

Capítulo 1. Introducción

1.1 El Sistema de Invención Robótica (RIS 2.0)

La página principal del Grupo Lego en la sección Acerca de nosotros establece:

“Nuestro nombre es resultado de la combinación del Danés *leg godt* que significa jugar bien. Es ambos, nuestro nombre y nuestra naturaleza. Creemos que jugar es el ingrediente esencial en el crecimiento y el desarrollo de un niño. Jugar acrecienta el espíritu humano. Jugar alienta la imaginación, el pensamiento conceptual y la creación.”[1]

En este pequeño párrafo se aprecia que Grupo LEGO® está interesado en una educación de los niños que sea al mismo tiempo buena y divertida. Así mismo, se preocupa por ser una empresa de excelente calidad y estar a la vanguardia en todo aquello que sirva para este fin de aprender jugando.

1.1.1 Antecedentes

A lo largo de 20 años investigadores como Mitchel Resnick del MIT Media Lab o Brian Silverman de la compañía Playful Invention al igual que algunas empresas como Grupo LEGO®, han colaborado entre sí para diseñar una variedad de nuevas tecnologías para niños.

Específicamente, se han interesado en crear equipos de construcción, es decir, sistemas que permitan a los niños involucrarse en la creación y diseño de una diversidad de cosas que van desde simples historias animadas o esculturas interactivas hasta simulaciones, y construcciones robóticas.

La idea: lograr que los niños estén más acostumbrados al uso de nuevas tecnologías y al mismo tiempo ayudarlos a explorar importantes conceptos en las Áreas de Matemáticas, Física e Ingeniería.[2]

Es así que el Sistema de Invención Robótica de LEGO (*Robotics Invention System* o RIS) fue basado en un proyecto del MIT Media Laboratory, “El Bloque Programable” (“*The Programmable Brick*”) que se ha desarrollado desde 1986. El proyecto empezó con un trabajo previo del MIT acerca de los niños y la tecnología.[3]

1.1.2 El Hardware.

En 1998 Grupo LEGO® lanzó al mercado el MINDSTORMS™ de LEGO®, con dicho lanzamiento se cambió el concepto tradicional de *construir y jugar* de la empresa. Desde su introducción, el conjunto de herramientas de construcción y programación robótica LEGO® MINDSTORMS™ se ha convertido en el producto más vendido en la historia de Grupo LEGO®. “...el Sistema *Robotics Invention* ha alimentado la imaginación y satisfecho al inventor interno tanto de varias generaciones LEGO como el de entusiastas en robótica.”[4]



Figura 1.1. MINDSTORMS™ (Robotics Invention System 2.0)

El MINDSTORMS™ de LEGO® (producto #3804) es un kit de robot contenido en una caja, consiste de un modulo de computador (el RCX2.0™) un inventario de partes compuesto por 718 piezas TECHNICSTM - incluyendo 2 motores, 2 sensores de contacto, y un sensor de luz - y una *CONSTRUCTOPEDIA* o guía de construcción para el Robotics Invention System 2.0 con sugerencias, consejos y trucos que ayudan al usuario a familiarizarse con el kit.[5]

Además incluye un CD de instalación del programa Robotics Invention System 2.0 así como una torre infrarroja, ambos elementos son indispensables para programar la unidad RCX desde la PC.



