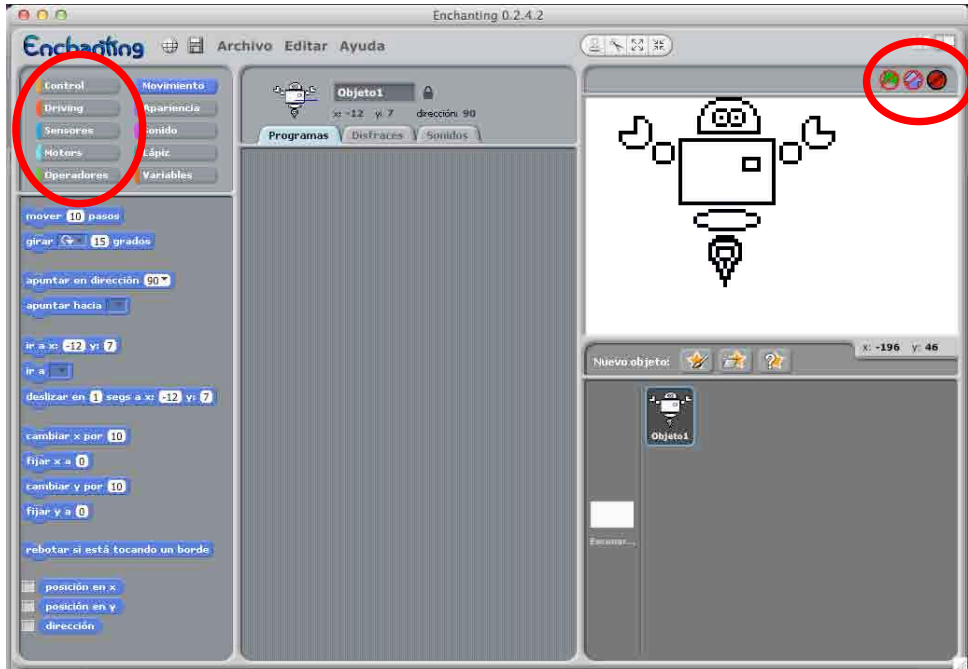
¹ **Enchanting** (<http://enchanting.robotclub.ab.ca/tiki-index.php>)

Está desarrollado por *Clinton Blackmore*, cabeza visible de la asociación *Southern Alberta Robotics Enthusiasts (SABRE)* de Canadá, junto con un grupo de colaboradores.

Véase la referencia a Enchanting en la web del Observatorio Tecnológico del Instituto Nacional de Tecnologías Educativas:

<http://recursostic.educacion.es/observatorio/web/ca/software/software-educativo/1018-monograficodesarrollos-de-scratch-para-robotica-enchanting-y-s4a>

² El firmware es el programa de más bajo nivel que controla la electrónica del NXT. Suele estar grabado en una memoria tipo ROM, flash. Está en la base del sistema operativo y del lenguaje con el que se programa el NXT.



- Dicha ejecución se controla con los “iconos de control”



que sólo se muestran activos (sin los símbolos de “prohibido”) cuando el robot está conectado vía USB.

- Aparecen dos nuevas categorías de bloques “robóticos”: “Motors” y “Driving”, que no existen en BYOB/SCRATCH, y que van dirigidas a la programación del movimiento del NXT como un robot “real”:
 - “Motors”, que establece las conexiones de los motores a los puertos del NXT y contiene los bloques para controlar los motores de modo individual.
 - “Driving”, que adquiere información de la disposición física de las ruedas motrices del robot (su tamaño y su separación) y contiene los bloques para controlar globalmente el movimiento del motor (el movimiento conjunto de ambos motores) y programar directamente el desplazamiento lineal del robot (en cm u otras unidades) y su movimiento angular (en grados).

