

# Desarrollo e implementación de un sistema de teleoperación utilizando lego mindstorms

Develop and Implementation of a Teleoperation System  
Using Lego Mindstorms

FERNANDO LÓPEZ ARIAS

Estudiante de Tecnología en Electrónica de la Universidad Distrital Francisco José de Caldas, Facultad Tecnológica, miembro del grupo de investigación ROMA.  
fdo.web@gmail.com

Rafael Antonio Niño Vargas

Estudiante de Tecnología en Electrónica de la Universidad Distrital Francisco José de Caldas, Facultad Tecnológica, miembro del grupo de investigación ROMA.  
diplorein08@yahoo.es

DORA LILIA CASTAÑEDA TIBAQUIRÁ

Ingeniera electrónica Universidad Distrital Francisco José De Caldas; Especialista en Telecomunicaciones Móviles; miembro del grupo de investigación ROMA, docente adscrita a la Facultad Tecnológica de la Universidad Distrital.  
dlcastanedat@udistrital.edu.co.

Fecha de recepción: 14 de septiembre de 2007

Clasificación del artículo: investigación

Fecha de aceptación: 18 de febrero de 2008

**Palabras clave:** teleoperación, robótica, comunicación infrarroja, fútbol robótico.

**Key words:** teleoperation, robotics, infrared communication, robot soccer.

## RESUMEN

En este artículo se presenta el desarrollo de un sistema de teleoperación supervisado; para éste se utiliza Robotics Invention System 2.0 de Lego Mindstorms, por su facilidad de uso y adaptabilidad a funciones específicas. Para una aplicación en fútbol robótico, constituido por dos plataformas R.I.S 2.0, las cuales son monitoreadas y controladas desde una estación remota por un operador humano

para cada una y con esto lograr incursionar en el área de la teleoperación, manejando estas plataformas robóticas para nuevas aplicaciones dentro del grupo de investigación ROMA (Robótica Móvil Autónoma).

## ABSTRACT

This paper presents the development of a supervised teleoperation system using Lego Mindstorms' Robotics Invention System 2.0. The kit is easy to

use and adaptable to specific functions, like an application in robot soccer, made up with two R.I.S. 2.0 platforms, which will be monitored and controlled from a remote station by human operators individually. The main goal is the incursion into

teleoperation area by the use of robotics platforms in new applications at the research group ROMA, Autonomous – Mobile Robotics, by its name in Spanish.

\* \* \*

## 1. Introducción

A través de la historia, el ser humano ha desarrollado máquinas que tienen como finalidad aumentar sus destrezas para realizar alguna tarea, en la cual su participación es necesaria para lograr el objetivo con el cual fue construida; asimismo, con el pasar de los años y para nuevas aplicaciones en Robótica, se encuentran situaciones y lugares peligrosos, poco asequibles al ser humano, así como entornos no estructurados y dinámicos, en los cuales los problemas de percepción y planificación automática son complejos, ver [1]. Lo que implica robots autónomos u operados desde una estación remota, para controlar sus actividades, por medio de un sistema de comunicación que permita un seguimiento y abstracción del entorno, en el cual se mueve el robot, utilizando para ello cámaras o sistemas de percepción del entorno, distintos de acuerdo con la aplicación y funcionalidad del robot.

La investigación se basa en la teleoperación de plataformas robóticas existentes (R.I.S. 2.0 de Lego Mindstorms), que maneja el grupo de investigación ROMA. Lego Mindstorms es una plataforma para el diseño y desarrollo de robots, que sigue la filosofía de la marca LEGO, armar y construir todo tipo de objetos uniendo bloques interconectables, ver [2]; cuyo nombre viene de las palabras en danés *leg godt* (jugar bien), y de *Mindstorms* (lluvia de ideas), ver [2].

La investigación se realiza, a fin de poder sentar las bases en teleoperación de estos dispositivos y así generar nuevos proyectos, cumpliendo los objetivos del grupo basados en su misión de proveer a la comunidad académica de la Universidad Distrital y al país, en general, nuevos conocimientos y recursos (productos) en el área de la robótica móvil, con una

sencilla aplicación de fútbol robótico, mediante un control supervisado (*supervisory*), ver [1].

