



**Profesor**  
*Horacio A. Coral Enríquez*

# Guía de Laboratorio No. 1

## Introducción a la Plataforma LEGO Mindstorms

### 1. DESCRIPCIÓN

En esta primera práctica se pretende familiarizar al alumno con las herramientas básicas necesarias para la implementación de algoritmos de control en la plataforma de Lego Mindstorms NXT.

El lenguaje que se utilizará para la programación es el denominado NXC (Not eXactly C) para los Lego NXT, el entorno de programación será el proporcionado por el software Bricx Command Center.

### 2. OBJETIVOS

- Identificar los componentes principales de la plataforma lego mindstorms como lo son los sensores, actuadores y unidades de procesamiento (Brick).
- Examinar las características técnicas de los sensores, actuadores y los componentes internos del Brick.
- Estudiar las diferentes herramientas de programación presentes en el software Bricx Command Center.
- Reconocer las estructuras básicas de programación del lenguaje NXC.

### 3. GENERALIDADES DE LA PLATAFORMA LEGO NXT



Figura 1: Brick NXT

Lego Mindstorms NXT es un kit de robótica programable que incluye, además de componentes de construcción básicos de Lego, un Brick inteligente y una serie de sensores que permiten al robot realizar diferentes movimientos y operaciones. El Brick se comunica con los sensores por medio de cables de conexión especiales RJ12 y por medio de USB o bluetooth con el computador.

### 3.1. Descripción de hardware

Externamente, el Brick está compuesto por tres puertos de salida ubicados en la parte superior izquierda y denominados A, B y C como se muestra en la Figura 1. En la parte superior derecha se encuentra el conector USB con el cual se realiza la comunicación con el computador. Los cuatro puertos que se encuentran abajo son los puertos de entrada. Además, el Brick cuenta con un parlante incorporado que permite la reproducción de archivos de sonido. El botón naranja enciende el Brick, las flechas sirven para desplazarse y el botón gris oscuro para retroceder o apagarlo.

Por otro lado, la Figura 2 detalla los componentes hardware del Brick y cómo estos se interconectan entre sí. A continuación se realiza una descripción general de los componentes:

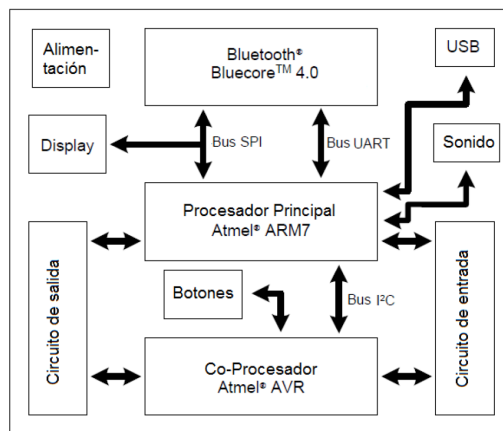


Figura 2: Estructura interna del Brick

**Procesador principal:** El procesador principal tiene las siguientes características técnicas:

- Procesador ATMEL ARM7 de 32 bits (AT91SAM7S256)
- 256 Kbytes de FLASH
- 64 Kbytes de RAM
- 48 MHz

Este procesador se encarga principalmente del manejo de la alimentación, de la creación de señales PWM (Pulse Width Modulation) para los tres motores y de realizar la conversión Analógica/Digital de los puertos de entrada. De igual forma, como se observa, también realiza la comunicación con el dispositivo Bluetooth, con el sonido y el display.

**Co-Procesador:** Por su lado, el co-procesador se comunica con el procesador principal mediante  $I^2C$  a 380 kbytes/s. La comunicación se realiza mediante la actualización de 2 registros de memoria cada 2ms. Las especificaciones del co-procesador son:



- Procesador ATMEL AVR de 8 bits (ATmega48)
- 4 Kbytes de FLASH
- 512 bytes de RAM
- 8 MHz

**Comunicación Bluetooth:** La comunicación inalámbrica se realiza mediante el dispositivo CSR BlueCore TM v2.0 +EDR System cuyas especificaciones son las siguientes:

- Serial Port Profile (SPP)
- 47 Kbytes de RAM interna
- 8 Mbit de FLASH externa
- 26 MHz