2. LA DESCARGA DE ENCHANTING Y LA INSTALACIÓN DEL FIRMWARE LEJOS NXJ



La descarga e instalación de Enchanting en nuestro ordenador (para Windows, Mac OS y Linux) viene indicada en la página "download" de su web: http://enchanting.robotclub.ab.ca/tiki-index.php#Download_Enchanting




Recordemos que previamente a la instalación de Enchanting se debe descargar e instalar el programa controlador "LEGOs Fantom Driver for the NXT", que permite que el sistema operativo de nuestro ordenador reconozca y se comuniqué, vía USB, con el dispositivo externo NXT.

Una vez instalado y abierto Enchanting, lo primero que debemos hacer es conectar, vía USB, el robot y sustituir su firmware original por el firmware lejos NXJ, que permite a nuestro robot ejecutar programas Java. Para ello, cliquearemos en la opción "Enviar Firmware ..." del menú "Archivo" de la ventana.

3. EL CONTROL DE LA EJECUCIÓN DE PROGRAMAS

- La conexión del entorno Enchanting con el robot es delicada y el proceso de traducción y de paso de los programas al robot es algo lento. Se deben conocer algunas reglas:

- Los programas deben tener el bloque  como bloque inicial y conviene que lleven el bloque  como bloque final.
- Si el microprocesador NXT se bloquea, se debe desconectar la alimentación de las pilas/batería y luego reconectarla.
- En los problemas más serios de falta de conexión (mensaje de "no device connected") se requiere el reinicio del propio ordenador.

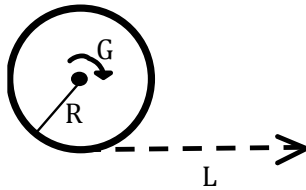
- El icono de control  es un "upload and play", de modo que descarga ("sube") el programa al NXT e inicia su ejecución. Sirve para "probar" los programas ya que la conexión con el cable USB debe mantenerse durante la ejecución del programa.
- El icono de control  es un "upload", de modo que simplemente descarga el programa en el NXT. Ello permite desconectar luego el cable USB y que el robot ejecute autónomamente el programa. El programa queda almacenado en la carpeta "files"; se debe pulsar doblemente en el botón naranja del NXT para alcanzar la ejecución del programa y luego pulsar una vez más para iniciar dicha ejecución. La carpeta almacena un solo programa Enchanting, sin nombre específico. Cada programa subido sustituye al programa anterior.
- El icono de control  detiene todo (si hay conexión USB).

4. “DRIVING” (MOVIMIENTOS EN EL PLANO) CON ENCHANTING

La categoría de instrucciones “Driving” permite programar el robot “real” LEGO NXT del mismo modo que programamos el robot “virtual” de SCRATCH/BYOB. Enchanting simplifica mucho la programación del movimiento físico del nuestro robot permitiéndonos programarlo mediante instrucciones que comandan la distancia lineal que recorre y los grados que gira. Así, es posible en Enchanting:

- Escribir como dato del bloque “desplazar...” (“drive...”) la *distancia lineal* del desplazamiento en cm (o en centímetros, pulgadas...).
- Escribir como dato del bloque “girar...” (“turn...”) el *ángulo* de giro del robot en grados.

Para ejecutar estas instrucciones, Enchanting, a partir de los datos anteriores (distancia lineal y ángulo de giro), debe calcular los correspondientes valores de rotación de los motores del robot.



Como se explica más ampliamente en el capítulo 7 (sobre procedimientos de Avanzar y Girar en EV3), el ángulo G que debe girar el motor de un robot (si está directamente conectado al eje de la rueda) para que este avance una distancia L es:

$$G = \frac{360}{2\pi R} L = \frac{360}{\pi D} L \quad (\text{siendo } D \text{ el diámetro de la rueda})$$

y vemos que G no sólo depende de la distancia L que avanza el robot, sino también del *diámetro* D de sus ruedas.