Para lograr el objetivo descrito en el párrafo anterior es necesaria la realización de pruebas de comunicación entre las plataformas robóticas de Lego y el PC, basado en la revisión bibliográfica tanto de hardware como de software. Es así como en el presente artículo se muestra el cumplimiento de los objetivos propuestos y su descripción, en particular.

## 2. Sistema teleoperado supervisado

En un sistema teleoperado supervisado, el hombre dirige y monitoriza las actividades de un sistema semiautónomo de control, ver [1]. Para este caso, es la plataforma robótica RIS (Robotics Invention System 2.0 de Lego Mindstorms), supervisada mediante una cámara, que da el posicionamiento del jugador robótico dentro del área de acción. Constituido por un ladrillo programable RCX (Robot Control System) de Lego, y su torre de comunicación por infrarrojo como se ve en la figura 1; es controlado mediante un computador dirigido por su operador humano.

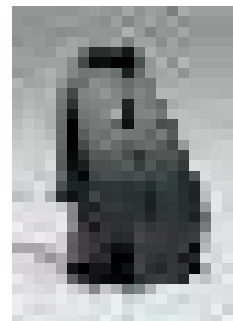


Figura 1. Torre Infrarroja por USB (<http://mindstorms.lego.com/> recuperado el 6 de agosto de 2007).

## 2.1 Prototipo robótico RIS 2.0

Se utiliza un bloque RCX de Lego, como se observa en la figura 2 por cada jugador; éste es el cerebro de *Robotics Invention System*, que contiene el sistema de comandos para el robot. Es una mini computadora integrada en un ladrillo Lego, con tres puertos de salida A, B y C para conectar motores y lámparas, y tres puertos de entrada 1, 2 y 3 para conectar sensores. Se puede programar con un PC o utilizar uno de los cinco programas que tiene integrados.

El núcleo del RCX es un microcontrolador Hitachi H8, exactamente el H8/3292. Tiene una velocidad de funcionamiento de 16 MHz y está alimentado con 5 V, ver [2].



**Figura 2.** Ladrillo RCX de R.I.S. 2.0 de Lego Mindstorms (<http://mindstorms.lego.com/> recuperado el 6 de agosto de 2007).

### 2.1.1 Programación del RCX

Para la aplicación del fútbol robótico, el programa que lleva el RCX se desarrolla con el Software Robolab 2.5.4, por su facilidad en cuanto a su lenguaje y a la aplicabilidad dentro del grupo de investigación ROMA. Como son dos bloques RCX, cada uno debe tener un programa distinto en cuanto a la recepción de datos por parte del PC, por medio de una torre de comunicación por infrarrojo, pero semejantes en cuanto a funcionalidad. De acuerdo con el dato recibido, el RCX ejecuta movimientos hacia adelante, atrás, izquierda, derecha, o sin movimiento, sin generar respuesta confirmando la recepción de un dato.

## 2.2 Protocolo de comunicación

Cada envío y respuesta es enviado como un paquete de bytes en comunicación serial entre el computador y el RCX. La configuración serial está conformada por dos equipos transmisores/receptores, uno conectado al computador vía puerto RS232 o USB y el otro aquél que incluye el RCX. La velocidad es de 2.400 baudios; cada byte se envía con un bit de inicio al comienzo y un bit de stop al final; también se incluye un bit de paridad impar en cada byte. De esta manera cada paquete es en realidad de 11 bits así: 1 bit de inicio, 8 bits de datos, 1 bit de paridad y 1 bit de stop. En realidad cada byte, excepto el primero “0x55”, es enviado dos veces, pero el segundo es el complemento del primero. El resultado es un paquete con un número igual de unos y ceros (el byte de encabezado “0x55” también tiene igual cantidad de unos y ceros). La razón de enviar el complemento es para que la duración de los mensajes sea consistente, ver [3].