PRODUCT DETAILS
Figura 1.2. Kit RIS 2.0



Figura 1.4. CD-ROM y Constructopedia

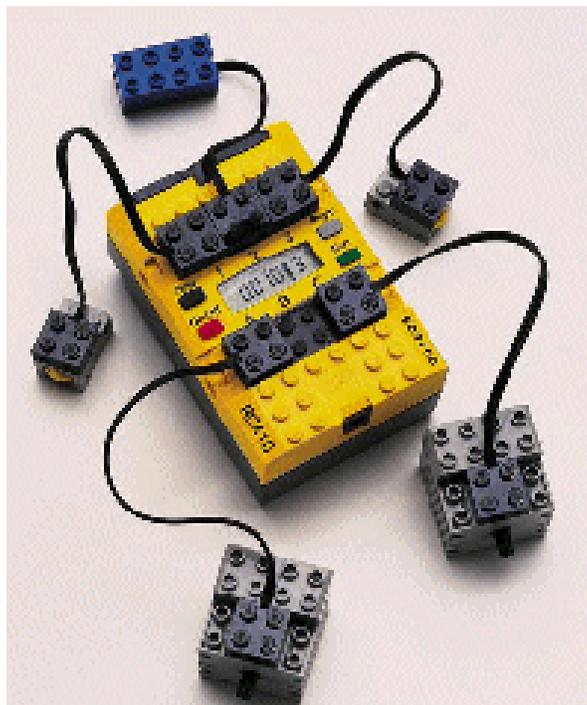


Figura 1.3. RCX 2.0, Motores y Sensores.

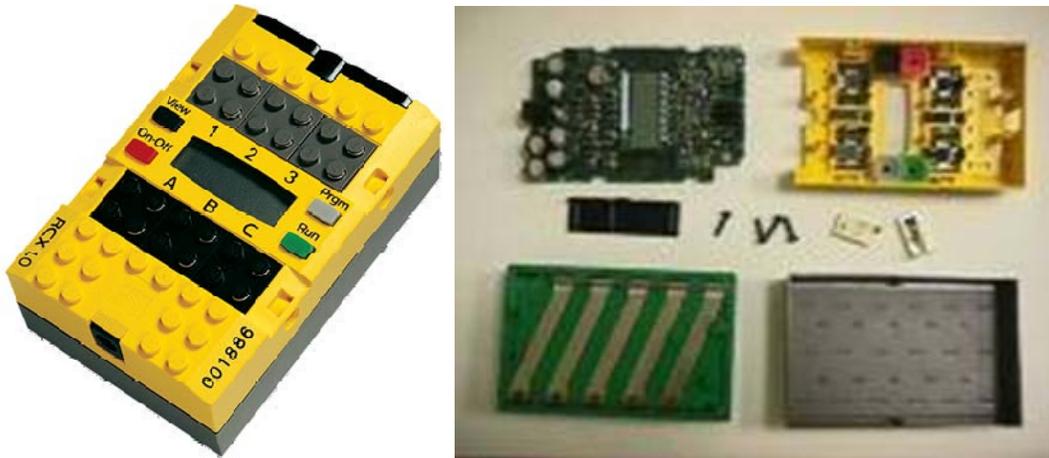


Figura 1.5. Unidad RCX™ 1.0.

Dentro de la primera versión del RCX encontramos:

- Un microcontrolador Hitachi H8/3292 con las siguientes características:
 - ✓ Procesador de 8-bit con 16-bit para direccionamiento.
 - ✓ Tamaño de ROM 8k
 - ✓ Tamaño de RAM 512
 - ✓ Velocidad de 16MHz @ 5V
 - ✓ 2 Timers 8-bit
 - ✓ 1 Timer 16-bit
 - ✓ Convertidor A/D de 8-bit
 - ✓ 43 pines E/S
 - ✓ 8 pines exclusivamente de Entrada
 - ✓ 1 puerto Serial
 - ✓ 10 salidas con capacidad de 10mA
- Además está una memoria RAM de 32k y otros dispositivos.

1.1.3 El Software.

Se ha dicho ya que el kit incluye un disco compacto para la instalación del software RIS 2.0. Cuando el usuario ejecuta el archivo de instalación en una computadora (ver el Apéndice A para conocer los requerimientos mínimos del sistema) se crean la interfaz de programación y una biblioteca con archivos de ejemplo entre los que destacan tres actividades guiadas paso a paso, que permiten al usuario familiarizarse rápidamente con el ambiente gráfico de programación. También se incluyen seis

actividades opcionales para aquel usuario que una vez concluidas las actividades del tutorial se sienta capaz de enfrentar nuevos retos.