Adicionalmente, el ángulo G que debe girar el motor de cada rueda (en sentido inverso) cuando el robot pivota un ángulo $ALFA$ en torno al punto medio del eje E entre sus ruedas motrices, es:

$$G = \frac{E/2}{R} ALFA = \frac{E}{D} ALFA \quad (\text{siendo } D \text{ el diámetro de la rueda})$$

y vemos que G no sólo depende del ángulo $ALFA$ que gira el robot, sino también de la *distancia* E entre las ruedas ³ y del diámetro D de las mismas.

³ Esta distancia E se denomina técnicamente “vía” o “rodada del eje” y es la distancia entre el centro de las dos ruedas de un mismo eje en un vehículo (en inglés: “track width”, “axle track” o, simplemente, “track”).

4.1. Configuración inicial en “Driving”

Por todo ello, la primera vez que se accede a la categoría “Driving”, se debe activar el mensaje **Configure Drive Type** que nos permite acceder a la ventana del Editor de Configuración.



En ella debemos arrastrar el único bloque existente (Differential drive) para llevarlo debajo del encabezamiento “Drive Type” y construir el programa informativo que Enchanting necesita:

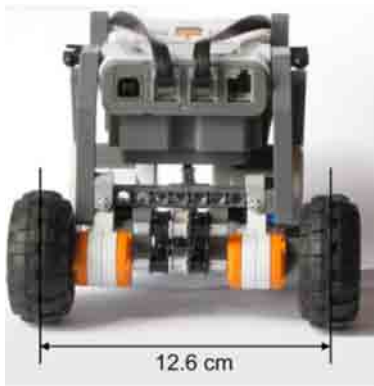
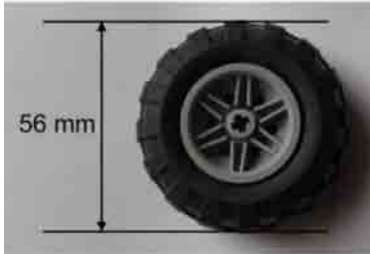
En dicho programa debemos introducir los datos específicos de nuestro robot:

- *Diámetro D* de nuestras ruedas motrices (wheel diameter).
- *Distancia E* entre ambas ruedas motrices (track width)⁴.
- Debemos igualmente informar de los puertos del NXT en los que conectamos nuestros motores, así como del sentido que a priori asignamos al movimiento de los mismos⁵.

⁴ Conviene que expresemos los valores del diámetro D y la distancia E en unidades de cm, con decimales; en consecuencia, los desplazamientos L del robot los pediremos en centímetros, que parece la unidad más conveniente.

⁵ Recordemos que el sentido de giro (a derecha o a izquierda) es arbitrario, y depende de la orientación de los motores en la construcción del robot. Si el robot gira en el sentido contrario al de las instrucciones de girar, simplemente se debe intercambiar la asignación de los puertos a los motores.

Recordemos que el sentido de avance y retroceso es, igualmente, arbitrario. Si el robot camina en el sentido contrario al de las instrucciones de desplazar, simplemente debemos seleccionar “in reverse” en vez de “forwards” en el parámetro “motors drive”.



a) La medida del diámetro D de las ruedas motrices

El diámetro de la rueda es su diámetro de rodadura, esto es, la distancia radial entre sus bordes, y podemos medirlo con una simple regla.

Recordemos que las cubiertas LEGO llevan inscritas su marca y sus dimensiones (diámetro x ancho) en el lateral. Ello nos permite conocer su diámetro sin necesidad de medirlo.

Por ejemplo, la rueda de la imagen, que es la más común en los robots de LEGO, tiene un diámetro D de 56 mm.

b) La medida de la distancia E entre las ruedas

La distancia E es la distancia entre los centros de las bandas de rodadura de las cubiertas. Por comodidad puede medirse igualmente la distancia entre los laterales izquierdos (o derechos) de ambas ruedas.

Por ejemplo, la distancia E entre ruedas en el robot de la imagen es de 12,6 cm.