El Brick NXT puede comunicarse simultáneamente (uno a la vez) con otros tres dispositivos mediante Bluetooth. Esto significa que si el maestro está enviando información a uno de los esclavos, no podrá recibir datos de otro aunque la comunicación esté establecida. La comunicación puede realizarse con dispositivos que hayan sido programados para usar el protocolo de Mindstorms y a su vez puedan usar el SPP, es decir que un Brick puede enviar y recibir información de otros durante la ejecución de un programa. El dispositivo Bluetooth se comunica con el procesador principal mediante una interfaz UART cuya velocidad es de 460.8 kbit/s.

**Puertos de salida:** Los Puertos de salida se usan para controlar actuadores que se encuentren conectados al NXT. La implementación de los puertos está construida de forma que el dispositivo conectado pueda devolver información al Brick sin tener que usar un puerto de entrada. Las señales de PWM de salida están controladas por un driver de motor interno que puede proveer hasta 700mA y 1A pico. Las señales de entrada pueden medir los pulsos que realiza el motor y por lo tanto saber su dirección y velocidad(pulsos por segundo).

**Puertos de entrada:** Dependiendo del sensor conectado a uno de los cuatro puertos de entrada, es posible medir diferentes parámetros del mundo físico. Gracias a la construcción de los puertos, es posible usar sensores que hagan mediciones tanto analógicas como digitales. El procesador AVR se encarga de convertir los datos análogos a digitales mediante un conversor de 10 bits y con una frecuencia de muestreo de 333Hz. Por su lado, los datos digitales son enviados al procesador ARM7 mediante  $I^2C$  a 9600bit/s.

**Display:** Se trata de un display en blanco y negro con una resolución de 100x64 pixeles controlado mediante un UltraChip 1601. El procesador principal se comunica mediante SPI (Serial Peripheral Interface) con el UltraChip 1601 a 2MHz y es capaz de actualizar el display enteramente en 17ms. El Display puede resultar útil en la realización de ejercicios pues puede proporcionar los valores de las variables conforme un programa está en ejecución.

**Sonido:** El sonido incluye un circuito amplificador de sonido que mejora la calidad y el volumen de éste. El ARM7 controla el sonido mediante un PWM y los filtros ayudan a reducir el ruido causado por sobremuestreo. El parlante tiene una resistencia de 16 Ohms y consume potencia según la frecuencia del sonido.

**Alimentación:** El Brick usa 6 baterías AA que se recomienda sean alcalinas o bien se use el paquete de baterías recargables de 1400mAH. Para información más detallada acerca del hardware, conexiones físicas y esquemáticos detallados, es posible descargar una guía completa de la página de LEGO en la pestaña Hardware Developer Kit.

## 3.2. Sensores y Actuadores

A continuación se presenta una breve descripción de los componentes que pueden ser conectados al Brick tanto de entrada como de salida.

### 3.2.1. Sensor de ultrasonido



Figura 3: Sensor de Ultrasonido

El sensor de ultrasonido es muy útil en las funciones del robot pues le permite detectar obstáculos y medir distancias. También puede ser usado como sensor de movimiento. Según la configuración deseada, el sensor puede medir distancias en centímetros o pulgadas con una precisión de  $\pm 3cm$  desde  $0cm$  hasta  $255cm$ .

Para poder detectar obstáculos, el sensor usa el mismo principio de los murciélagos enviando una onda de ultrasonido, que al rebotar permite al sensor estimar un tiempo y calcular la distancia en función de éste. Para su mejor funcionamiento se recomienda la detección usando superficies grandes, duras y no curvas. Este sensor se comunica con el Brick mediante  $I^2C$ .

### 3.2.2. Sensor de Luz



Figura 4: Sensor de Luz

El sensor de luz sirve para diferenciar lugares oscuros de lugares iluminados al igual que para medir la intensidad de luz de los colores.