El RCX responde a Opcodes o códigos de operación, los cuales le indican qué tipo de acción debe ejecutar. Estos Opcodes están codificados según números. Por lo general el paquete empieza con los siguientes tres bytes o encabezado: (0x55 0xFF 0x00). Finalmente, se envía un *Checksum* que suma los valores de los códigos enviados en un byte precedido de su complemento. El Opcode para enviar un mensaje de un byte de longitud al RCX es “0xF7” acompañado del byte del mensaje y su complemento.

La trama para enviar un mensaje directo (número 3) al RCX es:

```
0x55 0xFF 0x00 0xF7 0x08 0x03 0xFC 0xFA
0x05
```

Encabezado Opcode Message Checksum

Checksum = Opcode + Message

Checksum = 0xF7 + 0x03 = 0xFA

Complemento de 0xFA = 0x05.

Checksum = “0xFA 0x05”

Ver [4].

### 2.2.1 Sistema anti-colisión de datos

Cuando se trabaja con sistemas de comunicación vía infrarrojo, dada la naturaleza del sistema, no es posible trabajar comunicación de varios canales como los que ofrecen las distintas frecuencias cuando se trabaja con Radio Frecuencias (RF). Entonces en lo que respecta al proyecto, la teleoperación de dos robots por vía infrarroja desde dos estaciones (torres USB) diferentes acarrea varios inconvenientes:

- El software del RCX (robot) conocido como *firmware*, reenvía datos a la torre para hacerle

saber a esta que el dato o comando se recibe satisfactoriamente. Debido a esto, al enviar, por ejemplo, el operador uno un dato al robot uno, habrá respuesta por parte de ambos robots; incluso, si por programa se le dice al robot dos que no tenga respuesta alguna a ese comando, éste responderá para confirmar cualquier recepción. Estas respuestas simultáneas se entremezclan o traslapan, conformando un dato ilegible que llega tanto a torres como a los RCX, probablemente generando un error, como se ve en la figura 3.

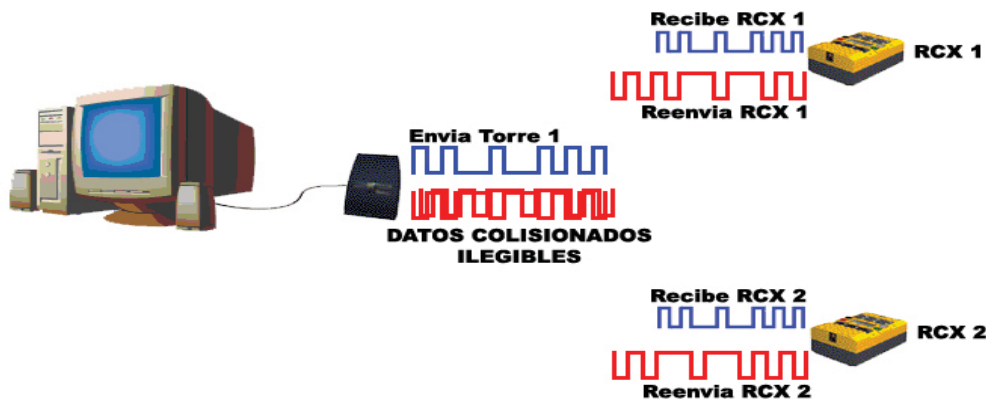


Figura 3 Interferencia por respuesta simultánea de los dos RCX (fuente: los autores).

Para solucionar la entremezcla o colisión de datos, se hace uso de un método llamado *Broadcasting* (emisión), cuya característica es la emisión de datos, sin que exista retroalimentación. En el *firmware* del Lego existe un opcode que es el único que no reenvía dato alguno al ser recibido; este opcode es el mensaje entre RCX o correo (“msgs #”) y puede ser enviado desde la torre USB, de esta manera se hace que el RCX reaccione solo con valores de correo preestablecidos.

- El envío de datos sin que exista retroalimentación parece ser la solución; sin embargo, ante el uso de dos o más robots puede ocurrir que el operador uno y el operador dos envíen un dato a sus respectivos robots al mismo tiempo. Sin importar la velocidad de transmisión de los datos, la colisión de datos sucederá, y ambos

RCX reciben un dato corrupto, que hace que se ejecute la acción errónea o simplemente no la hagan. Ver figura 4.

