

# Diseño y construcción de un robot móvil orientado a la enseñanza e investigación\*

Nelson David Muñoz\*\*, Carlos Andrés Andrade\*\*\* y  
Nelson Londoño Ospina\*\*\*\*

---

## Resumen

*Se presenta el diseño de hardware y software de una plataforma robótica móvil, útil para realizar labores de enseñanza e investigación en algoritmos de control para vehículos autónomos o teleoperados. El robot cuenta con un microcontrolador MC9S12 de 16 bits; posee sensores de luz infrarroja, ultrasonido, iluminación y encoders; además, cuenta con puertos libres para futuras expansiones y un sistema de comunicación inalámbrica. Se describen las principales características del prototipo construido.*

**Palabras claves:** Robótica móvil, control de robots, inteligencia computacional, microcontroladores.

## Abstract

*The hardware and software design of a mobile robotic platform is showed, useful to education and research in algorithms of autonomous control or teleoperation vehicles. The robot is based on 16 bits microcontroller, MC9S12; has sensors of infrared light, ultrasound, illumination and encoders; in addition it counts on free ports for future expansions and a wireless communication system. The main characteristics of the constructed prototype are described.*

**Key words:** Mobile robotics, control of robots, computational intelligence, microcontroller.

Fecha de recepción: 22 de febrero de 2006  
Fecha de aceptación: 30 de abril de 2006

---

\* Este trabajo fue desarrollado por el Grupo de Investigación en Robótica y Mecatrónica, adscrito al GIMEL [<http://ingenieria.udea.edu.co/giraa/>], en el marco del proyecto MC02-1-09, financiado por el Comité para el Desarrollo de la Investigación (CODI) de la Universidad de Antioquia (Colombia).

\*\* Ingeniero electrónico, profesor de diseño de circuitos digitales, Departamento de Ingeniería de Sistemas, Universidad de Antioquia. [nmunoz@udea.edu.co](mailto:nmunoz@udea.edu.co)

\*\*\* Ingeniero electrónico, Área de Hidrometría e Instrumentación, Empresas Públicas de Medellín. [andrax@epm.net.co](mailto:andrax@epm.net.co)

\*\*\*\* Ingeniero electrónico, profesor de control automático, Departamento de Ingeniería Eléctrica, Universidad de Antioquia. [nlondono@udea.edu.co](mailto:nlondono@udea.edu.co)

## INTRODUCCIÓN

Una plataforma robótica móvil es un robot para experimentación e investigación en ambientes reales. Estos vehículos poseen características similares a los robots de aplicación industrial o comercial. Las plataformas robóticas se emplean intensivamente en la fase de desarrollo de proyectos de investigación aplicada de sistemas robotizados, en la que, por ejemplo, se utilizan para la prueba y validación de arquitecturas de control de robots, o para examinar algoritmos de navegación autónoma o semi-autónoma y con diferentes tipos de sensores, actuadores y demás dispositivos electrónicos. Como ejemplos se pueden mencionar los siguientes: robot Khepera [1], Nomad 200 [2], Magellan y B21R [3], plataformas ampliamente utilizadas para experimentación por investigadores en las áreas de robótica y sus afines.

La compra o adquisición de una de estas plataformas robóticas generalmente trae asociados altos costos, por trámites de importación, adquisición de repuestos, licencias de *software* propietario, soporte técnico, etc. Por esta razón, grupos de I+D en robótica y automatización que cuentan con presupuesto reducido, optan por desarrollar sus propios prototipos. En el ámbito local, varias instituciones universitarias han desarrollado diversos robots experimentales, entre los que se encuentran los siguientes: el Gonzo, de la Universidad del Valle [4]; Telémaco y Walker I, de la Universidad del Norte [5]; Giraa\_01, de la Universidad de Antioquia [6]; Un-2000, desarrollado en la Universidad Nacional y empleado para estudiar técnicas de aprendizaje basadas en redes neuronales [7], y P-metin, de la Universidad Tecnológica de Pereira [8].