Una vez rellenados los datos del bloque "Differential drive" y aceptada la ventana del Editor de Configuración, la categoría "Driving" nos muestra, y los bloques que la constituyen, que ya están listos para su uso con nuestro robot específico.

En esta categoría se nos informa, adicionalmente, de que las instrucciones que construyamos con estos bloques "no son 100% precisas", es decir, que el comportamiento real (en traslaciones y giros) del robot puede no corresponder con precisión a

los datos de traslación y/o giro que hayamos escrito en las instrucciones.

Con un poco de paciencia, podemos mejorar sensiblemente la precisión en el comportamiento del robot realizando un *calibrado*, con ligeras modificaciones, que llevaremos a cabo por ensayo y error, en los valores de *diámetro D* y *distancia E* introducidos inicialmente en el Editor de Configuración. Recordemos que dichos valores

los podemos escribir y modificar con precisión de milímetros (y aún menor), pero que los valores empíricos iniciales de las medidas tomadas sobre nuestro robot con una sencilla regla no suelen alcanzar dicha precisión...). Hay problemas adicionales de inercia, de frenado de las ruedas, de derrape, etc., que también exigen el calibrado de nuestro robot.

4.2. "Driving" en coordenadas intrínsecas

Recordemos que las instrucciones en coordenadas intrínsecas describen movimientos de desplazamiento (lineal y angular) que se ejecutan a partir del estado en que se encuentra el robot al inicio de cada instrucción. Son instrucciones "incrementales" y describen movimientos relativos (formas geométricas) y no posiciones absolutas del robot con respecto a un origen de coordenadas.

a) Instrucciones iniciales

Enchanting nos permite establecer la velocidad lineal, la velocidad angular y la aceleración lineal del robot. Son instrucciones muy potentes si deseamos enseñar los principios físicos de la cinemática (los términos de posición, velocidad aceleración...). En las tareas más simples, como las que nos ocupamos de la iniciación a la robótica, estas instrucciones sirven como instrucciones iniciales para establecer la velocidad con la que el robot se desplaza y gira. Unos valores adecuados (medios) son los de la figura.



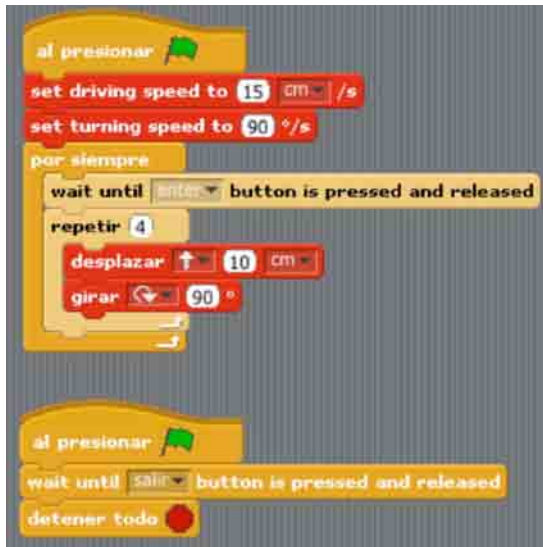
b) Instrucciones "con datos de avance" (lineal y angular)

Las instrucciones clásicas, semejantes a las de SCRATCH/BYOB, son las dos primeras de la figura, que permiten desplazar al robot una determinada longitud L y girarlo unos determinados grados ALFA (pivotando en torno al punto medio del eje entre sus ruedas)

La instrucción "arc..." permite ejecutar giros más realistas, de ángulo ALFA, pero con un determinado radio de giro de la curva.