Gracias a este sensor, es posible detectar si la luz se encuentra encendida o apagada en un cuarto, o si las cortinas están abiertas comparando la intensidad de luz que entra por una ventana. De igual forma, el robot será capaz de seguir una fuente luz. Usando un principio parecido al sensor de ultrasonido, el sensor de luz puede también identificar un rango de colores en escala de grises enviando un haz de luz y comparando con la reflexión que una superficie emite. El resultado será la obtención de un número entre 0 y 100 de completa oscuridad a luz brillante.

### 3.2.3. Sensor de Sonido



Figura 5: Sensor de Sonido

El sensor de sonido detecta la cantidad de sonido ambiental de 0 a 100 % y puede ser configurado para medir en decibelios o en decibelios ajustados. En el modo de decibelios ajustados, la sensibilidad del sensor es ajustada para que se identifiquen solamente sonidos que el oído humano puede detectar. En el modo de decibelios, se detectan algunas frecuencias que pueden estar tanto arriba como abajo del rango de audición humana.

La cantidad de presión sonora detectada puede leerse así:

- 4-5 % Un salón silencioso.
- 5-10 % Una persona hablando ubicada a una gran distancia del sensor.
- 10-30 % Una conversación mantenida cerca al sensor.
- 30-100 % Alguien gritando o música con un alto volumen.

### 3.2.4. Sensor de Rotación



Figura 6: Sensor Gyro

El sensor de rotación o sensor Gyro es fabricado por HiTechnic y detecta la rotación del robot devolviendo información acerca del número de grados rotados por segundo y de la dirección de rotación. Este sensor es muy útil para proyectos que impliquen estructuras que deban sostenerse o balancearse.

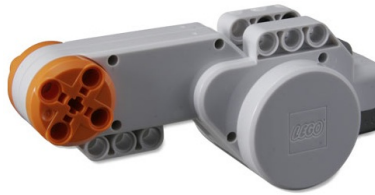


Figura 7: Servomotor

### 3.2.5. Servo-motores

El movimiento del robot está a cargo de los servo motores. Cada motor está equipado con un sensor de rotación que permite controlar el movimiento que se quiere realizar. El sensor provee información acerca de la posición en grados con una precisión de  $\pm$  un grado permitiendo al usuario controlar la velocidad de rotación. La máxima velocidad de rotación sin carga es de 170 rpm. Para controlar la velocidad se usa la rotación en términos de porcentaje (es decir de 0 a 100).

## 3.3. Programación

El Brick NXT puede ser programado por muchos ambientes dependiendo del objetivo que se quiera lograr. Se debe recordar que aunque en un principio el kit de robótica apunta a un mercado infantil, el hardware y la arquitectura interna del Brick permiten desarrollos mucho más elaborados de forma que puedan ser usados en un ambiente de educación más elevado. Algunos de los software más comunes son:

**NXT-G:** Es el programa que viene incluido en el kit y cuyo mercado son los niños. Cuenta con una interfaz gráfica basada en LabVIEW que permite una programación con íconos y a su vez la puesta en funcionamiento básica de motores y sensores. NXT-G puede ser usado en Windows y es posible transferir datos mediante USB o Bluetooth. Es una herramienta que permite ver fácilmente el flujo de control mediante su interfaz gráfica pero al ser una programación de más alto nivel, los recursos del Brick no se usan eficientemente.

**NXTOSEK:** Es una plataforma para LEGO MINDSTORMS basada en el estándar OSEK. Para poder programar el Brick con este ambiente es necesario instalar un nuevo firmware para que sea posible realizar operaciones en punto flotante. Actualmente los brick que se usan en el laboratorio de control tienen capacidad de manejo de punto flotante.

**Bricx Command Center:** Es un programa que usa un lenguaje llamado NXC (Not eXactly C) basado en NBC (Next Byte Codes) y lenguaje assembler cuya sintaxis es común con el lenguaje C. Con este software es posible implementar controladores y rutinas más complejas como se verá en el transcurso de la práctica.

## 3.4. Lenguaje NXC

NXC es un lenguaje de programación de alto nivel que es parecido en estructura a C y está basado en un compilador NBC (Next Byte Codes) que a su vez es un lenguaje con sintaxis de assembler que puede ser usado para programar los NXT mediante el *Bricx Command Center* que provee la interfaz de programación. El API está definido en un archivo conocido como *header file* y es incluido al compilar de forma predeterminada.