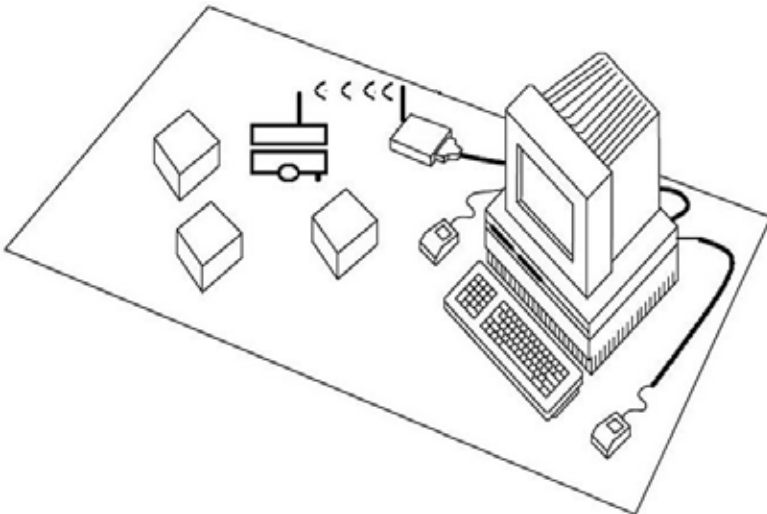
En este artículo se presenta el diseño y construcción de una plataforma robótica móvil de bajo costo y altas prestaciones, adecuada para la enseñanza y la investigación de temas referentes a la robótica móvil y sistemas robotizados en entornos estructurados estáticos y dinámicos, como, por ejemplo, el desarrollo y validación de algoritmos de navegación y control de robots.

El artículo está estructurado en la siguiente forma: en la sección 1 se presenta una descripción de las especificaciones de la plataforma; en la sección 2 se describe la estructura del *hardware*; en la sección 3 se expone la implementación del *software*; en la sección 4 se presentan los resultados de la construcción del prototipo del robot; en la sección 5 se analizan la expansibilidad y los trabajos futuros que se desarrollarán con este primer prototipo, y finalmente, en la sección 6 se presentan las conclusiones.

## 1. ESPECIFICACIONES BÁSICAS DE LA PLATAFORMA Giraa \_02

Concebida para la experimentación, la plataforma Giraa\_02 presenta características de autonomía y cuenta con varios sistemas de sensado, pues mientras unos sensores se ven afectados por el entorno, otros no presentan esta dificultad [9], así: los sensores de infrarrojo, también llamados sensores de proximidad [10], son afectados por factores como superficies translúcidas, componentes de la luz ambiente, el color de los objetos (especialmente el color negro), etc.; mientras que estos factores no interfieren con los sensores de ultrasonido, los cuales se ven afectados con la variación de la presión atmosférica y la temperatura [11]. Otro factor que afecta es la absorción de la onda ultrasónica que puede presentar objetos elaborados con materiales poco densos, tales como espuma o icopor, etc. Lo anterior denota la importancia de la combinación de sistemas de sensado de diferentes tecnologías para permitir al robot conocer su propio estado y el de su entorno [12].

Otra característica importante que ofrece el robot es su tamaño, que es adecuado para programar la plataforma desde un pc y desplazarse en una superficie tan reducida como un escritorio, tanto en operación autónoma como “teleoperada” (remota supervisada), tal como se observa en la figura 1. Además, posee una arquitectura abierta y expansible, de tal manera que a futuro permite adaptar módulos adicionales de sensores, actuadores y otros dispositivos.