Para evitar que ambos operadores transmitan datos simultáneamente es necesario que de alguna manera las torres se sincronicen; esto con el objeto de dar prioridad a alguna de ellas, para que envíe sus datos a su robot; acto seguido, la torre uno da “permiso” a la otra torre de enviar sus datos a su robot, mientras la primera torre queda en “silencio”, es decir, no envía dato alguno.

### 2.2.2 Protocolo desarrollado como solución

A fin de evitar choque de información a través del medio, se plantea un sistema conformado por dos torres infrarrojas, cada una para un RCX, donde una de ellas se ha denominado “maestra”, y la segunda

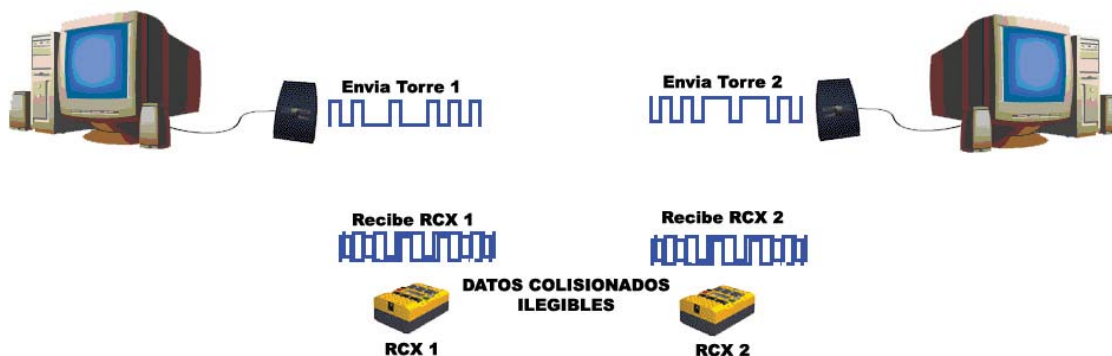


Figura 4. Interferencia por envío simultáneo de datos por las dos torres infrarrojas (fuente: los autores).

torre “esclava”, dependiente de las órdenes de la torre maestra.

*Sistema maestro*, correspondiente al software de control, la torre de comunicación maestra y el primer PC, consiste en la ejecución de órdenes propias, cada cierto periodo de tiempo (300 ms), correspondiente al envío de la orden, seguida del mensaje “100”, como se especifica en el siguiente diagrama.

*El Sistema esclavo* corresponde al software de control, la torre de comunicación esclava y el segundo PC. Ésta espera mensajes e ignora si el dato recibido no es “100”; si es así, es habilitada para enviar un mensaje al RCX correspondiente, y queda nuevamente en modo de espera, según está planteado en el diagrama 2.

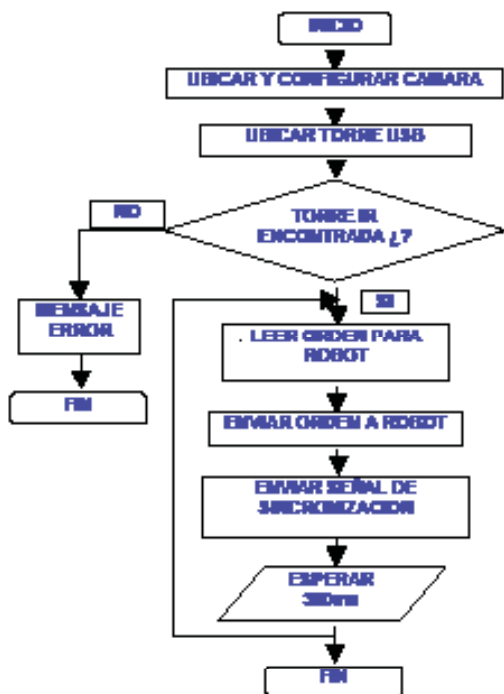


Diagrama 1. Diagrama de flujo del sistema maestro (fuente: los autores).