Figura 1.6. Ventana de interfaz del Software Robotics Invention System 2.0

Sin embargo, de acuerdo con varios autores [6], el software restringe en gran medida la diversidad de aplicaciones que el kit podría desempeñar si no existieran tantas limitaciones de programación. Lo anterior promovió el desarrollo de nuevas interfaces más versátiles.

“Cuando LEGO comercializó el Sistema de Invención Robótica ofreció un atractivo hardware para montar robots con un software de limitadas funciones. Esto hizo que algunas personas se decidiesen a desarrollar nuevos entornos de programación. Entre ellos se encuentra David Baum que desarrolló el lenguaje de programación para robots LEGO NQC.”[6]

NQC es un lenguaje sencillo con una sintaxis parecida al lenguaje de programación C, que puede ser usado para programar el RCX. Es un software de distribución gratuita amparado bajo la Licencia Pública Mozilla (MPL). Originalmente creado por David Baum, utiliza las habituales estructuras de control, funciones, subrutinas y variables de C, pero no se trata de un lenguaje de programación de propósito general debido a que hay muchas restricciones originadas por las limitaciones propias del *firmware* incluido con el kit del RCX.[8]

A partir de 2004 Baum dejó en manos de John Hansen el desarrollo de nuevas versiones del NQC. En sí, NQC es un programa que utiliza comandos por línea, es decir, no tiene una interfaz gráfica.

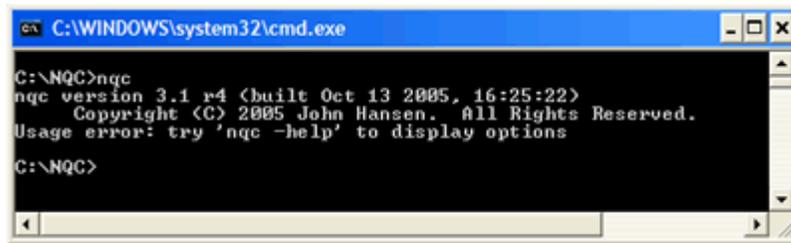


Figura 1.7. Interfaz de programación NQC.

El departamento de Informática de la Universidad de Utrecht desarrolló el RCX Command Center. Esta aplicación ofrece un ambiente de desarrollo integrado (IDE) con funciones como: panel para el control directo del robot desde el equipo PC, teclado para que el robot reproduzca música o la posibilidad de ejecutar diagnósticos. Dicha aplicación también es gratuita y está disponible en Internet. John Hansen ha desarrollado mejoras con el nombre de Bricx Command Center.