NXC tiene dos tipos de bloques de código: tareas (*tasks*) y funciones (*functions*). El número máximo de *tasks* y *functions* permitidas en total es de 256. Las estructuras *function* trabajan de la misma forma que en C, en donde se usan para realizar alguna operación con algunos parámetros y retornar otros, dependiendo de la rutina programada. Por su lado, las estructuras *task* pueden ser ejecutadas simultáneamente; sin embargo siempre debe existir una *task* principal (*main*) que es la que se ejecuta antes que cualquier cosa.

A continuación se puede observar un ejemplo de un programa creado con el lenguaje **NXT** usando el **Bricx Command Center** como interfaz, para el movimiento de dos servomotores conectados en los puertos de salida A y C del brick:

```
task main()
{
    OnFwd(OUT_A,75);
    OnFwd(OUT_C,75);
    Wait(4000);
    OnRev(OUT_AC,75);
    Wait(4000);
    Off(OUT_AC);
}
```

El programa inicia en la rutina **task main()** y realiza seis instrucciones. Cabe notar que de acuerdo con la sintaxis usada en lenguaje C, cada instrucción debe llevar un punto y coma al final para poder ser compilada. A continuación se describe cada línea de código:

**OnFwd(OUT\_A,75):** Esta instrucción hace funcionar el motor conectado a la salida A del brick. *OnFwd* hace que el motor se mueva hacia adelante y sus parámetros indican la salida que se quiere controlar y la rapidez con la que el motor se moverá (en porcentaje con respecto a la velocidad máxima). Es este caso el servomotor conectado al puerto A se moverá hacia adelante con una rapidez del 75 %.

**OnFwd(OUT\_C,75):** Se trata de la misma instrucción anterior para otro motor conectado al puerto de salida C del brick.

**Wait(4000):** Para que los motores continúen encendidos durante un tiempo, se usa la instrucción *Wait()* cuyo parámetro es un tiempo en milisegundos. En este caso los motores de los puertos A y C del brick se mueven 4 segundos hacia adelante.

**OnRev(OUT\_AC,75):** Tras esperar 4 segundos, ambos motores se accionan con la misma rapidez pero en la dirección contraria. Esta instrucción es equivalente a *OnFwd(OUT\_AC,-75)* pues indica moverse hacia adelante con una rapidez negativa.

**Off(OUT\_AC):** Finalmente, ambos motores se apagan.

En conclusión, si los motores fuesen el mecanismo de desplazamiento de un robot, éste se movería hacia adelante 4 segundos, luego hacia atrás otros 4 segundos y el programa terminaría. Como se observa las salidas y entradas están asociadas a constantes de la forma OUT\_A, OUT\_B, OUT\_C o bien IN\_1, IN\_2, IN\_3, IN\_4, respectivamente.

Las instrucciones usadas para el control de los sensores y actuadores varían dependiendo de la aplicación. A continuación se presenta una breve descripción de algunas instrucciones útiles al programar con NXC:



**SetSensorType(port, const type)** Esta función sirve para definir el tipo de sensor que se encuentra conectado a una entrada. *port* podría ser cualquiera de los puertos de entrada (IN\_1, IN\_2, IN\_3, IN\_4) y *const type* es una constante que indica el tipo de sensor. En la siguiente tabla se listan algunos de los valores que puede tomar este parámetro y su significado.

Tipo de Sensor	Significado
SENSOR_TYPE.TEMPERATURE	Sensor de temperatura
SENSOR_TYPE.LIGHT_ACTIVE	Sensor de luz
SENSOR_TYPE.SOUND_DB	Sensor de sonido en modo dB
SENSOR_TYPE.SOUND_DBA	Sensor de sonido en modo DBA
SENSOR_TYPE.LIGHT.LOWSPEED	Sensor digital con $I^2C$

Tabla 1: Constantes de tipo de sensor.

**Sensor(port)** Esta función sirve para leer un sensor dependiendo del tipo de sensor y de las unidades en las que se configure.