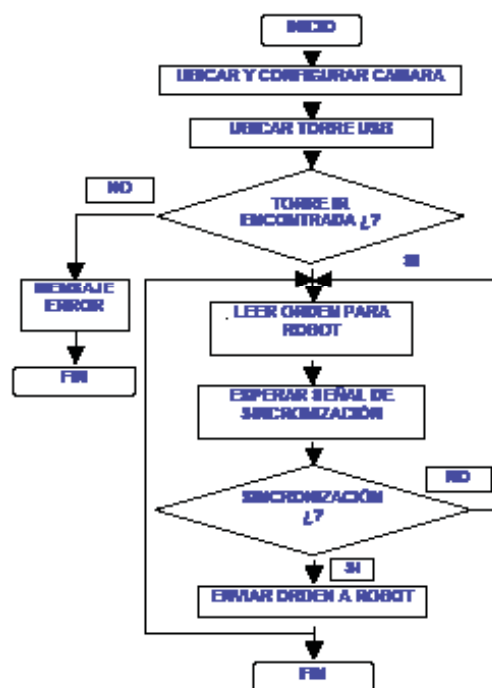


Diagrama 2. Diagrama de flujo del sistema esclavo (fuente: los autores).

El mensaje “100” es el encargado de dar sincronía a la transmisión de datos, por parte de la torre maestra y la esclava, a fin de evitar la transmisión simultánea.

De esta manera las órdenes al RCX maestro son enviadas cada 300 m, seguidas del mensaje “100”, que reconoce la torre esclava y el cual le permite enviar sus comandos al RCX esclavo. Es inevitable

un retardo, debido al tiempo de respuesta generado por el tiempo de apertura del puerto como receptor, el tiempo de lectura del puerto y el tiempo de procesamiento del dato recibido, ver la figura 5.

Esta sincronía permite concluir que una comunicación bidireccional es lenta para un sistema de teleoperación y se puede hacer aún más lenta si se incluyen varios robots.

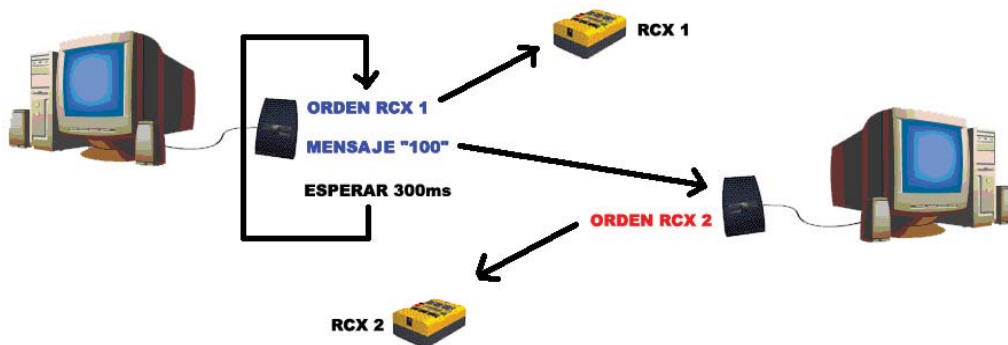


Figura 5. Protocolo de comunicación sincronizado (fuente: los autores).

### 3. Supervisión de los robots mediante cámara

Dado que se trabaja con sistemas de teleoperación supervisada, es necesaria la utilización de cámaras para observar los movimientos de los RCX en el campo de juego; esto se implementa de manera individual, es decir, una cámara por cada operador, en este caso son dos equipos, lo que proporciona una perspectiva única del campo de juego para cada operador humano.

La supervisión por medio de la cámara suministra un video muy cercano al de tiempo real; con esto evita fallas en la teleoperación de los RCX, por demora en la recepción y procesamiento de las imágenes. Además, con esta imagen en vivo, el operador puede enviar órdenes y obtener una retroalimentación de la acción ejecutada.

La cámara utilizada fue una LEGO CAM, como se ve en la figura 6; ésta registra hasta treinta tomas por segundo, además tiene un micrófono incorporado para capturar sonidos, e incluye una conexión USB

para fácil instalación. Ésta puede reaccionar a lo que ve, detectar cambios en la luz, el movimiento e incluso el color; pero en este caso sólo se utiliza para la supervisión de los RCX, a través del video proporcionado por LEGO CAM.