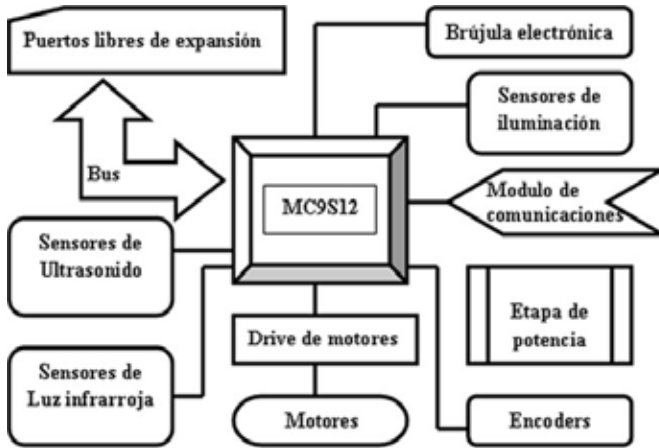


**Figura 1**  
Robot Giraa\_02 en configuración “teleoperado”

## 2. ESTRUCTURA DEL *HARDWARE*

El prototipo se diseñó de forma modular, compuesto por cinco subsistemas fundamentales:

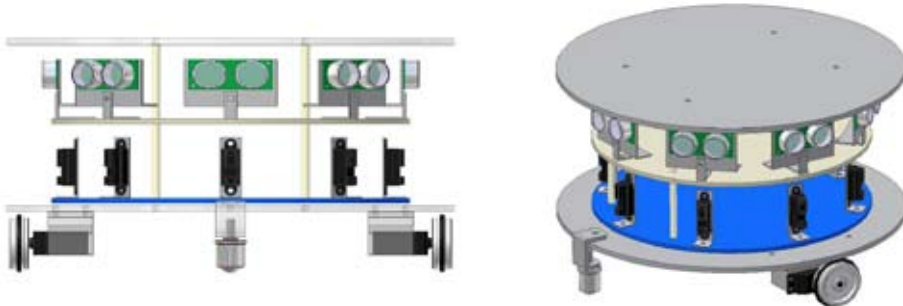
- *Subsistema de procesamiento de la información.* Basado en el microcontrolador MCU de 16 bits, el MC9S12, que se encarga de comandar todas las acciones del robot, tales como procesar la información proveniente del puerto de comunicación serial, o sensores, enviar órdenes a los motores en respuesta a la ejecución de un algoritmo de control, o por órdenes enviadas desde un PC (en configuración “teleoperado”). Este microcontrolador posee módulos especiales, e instrucciones para la ejecución de algoritmos basados en lógica difusa [13], lo que facilita el desarrollo y aplicación de técnicas de inteligencia computacional para el control de la plataforma. Es importante resaltar que el microcontrolador cuenta con un gran número de puertos libres de expansión, lo que facilita la adaptación de nuevos dispositivos en el robot.
- *Subsistema sensorico.* Está compuesto por sensores de luz infrarroja, sensores de iluminación, sensores de ultrasonido y codificadores de posición o *encoders*.
- *Subsistema de comunicación.* Permite la interacción con el robot a través de un protocolo y una interfaz RS-232 para conectarse serialmente con el PC. También está dotado de módulos de comunicación inalámbrica por RF (radio frecuencia), que permiten comunicar un computador y el robot a una velocidad máxima de 4.800 baudios.
- *Subsistema actuador.* Las ruedas del robot son accionadas, cada una, por un motor DC, acoplado mecánicamente a un tren de engranajes que permite reducir su velocidad y aumentar su torque [14]. La potencia eléctrica suministrada a los motores está controlada mediante la técnica de modulación de ancho de pulsos (PWM en inglés) [15].
- *Subsistema de potencial/alimentación.* Se diseñó para permitir que el robot operara con una autonomía de aproximadamente diez horas, en pleno funcionamiento o conectado a una fuente externa. El esquema del sistema central se observa en la figura 2.