Un ejemplo es el siguiente. Ya sabemos que el programa de la figura recorre (y dibuja, si el robot tiene un lápiz) un cuadrado de 10 cm de lado. Para evidenciar el carácter de instrucciones intrínsecas de "desplazar" y "girar", podemos insertar dicho programa en un bu-

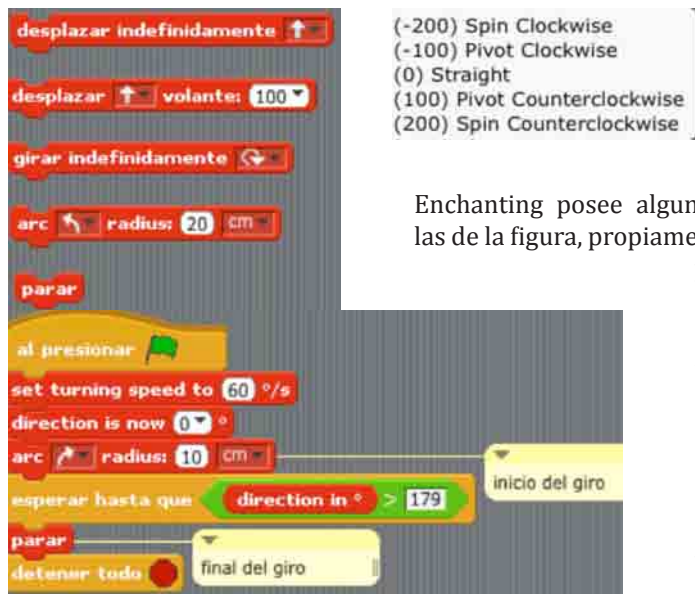




de indefinido que inicia el cuadrado cada vez que pulsamos la tecla “enter” (tecla naranja) del NXT. Podemos así llevar manualmente el robot a distintas posiciones y distintas orientaciones en un plano, pulsar “enter” y comprobar que siempre ejecuta cuadrados (de 10 cm) a derechas a partir de dichas posiciones iniciales.

Hemos establecido un segundo programa, que se ejecuta simultáneamente al primero, para poder detener la ejecución del bloque “por siempre” y finalizar la tarea, cuando pulsemos la tecla “stop” (tecla gris) del NXT.

c) Instrucciones de “start” y “stop”



Enchanting posee algunas instrucciones, como las de la figura, propiamente “robóticas”, de inicio y fin de eventos.

Con ellas se lanza un movimiento permanente del robot mientras que el intérprete continúa la ejecución de las siguientes instrucciones del programa a la vez que se mantiene el movimiento iniciado, que sólo finaliza cuando el

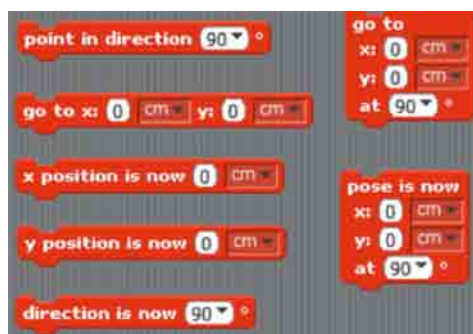
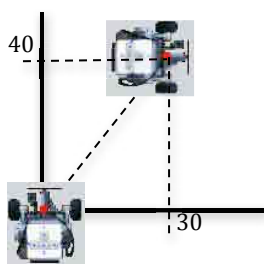
intérprete encuentra la instrucción “parar” (que usualmente se condiciona a la medida de algún sensor).

Un ejemplo es el siguiente. Usando el “sensor interno” **direction in °** que indica la posición angular del robot, e iniciando dicha posición a cero en el estado inicial del robot con **direction is now 0 °**, podemos realizar (véase figura en página anterior) unsemicírculo de 10 cm de radio.

4.3. “Driving” en coordenadas cartesianas

Usando las instrucciones de la figura, si colocamos un robot en un punto cualquiera del plano y establecemos (arbitrariamente) dicho punto como el origen de coordenadas, y establecemos adicionalmente su orientación como el origen de su orientación (dirección) angular, podemos mover dicho robot en coordenadas cartesianas, ubicándolo en cualquier posición del plano y con cualquier orientación de su “cabeza” (cap).