Figura 6. LEGO CAM (<http://mindstorms.lego.com> recuperado el 6 de agosto de 2007).

Características de LEGO Cam: conexión USB (5 metros de cable), foco ajustable, micrófono incorporado, treinta tomas por segundo, alta resolución de color en las imágenes: 352 x 288, ver [5].

### 4. Interfaz de teleoperación

La interfaz de teleoperación es el medio de interacción entre el operador humano y la plataforma



Lego; ésta se realiza mediante un computador, el cual tiene conectado vía USB, una torre de comunicación por infrarrojo para realizar el envío de datos al RCX. El software permite al usuario teleoperar al RCX al dirigir el movimiento y controlar su dirección de avance, para cumplir con la aplicación de jugar fútbol robótico. Para el cumplimiento de este objetivo, se trabaja con Visual Basic Express Edition [6], un software de programación, de registro gratuito y libre, de fácil instalación, incluso en computadores sin acceso a Internet [7]. Esto permite trabajar con Robotics Invention System 2.0 de Lego Mindstorms, mediante sintaxis LASM, *assembly instructions set* de LEGO para interpretar y ejecutar programas desde el PC.

### 4.1 Lego (LASM)

Para Lego Mindstorms existen comandos por bytes de operación (*byte code commands*), los cuales son interpretados por el RCX o máquina virtual, también llamados Opcodes. Estos, mediante una sintaxis en ensamblador de LEGO (LASM), permiten ser ejecutados y descritos mediante software que permita la interpretación de esta sintaxis LASM. Esto significa que un comando para enviar mensajes en un programa es escrito en LASM como “msg5”, pero el *byte code* generado es “0xF7 0x05”, donde “0xF7” es el Opcode para los mensajes, y “0x05” es el valor del mensaje en hexadecimal. Si se tiene en cuenta el protocolo que se trabaja, en esta transmisión de datos, a cada byte se le agrega su complemento; es por esto que el Opcode queda: “0xF7 0x08”, y el mensaje “0x05 0xFA”. Para que LASM fuera interpretado en el Lenguaje Visual Basic, es necesaria la utilización de un SDK especializado para Lego, ver [8].

### 4.2 Lego Mindstorms SDK 2.5

SDK (Software Development Kit - Kit de desarrollo de software o devkit). Un SDK es un conjunto de herramientas y programas de desarrollo, el cual permite crear aplicaciones para un determinado paquete de software, estructura de software, plataforma de hardware, sistema de computadora,

consulta de videojuego, sistema operativo o similar. Por su parte, el SDK 2.5 de Lego es un paquete gratuito ofrecido por Lego, para el desarrollo de aplicaciones con RIS de Lego bajo plataformas Visual Basic y C++, ver [9].

Para el caso de Visual Basic, este ofrece una librería de tipos LEGO VPBrick 2.1 type Library, la cual incluye propiedades, métodos y eventos que permiten una comunicación directa con la torre del Lego, para la manipulación de todas las instrucciones reconocibles por el *firmware* estándar del RCX, pero bajo parámetros construidos en Visual Basic. Para su utilización se instala el SDK, acto seguido en el formulario se debe añadir la referencia LEGO VPBrick 2.1 type Library.

### 4.3 Logitech QuickCam® SDK 1.0

Debido a que la cámara utilizada por Lego Mindstorms (LEGO CAM) no es más que una cámara Logitech QuickCam, se emplea este SDK para poder fusionar la cámara con la aplicación en Visual Basic. Este SDK permite el desarrollo de herramientas sofisticadas, con base en el hardware de la cámara en entornos Visual Basic y C++. Por ejemplo, ajustes de las propiedades de la cámara como color, brillo, contraste, etc. Grabación de video, toma de fotografías, superposición de texto, detección de movimiento, entre otros [10].