**Figura 2**

Esquema del sistema central del robot Giraa\_02

En la figura 3 se muestra la posición de algunos componentes de la plataforma robótica móvil Giraa\_02, tal como los sensores de ultrasonido, en la parte superior, y los sensores de proximidad infrarrojos, en la parte inferior; el rodamiento esférico omnidireccional, en la parte posterior, ofrece mayor estabilidad a la plataforma; los motores y las ruedas están unidos a la estructura en un esquema de locomoción diferencial (ver figura 4).



**Figura 3**

Vistas del robot: a) vista posterior, b) vista en 3D

La plataforma robótica presenta una estructura resistente, poco peso, fácil de maniobrar y fácil acceso a conectores, suiches, sensores, motores, etc.

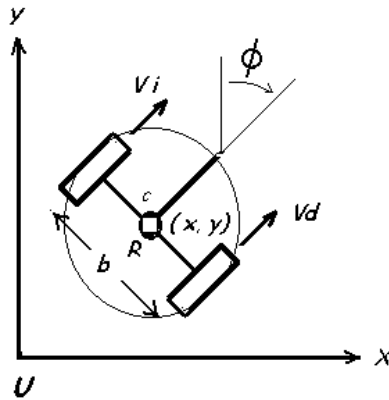


Figura 4

Esquema de locomoción diferencial empleado en el robot Giraa\_02

### 3. IMPLEMENTACION DEL SOFTWARE

El *software* básico implementado para el control de los subsistemas del robot fue programado usando el CodeWarrior, el cual es un ambiente de desarrollo enfocado al ANSI C para programar aplicaciones en sistemas embebidos [16]. Este ambiente es fácil de entender, trabaja por proyectos y está ampliamente estandarizado. Con miras a garantizar que el desarrollo del *software* sea modular y portable, se crearon librerías de funciones básicas, las cuales pueden ser utilizadas en la codificación de los algoritmos de control ejecutados en la plataforma.

Algunas librerías contienen la definición de variables y constantes necesarias en la elaboración de los programas ejecutados en la plataforma. Otras contienen primitivas o funciones básicas necesarias para interactuar con los sensores, los motores, etc. El contenido de las librerías puede ser invocado por cualquier programa que requiera el uso de los subsistemas del robot. Con el uso de librerías se pueden construir fácilmente rutinas y programas más complejos, reduciéndose el tiempo de desarrollo en la programación de algoritmos [17].

#### 3.1. Librerías de bajo nivel (dependientes del hardware)

Las librerías creadas en el desarrollo inicial del proyecto son las librerías dependientes del hardware o controladores (*drivers*). Prácticamente existe una librería por cada módulo periférico del microcontrolador. Las principales librerías creadas son las siguientes:

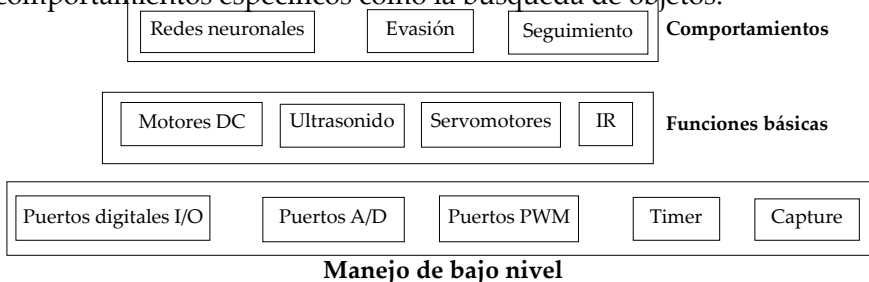
- Módulo de comunicación serial (transmisión, recepción)
- Módulo de temporizadores para referencias de tiempo, generación de interrupciones síncronas
- Módulo de temporizadores para captura y medición de eventos
- Módulo de memoria EEPROM (escritura, lectura)
- Módulo A/D para conversión de señales análogas a digitales
- Módulo de puertos digitales de entrada-salida
- Módulo PWM