Un ejemplo: el programa de la figura sitúa el estado inicial del robot en el origen de coordenadas y orientado hacia “arriba”. El programa lo traslada luego hasta la posición (x=30, y=40) y orientado hacia “la derecha”.



5. LA CATEGORÍA “MOTORS”

La primera vez que se accede a la categoría “Motors” se debe, igualmente, configurar la asignación de motores a los puertos del NXT. Para ello trasladaremos dos veces el bloque “Controlador motor con codificador Motor NXT” debajo de dos de los encabe-



zamientos “Motor Port...” (habitualmente debajo del “Port B” y el “Port C”) y asignaremos nombres distintos a cada motor (véase figura).

A partir de ese momento, la categoría “Motors” nos muestra ya los bloques que la constituyen que ya están listos para su uso con nuestro robot específico. Estos bloques van dirigidos a hacer girar directamente los dos (tres) motores del NXT de manera muy precisa y con varias posibilidades. Exploraremos estas posibilidades con unos ejemplos:

a) Uso de la instrucción “rótate una distancia angular determinada”

“rótate” es la instrucción básica para girar un motor un determinado ángulo. El programa de la figura hace avanzar linealmente el robot girando ambos motores una distancia angular de 360° (una vuelta) de modo simultáneo. Hay que escribir la primera instrucción “rótate” (motor izquierdo) como “just start it”, es decir, como una instrucción de tipo “start”, para conseguir la simultaneidad en el giro de ambos motores.



b) Uso de instrucciones “start” y “stop”

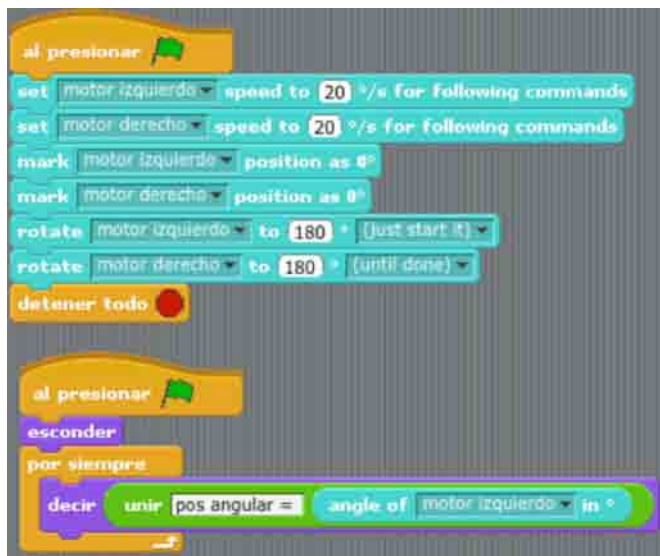
El programa de la figura inicia el giro de ambos motores y el robot camina durante 2 segundos.



El programa de la figura hace pivotar el robot sobre su rueda izquierda, girando durante 2 segundos. Se usan aquí instrucciones “start” que permiten especificar la velocidad de giro de cada rueda.



c) Uso de la instrucción “róotate hasta una posición angular determinada”



Como en “Driving”, en “Motors” podemos hacer uso de los sensores internos que nos permiten conocer las posiciones angulares de ambos motores. Podemos establecer determinados valores iniciales (usualmente 0°) y luego podemos mover los motores hasta alcanzar determinados valores finales.

El programa de la figura mueve simultáneamente ambos motores muy lentamente desde la posición angular 0° hasta la posición final 180° (media vuelta). El programa complementario escribe en el display del NXT las sucesivas posiciones angulares mientras dura el movimiento.

6. LA CATEGORÍA “MOVIMIENTO”

Curiosamente, en Enchanting, y al contrario que en BYOB, las instrucciones de la categoría “Movimiento” no sirven para mover el robot; para ello están las ya descritas “Driving” y “Motors”.