## 5. Resultados

Con la utilización de software como Robolab 2.5.4, y Visual Basic Express Edition, se logra la implementación de un sistema teleoperado supervisado con la plataforma Robótica RIS 2.0 de Lego Mindstorms en Fútbol Robótico. Así se controla la comunicación entre las torres infrarrojas y los RCX y se evita colisión de datos durante la transmisión, gracias al protocolo de comunicación por prioridad implementado a la aplicación de fútbol robótico.

### 5.1 Prototipo robótico

El prototipo robótico se ensambla utilizando RIS 2.0 de Lego, para desempeñarse como jugador en el

campo de fútbol para esta aplicación (ver figura 7). El prototipo robótico tiene en su diseño la facilidad de girar, y poder guiar la pelota a través del campo de juego (ver figura 10); este es armado como una plataforma móvil con dos motores en modo diferencial, cuatro ruedas y las respectivas fichas que complementan a este jugador; esto se realiza teniendo en cuenta que debe ser controlado por un operador humano, por medio de una comunicación infrarroja desde el computador.



Figura 7. "Jugador" Fútbol Robótico.

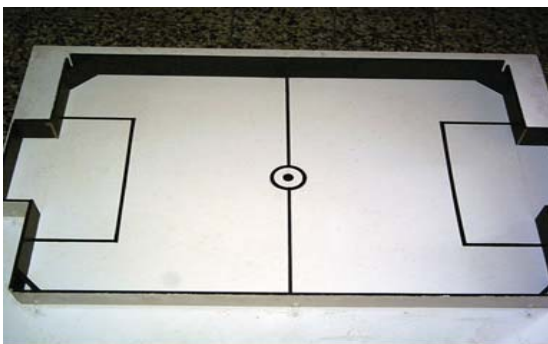


Figura 8. Fotografía del campo de juego, Fútbol Robótico.

### 5.2 Protocolo de comunicación

Se desarrolla un protocolo de comunicación consistente en el envío de datos, por medio de transmisión inalámbrica, ya que los datos son emitidos utilizando luz infrarroja, provista por la torre específica de comunicación para Lego Mindstorms. El protocolo consiste en evitar la transmisión de datos en un mismo instante de tiempo, por parte

de las dos torres que controlan a su respectivo RCX. Esto se logra por medio de una sincronización entre torres, evitando (mediante el envío de comandos específicos LASM al RCX) la no generación de respuesta por parte de estos módulos a las torres infrarrojas, obteniendo una teleoperación propia para la aplicación en fútbol robótico.

### 5.3 Utilización de cámara para la supervisión

Como se observa en la figura 9, por medio de la cámara, es posible supervisar la posición de los robots y sus movimientos, con el objeto de dar al operador humano una mayor perspectiva de lo que ocurre en el campo de juego, la respuesta de cada orden dada. Así se logra la finalidad de una teleoperación supervisada.

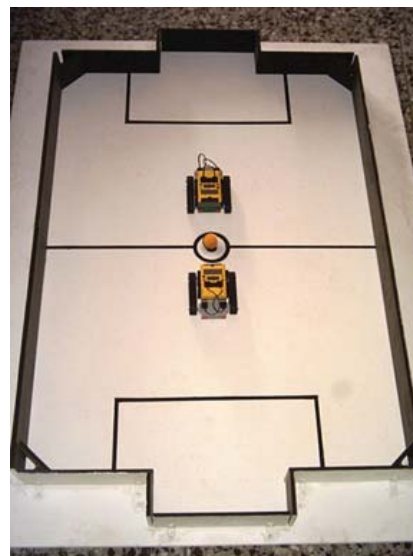


Figura 9. Fotografía de la supervisión de los RCX por medio de la cámara Lego Cam.

### 5.4 Interfaz de teleoperación

Para el buen desarrollo del proyecto fue necesario generar una interfaz que relacione al prototipo robótico de Lego con el manipulador humano. Esto se hizo a través de un computador y mediante Visual Basic Express Edition, en el cual, como se observa en la figura 10, se tienen las opciones de mover el robot hacia adelante, a la derecha, hacia atrás o a la izquierda, y una opción de stop; además,



## con-ciencias

se hace un reconocimiento de la torre infrarroja, para enviar los datos que gobiernan los movimientos seleccionados por el operador, con la finalidad de poder jugar fútbol robótico desde una estación remota con los prototipos robóticos de Lego.