### 3.2. Librerías de alto nivel (independientes del *hardware*)

Con las funciones implementadas se pueden generar otras librerías derivadas de un nivel superior de abstracción, las cuales son independientes del *hardware*. Por ejemplo, las librerías de motores llaman funciones que están definidas en otros módulos de código (puertos, PWM, *timers*) y no manejan directamente el *hardware* de la plataforma. Esto permite que, ante la eventualidad de un cambio de subsistema de procesamiento por uno más potente, se puedan reutilizar las librerías de alto nivel y sólo sea necesario desarrollar las librerías de bajo nivel que controlan los módulos periféricos. Las librerías de alto nivel creadas son las siguientes:

- Módulo de motores
- Módulo de sensores de ultrasonido
- Módulo de sensores IR
- Módulo de redes neuronales (BPN)

Cabe mencionar que se pueden desarrollar nuevas librerías a medida que se incorporen nuevos dispositivos *hardware* al robot. También está la posibilidad de optimizar las librerías actuales con la elaboración de algoritmos mejorados o comportamientos específicos como la búsqueda de objetos.



**Figura 5**

Programación del robot basada en librerías de funciones

Un ejemplo de la programación del robot se presenta en la figura 6, en la cual se muestra una sección del código de una función que simula el comportamiento reactivo de evasión de obstáculos. La función se llama *evitar*.

```
...
if(Distancia[2]<UMBRAL || Distancia[1]<UMBRAL)//obstáculo adelante
{
    direccion_time(2000,'b',80);
    direccion_time (1000, 'i',50);
    direccion('a',50);
}
else{//3
    if(Distancia[5]<UMBRAL || Distancia[6]<UMBRAL)// obstáculo atrás
    {
        direccion('a',80);
    }
    else{//2
        if(Distancia[0]<UMBRAL || Distancia[7]<UMBRAL)// obstáculo izquierda
        {
            direccion_time(LevelRot,'d',50);
            direccion('a',50);
        }
    }
    else{//1
        if(Distancia[3]<UMBRAL || Distancia[4]<UMBRAL)// obstáculo derecha
        {
            direccion_time(LevelRot,'i',50);
            direccion('a',50);
        }
    }
    else{//0
        direccion('a',50);// situación normal, velocidad crucero
    } //0
} //1
} //2
} //3
```

**Figura 6**

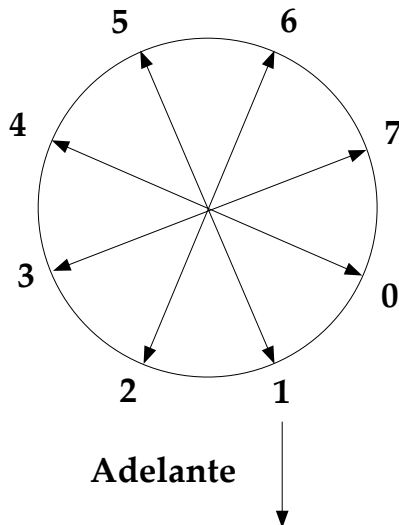
Sección del código de una función que simula la evasión de obstáculos

En el primer renglón se observa una variable denominada “*Distancia []*”; esta variable es el vector que guarda las distancias detectadas por los ocho sensores de ultrasonido. El orden de la asignación de estas distancias se indica en la figura 7.



El valor constante denominado **UMBRAL** es un valor definido en una macro (archivo \*.h), que especifica la distancia en centímetros a partir de la cual se realizará la evasión; por defecto, este valor está configurado a 5 cm, pero puede ser más bajo (mayor probabilidad de choque) o más alto (mayor probabilidad de comportamiento inestable).

Después de conocer estos detalles se puede interpretar lo que está definido en la primera línea; si la distancia en el sensor 2 es menor que el umbral o la distancia del sensor 1 es menor que el umbral, se debe realizar una determinada acción; esta situación corresponde a encontrar un obstáculo frontal. El robot intentará evitarlo rotando hacia la izquierda por un tiempo y después retornará a su desplazamiento hacia delante.