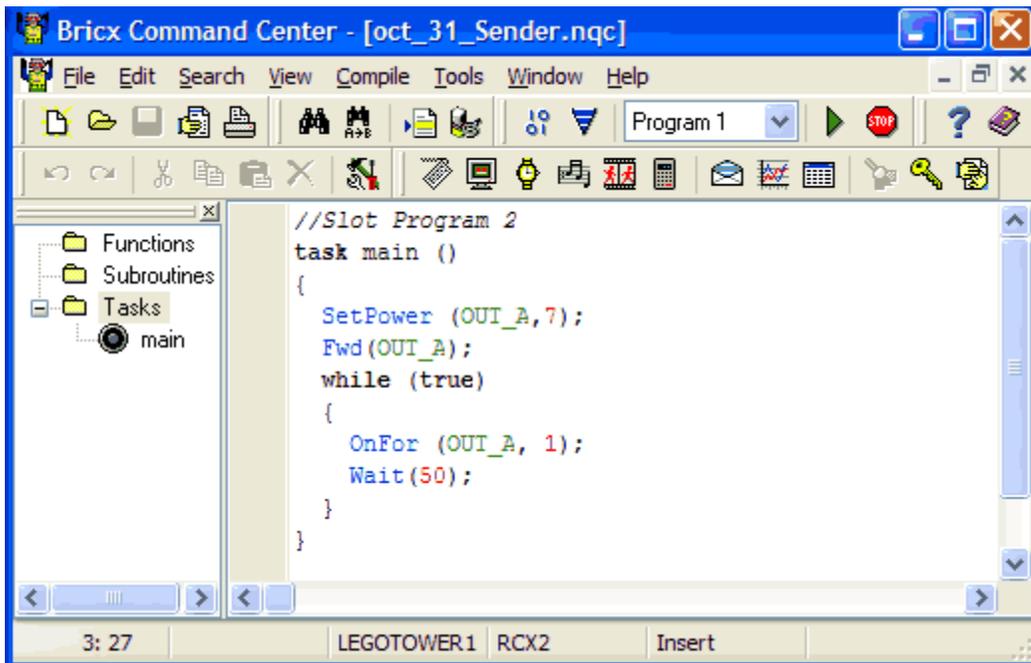


Figura 1.8. Interfaz de Programación BricxCC.

Una ventaja que ofrece este ambiente es la ventana de Templates o “plantillas” que se muestra en la Figura 1.9, ya que esta ventana incluye las instrucciones más comunes. Al dar clic en alguna de ellas se inserta el código en la ventana del archivo *nqc* exactamente donde se encuentre el cursor. Posteriormente al presionar la tecla F10 se sombrea automáticamente el siguiente campo de la instrucción que se debe cambiar. Esta forma de ingresar el código evita que se cometan errores comunes como falta de llaves, de paréntesis o puntos y comas.

Por otro lado el BricxCC incluye un archivo de ayuda que consta básicamente de dos capítulos: uno con información sobre el ambiente BricxCC y otro con información sobre NQC.

Dentro de este capítulo existen dos secciones: la primera explica detalladamente todo lo relacionado con el lenguaje: sintaxis, definición de variables y tipos de estructuras permitidas.

La segunda sección llamada NQC API define al conjunto de constantes, funciones, valores y macros que permiten el acceso a las diferentes capacidades del dispositivo como sensores, terminales de salida, temporizadores y la comunicación infrarroja.[8]

Tomando esto en consideración es fácil entender porque su relevancia en este proyecto, ya que se puede afirmar que el ambiente de programación Bricx Command Center es un software fácil de usar y bien documentado que presenta menores limitaciones de programación que el sistema Robotics Invention System 2.0.

Templates			
task .. (){..}	SetUpperLimit (... ..);	SetSensorType (... _TOUCH);	Counter (..)
sub .. (){..}	SetLowerLimit (... ..);	SetSensorType (... _LIGHT);	ClearCounter (..);
void .. (){..}	UpperLimit (..)	SetSensorType (... _TEMPERATURE);	IncCounter (..);
int ..;	LowerLimit (..)	SetSensorType (... _ROTATION);	DecCounter (..);
#define ..	SetHysteresis (... ..);		
#include ..	Hysteresis (..)	SetSensorMode (... _RAW + ..);	PlaySound (_CLICK);
#pragma reserve ..	SetClickTime (... ..);	SetSensorMode (... _BOOL);	PlaySound (_DOUBLE_BEEP);
	ClickTime (..)	SetSensorMode (... _EDGE);	PlaySound (_DOWN);
if (... ..)	SetClickCounter (... ..);	SetSensorMode (... _PULSE);	PlaySound (_UP);
if (...){..}	ClickCounter (..)	SetSensorMode (... _PERCENT);	PlaySound (_LOW_BEEP);
if (...){..}else{..}		SetSensorMode (... _CELSIUS);	PlaySound (_FAST_UP);
		SetSensorMode (... _FAHRENHEIT);	PlayTone (... ..);
		SetSensorMode (... _ROTATION);	MuteSound ();
while (...){..}	OnFwd (..);		UnmuteSound ();
do{..}while (..)	OnRev (..);	ClearSensor (..);	ClearSound ();
repeat (...){..}	Off (..);	Sensor (..)	
until (...);	Float (..);		
until (...){..}	OnFor (... ..);	Message ()	SetUserDisplay (... ..);
switch (...){case ...:break;default ...}	SetPower (... ..);	SendMessage (..);	SetRandomSeed (...);
acquire (ACQUIRE_){..}catch{..}	Wait (..);	ClearMessage ();	SetWatch (... ..);
monitor (EVENT_MASK (...)){..}catch{..}	SetSensor (... _TOUCH);	CreateDatalog (..);	SetSleepTime (..);
	SetSensor (... _LIGHT);	AddToDatalog (..);	SleepNow ();
	SetSensor (... _ROTATION);		BatteryLevel ()
	SetSensor (... _CELSIUS);		
start ..;	SetSensor (... _FAHRENHEIT);	Timer (..)	
stop ..;	SetSensor (... _PULSE);	ClearTimer (..);	
StopAllTasks ();	SetSensor (... _EDGE);	SetTimer (... ..);	
SelectProgram (...);		FastTimer (..)	
SetPriority (...);			
ActiveEvents (...);			
SetEvent (... ..);			
ClearEvent (...);			
ClearAllEvents ();			
EventState (..)			
CalibrateEvent (... ..);			