Figura 10. Interfaz de teleoperación bajo Visual Basic Express Edition.

### 6. Conclusiones

El envío de información inalámbrica por medio de una torre con transmisión de datos por infrarrojo, en particular para el bloque RCX de Lego, es posible. Siempre y cuando no exista otra transmisión de datos por el mismo medio, al mismo tiempo, se abre la posibilidad de implementar la captura de datos a través del RCX y el envío de dicha información; además hay una confirmación de los datos recibidos de la torre, lo que hace más confiable la comunicación.

La teleoperación de dos o más RCX implica un retardo en el envío de información, dado que hay

que priorizar la transmisión de datos. Por consiguiente, se realiza una sincronización por parte de las torres para el envío de la información, por el contrario, las órdenes dadas por el operador nunca son interpretadas por los RCX.

La desventaja de la comunicación por infrarrojo es su sensibilidad al ruido producido por la luz; igualmente su alcance es limitado por la poca potencia que maneja las torres de transmisión y depende de una línea visual directa, ya que cualquier obstáculo desvía las ondas de luz (reflexión de la luz).

Mediante la ejecución de este proyecto se definen las bases para el desarrollo de nuevas aplicaciones

en robótica, utilizando Robotics Invention System 2.0 de Lego, empleando lo ya desarrollado en el

campo de la comunicación, y en especial en sistemas de teleoperación de estos dispositivos.

### Referencias bibliográficas (t2)

- [1] Ollero. B., Anibal. (2001) *Robótica Manipuladores y Robóts Móviles*. Barcelona (España): Alfaomega.
- [2] Carnicero. R. F.; D'onofrio J.M.; De Arteaga del Alamo. I.; De Santiago, R.; Gualda, G.D.; López, D.E.; Martín E., J.A. y Mateus, A.J. (2006) *Introducción al diseño de microrobóts móviles* [en línea]. Disponible en: <http://www.depeca.uah.es/docencia/LibreEleccion/IDMRM/> [recuperado el 20 de abril de 2007].
- [3] RCX Manual. (Sin fecha) *Ole Caprani, University of Aarhus, Department of Computer Science* [en línea]. Disponible en: <http://legolab.daimi.au.dk/CSaEA/RCX/Manual.dir/RCXManual.html> [recuperado el 20 de abril de 2007].
- [4] RCX Internals. (Sin fecha) <http://graphics.stanford.edu/~kekoa/rcx/#OpCodes> [recuperado el 20 de abril de 2007].
- [5] Lego Mindstorms. (2007a) *Vision Command Product Details* [en línea]. Disponible en: <http://mindstorms.lego.com/eng/products/vc/vcdetails.asp> [recuperado el 9 de agosto de 2007].
- [6] España MSDN. (2007) *Microsoft Visual Basic 2005 Express Edition* [en línea]. Disponible en: <http://www.microsoft.com/spanish/msdn/vstudio/express/VB/default.mspx> [recuperado el 3 de agosto de 2007].
- [7] Microsoft.com. (2007) *Visual Studio Express Developer Center* [en línea]. Disponible en: <http://msdn.microsoft.com/vstudio/express/support/install/#cd> [recuperado el 3 de agosto de 2007].
- [8] Lego Mindstorms. (2007b) *Lego Mindstorms SDK 2.0* [en línea]. Disponible en: <http://mindstorms.lego.com/sdk2/> [recuperado el 6 agosto de 2007].
- [9] Lego Mindstorms. (2007c) *Lego Mindstorms SDK 2.5* [en línea]. Disponible en: <http://mindstorms.lego.com/sdk2point5> [recuperado el 6 de agosto de 2007].
- [10] Logitech. (Sin fecha) *Press Releases* [en línea]. **Disponible en:** <http://www.logitech.com/index.cfm/172/1242&cl&cl=US,EN> [recuperado el 6 de agosto de 2007].



Plaza de Bolívar, 12 de abril de 1948. Fuente: anónimo, en archivo José Vicente Ortega Ricaurte (Sociedad de Mejoras y Ornato de Bogotá)