**Figura 7**

Referencia de las distancias de los sensores de ultrasonido

Se usaron dos funciones que controlan el movimiento de los motores; estas funciones se llaman *direccion\_time* y *direccion*. La primera tiene como argumentos el tiempo de movimiento en milisegundos, un carácter que indica la dirección (a = adelante, b = atrás, i = izquierda, d = derecha, s = parar), y el último argumento será la velocidad de los motores, que puede ser hasta del 100% del ciclo PWM.

En caso de que no se cumpla esta condición, se examinará algún posible obstáculo en la zona posterior del robot en los sensores 5 y 6; ante este evento de “persecución”, el robot responderá con un aumento de su velocidad del 50% a un 80%.

En caso de que el robot tenga un obstáculo por el lado izquierdo, advierten los sensores 7 y 0, y reacciona con un giro en dirección contraria, hacia la derecha. En caso de que tenga un obstáculo por el lado derecho, advierten los sensores 3 y 4, y reacciona con un giro en dirección contraria, hacia la izquierda. En el caso en el cual no se detecta ningún obstáculo, el robot tendrá una dirección hacia delante con un nivel de 50% de velocidad (velocidad crucero).

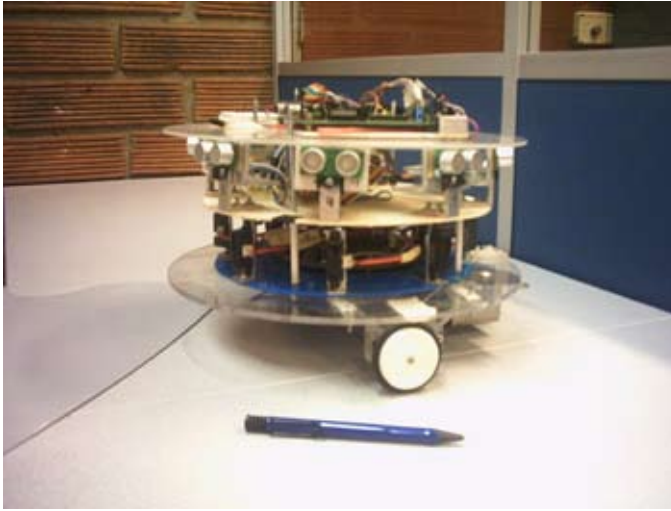
El anterior programa fue escrito con propósitos demostrativos, con el fin de ilustrar cómo se emplean varias primitivas o funciones básicas de las librerías y cómo se utilizan los sensores y los motores.

### 3.2.1. Librería de Redes Neuronales Artificiales

Esta librería tiene funciones que implementan ANNs (*Artificial Neural Network*); consiste en un perceptrón multicapa con algoritmo de aprendizaje *backpropagation*. Como el entrenamiento de la red es *offline*, es necesario entrar los pesos en los archivos de código fuente de la librería. Es posible crear hasta 60 neuronas entre todas las capas.

## 4. ELABORACION Y PRUEBA DEL PROTOTIPO

Se ha diseñado un sistema autónomo y expansible que se caracteriza por su modularidad, su fácil programación y por que consta de múltiples sensores, cuya combinación es ideal para el óptimo desempeño de cualquier robot. En la figura 8 se muestra el prototipo finalmente construido. Para efectos de probar su desempeño, se diseñó una serie de programas que permiten verificar el funcionamiento de todos los subsistemas del robot, tales como: la caracterización de los sensores, prueba de los motores, teleoperación del robot por medio de teclado desde el pc, evasión de obstáculos, seguimiento de contornos y de objetos, etc. También se han implementado algoritmos de navegación que emplean redes neuronales, y comportamientos reactivos de sentido y actuación.