**NumOut(x,y,value,clear)** Con esta función es posible escribir números en el display del brick. *x* es la posición vertical y *y* la posición horizontal (LCD\_LINE1, LCD\_LINE2, etc.). *value* es el valor que se quiere desplegar en la pantalla LCD y *clear* es una variable booleana que permite borrar el LCD antes de escribir. La función *TextOut* que requiere los mismos parámetros sirve para escribir texto en el display del brick.

**ClearScreen()** Permite borrar el display.

**MotorRotationCount(port)** Esta función devuelve la posición angular en grados de un servomotor conectado en un puerto de salida.

## 4. PROCEDIMIENTO

La práctica se divide en dos partes las cuales se desarrollarán en dos semanas (31 de Julio y 7 de Agosto de 2012). La primera parte corresponde a la familiarización con el hardware y sus características técnicas. La segunda parte corresponde a la introducción al lenguaje de programación que se empleará a lo largo del semestre y las diferentes herramientas que contiene la interfaz gráfica de programación.

Los puntos a continuación deberán ser desarrollados en grupos y entregados en un informe la semana siguiente a la culminación de las actividades (vea detalles en la sección 5 y 6). Los puntos prácticos deberán ser sustentados en clase de laboratorio.

### 4.1. Introducción al Hardware

1. Investigar sobre las características técnicas de los elementos internos que componen la unidad básica de procesamiento (Brick) como la capacidad en memoria Flash y Ram, velocidad de procesamiento y protocolo de comunicación del procesador y coprocesador (ver Figura 2).
2. Identificar las limitaciones en alimentación y velocidad de comunicación de los puertos de entrada y salida del Brick.



3. Investigar sobre las características técnicas de los siguientes sensores: ultrasonido, EOPD, color, giroscopio, acelerómetro, sensor de contacto (Touch), brújula magnética (Compas).

## 4.2. Introducción al Software

1. Describir la estructura básica de programación bajo lenguaje NXC en BricxCC y los diferentes tipos de variables que se pueden definir dentro de este entorno.
2. Diseñar e implementar dos algoritmos: uno que conduzca el movimiento en **posición** de un servomotor siguiendo el patrón de una señal diente de sierra; y otro que conduzca el movimiento en **velocidad** de un servomotor siguiendo el patrón de una señal diente de sierra. (Ver Figura 8).
3. Diseñar e implementar un algoritmo que utilice otro sensor diferente a la posición angular del servomotor.
4. En cada algoritmo anterior adicionar código para la visualización en el Brick de los valores leídos del sensor utilizado, junto con los nombres y códigos de los integrantes del grupo.

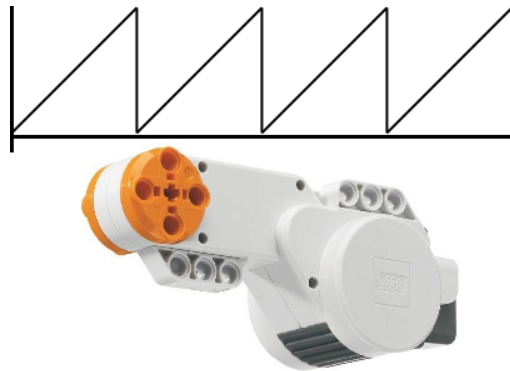


Figura 8: Señal del tipo diente de sierra

## 5. INFORME

El informe de la práctica debe usar el formato IEEE y contener los ítems mostrados a continuación:

- Título, Autores y Resumen en español.
- Introducción.
- Desarrollo de la sección PROCEDIMIENTO.
- Conclusiones.
- Bibliografía.



## 6. EVALUACIÓN

Entregar informe de la práctica según lineamientos dados en la sección 5. El informe se debe entregar únicamente vía e-mail a la dirección *horacio.unal@gmail.com* como máximo el *13 de agosto de 2012* y la evaluación sería de la siguiente manera:

- Presentación del Informe ( $P_{INF}$ ) (organización, redacción, ortografía, claridad, formato): **Valor 1.0.**
- Punto 4.1: **Valor 1.5.**
- Punto 4.2: **Valor 3.0.**
- Conclusiones (C): **Valor 0.5.**
- Ecuación de evaluación:

$$\text{Nota} = (P_{4.1} + P_{4.2} + C)P_{INF}$$