¿Para qué sirven, pues, los bloques de “Movimiento”? Sirven para mover no el robot, sino el icono que representa al robot (o cualquiera de sus disfraces) en un escenario que es el display del NXT.

El que pueda haber movimiento de objetos (virtuales) en la pantalla del NXT tiene una utilidad limitada en robótica, donde lo esencial es el movimiento del propio robot. No obstante, esta opción adicional de Enchanting permite:

- Definir uno o varios objetos, con sus disfraces. Son “sprites” sencillos, monocromos, de un tamaño máximo de 16 x 25 píxeles.
- Moverlos, como se mueven los objetos en BYOB, en la pantalla del NXT, de dimensiones 96 x 64 píxeles (aproximadamente).



Como ejemplo tenemos el programa de la figura que muestra en el display una “cara” de robot, con ojos que miran moviéndose a izquierda y a derecha...

Para el escenario del programa, hemos usado el propio editor de fondos y de objetos (semejante al de BYOB) y con él:

- Hemos creado un disfraz del escenario, el “fondo2”, con un dibujo simple de la cara del robot, con las cuencas de los ojos.
- Y hemos creado el “disfraz2” para el “sprite” de Enchanting, constituido por dos pequeños círculos representando las pupilas de dos ojos.



7. LA CATEGORÍA “APARIENCIA”



Los cambios de disfraz y los mensajes también los muestra Enchanting en la pantalla del NXT.

- La instrucción “decir...” muestra el mensaje vinculado al objeto (sprite), como ocurre en BYOB. Su posición en pantalla depende, pues, de la posición en la que situemos el objeto. Si sólo se desea ver el mensaje, se puede “esconder” el objeto. (En la imagen, hemos disfrazado el objeto de “punto grueso”.)
- La instrucción “imprimir...”, por el contrario, muestra el mensaje empezando por el extremo superior izquierdo de la pantalla. Esta instrucción permite informar de valores variables, tanto del movimiento del robot como de los sensores.



En el ejemplo se muestra el valor de la distancia medida por el sensor de ultrasonidos. Las medidas se muestran cada medio segundo (para poder leerlas). Este programa de lectura finaliza al pulsar la tecla gris “salida” del NXT.

8. LA CATEGORÍA “LÁPIZ”




Las instrucciones de la categoría “Lápiz”   van también dirigidas a la pantalla del NXT. Muestran, o no, las trazas de los movimientos de los objetos en dicha pantalla.


Dado que la pantalla del NXT es monocroma, la instrucción  sólo acepta el color negro.

9. SENSORES EN ENCHANTING

9.1. Los sensores “virtuales” en Enchanting

Cuando iniciamos un entorno Enchanting y abrimos la categoría “Sensores” vemos bloques que controlan eventos internos en el robot y los podemos considerar como sensores “virtuales”, ya que no corresponden a sensores “físicos”, que haya que conectar a un puerto. Los más interesantes para un uso robótico son:

- Las teclas del NXT: “enter” (naranja), “salir” (gris oscuro), “flecha izquierda” y “flecha derecha” (gris claro), se controlan con el operador lógico  , que devuelve “cierto” si la tecla seleccionada ha sido pulsada⁶.
- El cronómetro: se controla con el operador  , que devuelve el tiempo (en segundos con centésimas) transcurrido desde que se inició a cero el contador; con la instrucción  .
- Debemos también incluir los sensores internos que devuelven las posiciones angulares de cada motor y que aparecen dentro de la categoría “Motors”. Ya los hemos comentado al tratar dicha categoría (apartado 4 de este anexo).

⁶ Un uso práctico es que permite hacer un “pause” en cualquier parte de un programa, con la instrucción  y reanudarlo pulsando, por ejemplo, la tecla “enter”.