Figura 1.9. Ventana de Plantillas.

1.2 Justificación del Proyecto

Podemos afirmar que el kit mencionado anteriormente está limitado en cuanto a posibilidades de detección de variables físicas, ya que únicamente cuenta con dos sensores de contacto, un sensor de intensidad de luz y tres puertos de entrada. No obstante que hay expansiones comerciales diseñadas por la misma empresa que incluyen un sensor de temperatura y un sensor de rotación, éstas son en general costosas. Actualmente, la empresa LEGO lanzó al mercado el modelo LEGO® MINDSTORMS™ NXT, que incluye capacidades de percepción de sonido, visión ultrasónica, y versiones mejoradas del sensor de contacto y de luz.

El interés por este proyecto surge de la necesidad de aprovechar los kits disponibles actualmente en el laboratorio para el curso introductorio de mecatrónica, proporcionando al estudiante de accesorios adicionales para la realización de diversas aplicaciones.

Aunque en esencia el kit es un juguete destinado para cualquier niño que cumpla con el único requisito de ser mayor a 12 años (edad recomendada por la empresa), se ha tenido la experiencia en la *Universidad de las Américas* de que este kit puede ser utilizado para diferentes proyectos de experimentación, al menos como un comienzo: puede ayudar al estudiante a darse una idea de cómo medir un proceso simple, o decidir si cierta estructura es mejor que otra para soportar un peso específico sin tener que gastar demasiado tiempo construyéndola, puede ayudar a un profesor a explicar cierto algoritmo ante sus estudiantes sin necesidad de pasar mucho tiempo en el laboratorio, incluso se ha visto a estudiantes utilizar el kit simplemente para hacer simulaciones a escala de sistemas reales como elevadores, o grúas. En muchos de los casos el estudiante construirá posteriormente sus propios dispositivos sensores y actuadores, así como estructuras; con funciones más específicas y más apegadas a su proyecto, pero al menos el kit les ha servido ya como una primera aproximación.

Como objetivo de esta tesis se propone extender las capacidades actuales del RCX 2.0, el cual sirve como unidad de control (“cerebro”) de las invenciones logradas usando el kit MINDSTORMS™ de LEGO®, y de este modo ampliar las posibilidades de creación. La principal ampliación que se pretende realizar tiene que ver con el número de variables físicas que el robot puede percibir, por lo que se requieren revisar algunos conceptos en el área de sensores, para que a partir de ellos se puedan desarrollar nuevos circuitos electrónicos, que respeten las especificaciones técnicas actuales del RCX de

modo que sean compatibles (no dañen el microcontrolador mediante la demanda excesiva de corriente por ejemplo), seguros (respeten las normas para que no existan riesgos de corto-circuito o sobrevoltaje) y el kit completo pueda seguir siendo utilizado como un juguete.

Específicamente se pretenden construir los siguientes elementos:

- Multiplexor de entradas digitales y analógicas.
- Sensor de proximidad (ultrasonido).
- Sensor de temperatura.