**Figura 8**  
Plataforma Robótica Móvil Gira\_02

El robot posee una arquitectura *hardware* abierta y expansible, de tal manera que permite adaptar módulos adicionales de sensores (temperatura, acelerómetros, giroscopios, etc.), actuadores y otros dispositivos *hardware* como DSPs, FPGAs, uC, memorias, etc. En la tabla 1 se resumen las principales características del Gira\_02.

**Tabla 1**  
Principales características de la plataforma robótica móvil Gira\_02

<b>Características físicas</b>	
Altura	20 cm
Diámetro	30 cm
<b>Fuente de alimentación</b>	
Batería	12 V – 7 Ah
Tiempo de autonomía	10 horas (Aprox.)
Tiempo de recarga	12 horas (Aprox.)
Fuente externa Vdc	15 V
<b>Movilidad</b>	
Motores	Con reducción mecánica, 6 VDC
Ruedas	Diámetro: 4.5 cm

Rodamiento esférico	Omnidireccional
<b>Procesador</b>	
CPU	Motorola MC9S12 16 bits
Memoria	Ram: 12KB; Flash: 256 KB EEPROM: 4Kb
Entradas analógicas	16 canales (8 bits)
Módulos	8 canales de IC/OC 8 canales de PWM 2 SCI, 3 SPI
Puertos	89 I/O digitales,
<b>Sensores</b>	
Infrarrojos	8, Detector de proximidad Rango 4 –30 cm
Ultrasonido	8, medidores de distancia Rango 0.03 – 3mts
Iluminación	Fotorresistencias CdS
Encoders	Par reflexivo
<b>Comunicación inalámbrica</b>	
Transmisor	Alcance 100 metros, Frec. 315Mhz/434Mhz 4800 baudios máximo
Receptor	Frec. 315Mhz/434Mhz 4800 baudios máximo

## 5. EXPANSIBILIDAD Y TRABAJOS FUTUROS CON EL ROBOT Giraa \_02

Giraa \_02 posee puertos de expansión, con los cuales se pueden acoplar futuros desarrollos que complementen la utilidad del robot. Entre los más importantes se encuentran los siguientes:

- El diseño de un brazo mecánico.
- El acople de una pequeña cámara o retina electrónica.
- El desarrollo de un sistema de comunicación inalámbrica a larga distancia.
- El acople de un compás magnético, que detecta la orientación respecto al norte geográfico.

Todos estos módulos permitirán al robot ser más versátil, aumentar sus prestaciones e incrementar el campo de estudio para el cual fue diseñado originalmente. Las nuevas declaraciones, funciones o primitivas de *software* que se van elaborando a la par de los nuevos módulos de *hardware* se pueden agregar a las librerías actuales.

## 6. CONCLUSIONES

Giraa\_02 es un robot modular con el *hardware* y *software* básico de una plataforma para experimentación en robótica móvil, siendo una herramienta útil para prueba de algoritmos de control en ambientes reales. Dado que el MCU MC9S12 cuenta con instrucciones para lógica difusa, propicia experimentar y validar algoritmos basados en esta lógica multivaluada. Adicionalmente, el robot cuenta con una librería *software* de redes neuronales que facilita la implementación de algoritmos de navegación y control basados en técnicas de inteligencia computacional.

Cada vez se incrementa la velocidad de procesamiento y la capacidad de memoria en los microcontroladores; también los módulos *hardware* son más abundantes. Estas ventajas permiten incrementar el nivel de complejidad de las aplicaciones utilizando lenguajes de alto nivel, donde potentes compiladores generan el lenguaje de máquina y optimizan los recursos de los dispositivos sin requerir tanto esfuerzo del programador, lo que permite ocupar más tiempo en el diseño de los algoritmos y en cómo perfeccionar las aplicaciones a partir de un código bien escrito.

El concepto de modularidad con el que se diseñó la estructura de la plataforma Giraa\_02 y el *software* basado en librerías de primitivas o funciones independientes del *hardware*, permiten que el robot evolucione desde una versión muy simple de componentes hasta poseer elementos más complejos, sin tener que modificar o reescribir el código generado anteriormente. Los módulos adicionales pueden ser tan complejos como: módulos para manejo de una retina electrónica, un brazo mecánico y otros tipos de sensores. Igualmente, las librerías de *software* pueden actualizarse con nuevas funciones.