9.2. Los sensores “reales” en Enchanting



En el caso de los sensores físicos que deseemos conectar al NXT, y como ocurre en las categorías de bloques “Driving” y “Motors”, debemos clicar en

Configurar Sensores

para entrar en el Editor de Configuración de Sensores y configurar los puertos y dar nuestros propios nombres a los sensores que vamos a utilizar. Una vez realizado esto, los bloques correspondientes aparecen ya en la lista de dicha categoría y podremos usarlos. Aunque los puertos de conexión para cada sensor son optativos, es conveniente mantener un mismo criterio para la conexión de los sensores que utilicemos, que suele ser:

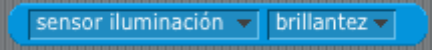
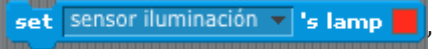


- Puerto 1: Sensor de contacto
- Puerto 2: Sensor de sonido
- Puerto 3: Sensor de luz
- Puerto 4: Sensor de distancia

Describiremos el uso de los sensores de distancia y de iluminación, que son los dos sensores más usados en nuestros proyectos de robótica.

Por ejemplo, si configuramos el sensor de ultrasonidos (como “sensor distancia”) obtenemos el bloque:

- **distancia del** sensor distancia **en cm**, que devuelve la distancia en cm que mide el sensor.

Si configuramos el sensor de luz para el NXT (como “sensor iluminación”), obtenemos varios bloques, entre ellos:

-  , que devuelve la iluminación que percibe el sensor, en una escala de 0 a 100.
-  , que permite programar si el sensor de luz llevará su pequeña lámpara roja encendida o apagada.
- Cuando la lámpara está encendida y el sensor muy cercano a una superficie (el suelo, por ejemplo), las medidas de iluminación se deben esencialmente a la cantidad de luz de la lámpara reflejada por la superficie, y la luz del entorno prácticamente no influye.
-  y  nos permiten realizar un proceso de calibración, en el que el NXT recoge una serie de valores de iluminación en los distintos puntos en los que situamos al robot, y finalmente reescala dichos valores, asignando el valor “0” al valor más bajo de la serie y “100” al más alto.



En la figura se muestra un programa que incluye un proceso de calibración. Durante el bucle de repetición colocamos el robot con el sensor sobre los diferentes puntos que luego va a recorrer en su trayectoria. Cuando pulsamos la tecla “enter” finaliza el proceso y si hemos previamente creado la variable “luz” (y estamos conectados por USB) podemos ver en el escenario de Enchanting cómo, si situamos el robot sobre los puntos de mayor iluminación del recorrido, el valor “luz” es próximo a 100, y si lo situamos sobre los de menor iluminación, el valor es próximo a cero.



Este proceso de calibración es particularmente interesante cuando programamos el robot como un “seguidor” de línea. Para constatar la sencillez y elegancia con la que se puede programar en Enchanting un robot seguidor, mostramos el programa de la figura⁷, en el que el seguimiento del borde derecho de una línea gruesa negra se consigue mediante un juego de velocidades complementarias para ambas ruedas.

Cuanto más se acercan al “blanco” (un valor próximo a 100) las medidas del sensor, más deprisa se mueve la rueda derecha y más despacio se mueve la izquierda, para que el robot se aproxime al borde. Cuanto más se acerque el sensor al “negro” (un valor próximo a “0”), ocurre el caso contrario. El factor de velocidad (3, en el ejemplo) simplemente aumenta la velocidad del seguimiento.

Antes de iniciar el recorrido por el borde de la línea, como en el caso anterior, se debe realizar el calibrado del robot, situando su sensor de luz sobre una zona totalmente blanca (fuera de la línea) y sobre una zona totalmente negra (dentro de la línea). Inmediatamente después, sin que el sensor haga otras medidas, se debe pulsar “enter” y colocar “a mano” el robot al inicio de la línea a seguir.

⁷ Basado en “The Line Follower” de MonashBlochBooks : <http://monash-blockbooks.appspot.com>