Se puede destacar que el prototipo construido presenta un diseño fiable, económico, de fácil construcción y fácilmente programable; esto lo hace adecuado para la robótica educativa y la investigación, especialmente en instituciones y grupos de investigación que cuentan con presupuesto reducido.

Los resultados obtenidos, tanto en el diseño como en la validación de algoritmos básicos de navegación, fueron altamente satisfactorios, toda vez que el robot trabajó en los diferentes modos de operación: autónomo, teleoperado, comandado por radiofrecuencia, y obedeció a las expectativas de respuesta y versatilidad propuestas al inicio del proyecto.

## RECONOCIMIENTOS

El robot móvil Giraa\_02 fue elaborado por el Grupo de Investigación en Robótica y Mecatrónica GIRAA, adscrito al GIMEL [<http://ingenieria.udea.edu.co/giraa/>], en el marco del proyecto MC02-1-09, financiado por el Comité para el Desarrollo de la Investigación (CODI) de la Universidad de Antioquia.

## REFERENCIAS

- [1] K-TEAM. *Khepera user manual*. Lausana: EPFL, 1998.
- [2] NOMADIC TECHNOLOGIES, INC. <http://www.robots.com/>. Diciembre/2005
- [3] REAL WORLD INTERFACE. <http://www.rwii.com/>. Enero/2006.
- [4] BACCA EB y PAZ EA. "Diseño e implementación de un robot móvil modular". *Energía y computación*. 2000, Vol. 9, No. 2, p. 34-39.
- [5] CUBAS C, CONSUEGRA M, LÓPEZ Ret al. "Dispositivos autómatas para navegación, detección, recolección de pelotas de tenis en ambientes conocidos". *Ingeniería y Desarrollo*. 2001, No. 9, p. 98-112.
- [6] MUÑOZ ND, TRESPALACIOS E, AEDO JE. "Giraa\_01: Una plataforma para investigación en robótica móvil. *Revista Silicio*. 2002, Vol.1, n.14, p. 26 – 31.
- [7] RAMIREZ G. "Método de aprendizaje simple para navegación de minirobots móviles rodantes". *Dyna*. 2003, Vol. 70, No. 138. p. 47-54
- [8] CÓRDOBA JP, RESTREPO C, RÍOS LE. "Descripción de una plataforma móvil implementada para el estudio de técnicas inteligentes de navegación: p-metin". *Scientia et Técnica*. 2005, Año XI, No 27, p.37-42.
- [9] JONES J, FLYNN A, SEIGER B. *Mobile Robots Inspiration to implementation*. Massachusetts: Editorial A K Peters, Ltd, 1998.
- [10] OLLERO A. *Robótica manipuladores y robots móviles*. Marcombo, Barcelona, 2001.
- [11] MCMANIS C. "Sensing obstacles with mobile robots". *Circuit Cellar*. 1996, No. 73, p. 12-16.
- [12] LONDOÑO N. La robótica: realidad y ficción. *Revista Silicio*. 2000. Vol 8, N° 11, p 41-52.
- [13] MOTOROLA, INC. HCS12 CPU V2.0 reference manual. Denver: Motorola literature, 2003.
- [14] GONZÁLEZ F. La robótica, otros medios de transmisión. *Electrónica y computadores*. 1998. No. 41, p..14-20.
- [15] ANGULO JM. *Robótica práctica*. 2 ed. Barcelona: Rede, 1985.
- [16] METROWERKS. *Codewarrior IDE user guide*. Austin: Metrowerks documentation, 2000.
- [17] SCHILDT H. C. *Manual de referencia*. Madrid: McGraw-Hill, segunda edición, 1990.