

# SIMULACIÓN Y ENSAMBLAJE DE UN PROTOTIPO PARA CONTROL Y NAVEGACIÓN DE UN ROBOT MOVIL

Lenin M. Andrango M.  
Nelson G. Sotomayor O

Rafael Fierro B., PhD.

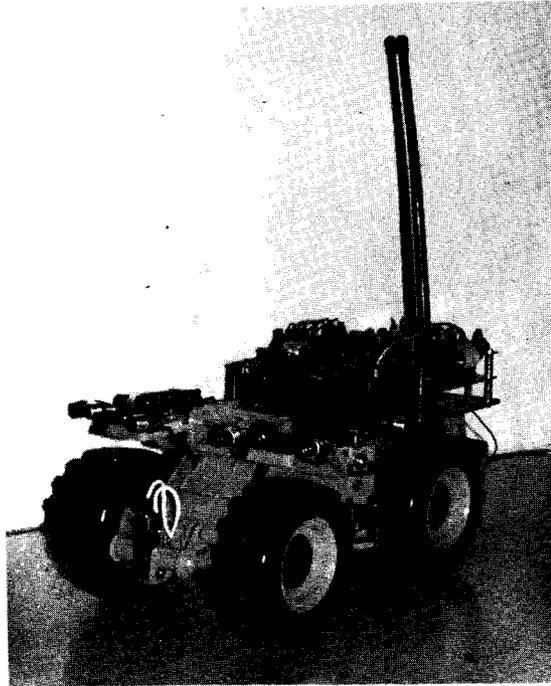


Figura 1. Plataforma Móvil *Pioneer 1*

## RESUMEN

Se desarrolló un programa en ambiente Windows 95/NT, que simula un robot móvil de tracción diferencial, en el se puede editar ambientes de trabajo del robot, el cual puede estar constituido por paredes, corredores y obstáculos que serán localizados por el usuario. Además del programa se ensambló un prototipo de robot móvil, el cual realiza tareas de navegación simples tales como marcha adelante, marcha atrás, movimientos de izquierda y derecha, tiene también la capacidad de buscar caminos alternativos de movimiento con el objeto de evitar chocar con obstáculos que se encontraran en su trayectoria dentro de su ambiente de trabajo. Se le incorporó además sensores para que tenga la capacidad de recoger información del medio ambiente que está explorando.

## ABSTRACT

A simulation program for mobile robot navigation and control has been developed under Win 95/NT operating system. The simulated differential drive mobile robot can navigate within a user defined environment filled by obstacles, walls and corridors.

A differential drive mobile platform has also been designed and built. It can perform basic navigation tasks such as: go forward, go backward, turn left, and turn right. Based on these basic tasks, more complex behaviors have been programmed and implemented on mobile robot. Thus, the robot can navigate avoiding obstacles, finding a free apace and gathering some information about its environment.

## 1. INTRODUCCION

En los últimos años la investigación sobre robots móviles ha tenido un gran desarrollo, esto se debe principalmente al desarrollo de simuladores en los cuales se pueden probar diferentes algoritmos de control antes de ser implantados en el robot real. Estas sofisticadas herramientas se han desarrollado para sistemas UNIX, siendo sus costos elevados y en cierta manera prohibitivos, ya que se encuentran en el orden de los miles de dólares.

Por lo anotado anteriormente y por la poca difusión de simuladores en un sistema operativo WINDOWS, en el presente trabajo se desarrolló un simulador con características similares a los de tipo comercial utilizando para ello el programa *Microsoft Visual C++* para Windows 95.

Además del simulador se ensambló un prototipo de robot móvil de tracción diferencial. El robot utiliza elementos de bajo costo a diferencia de las plataformas móviles comerciales cuyos costos suelen ser prohibitivos.

A la plataforma móvil se la denominó *Pionero1*, ya que constituye el primer trabajo en el campo de robótica móvil desarrollado en la Escuela Politécnica Nacional. El robot será descrito como dos partes separadas inicialmente, las cuales fueron acopladas para conseguir un funcionamiento adecuado, estas partes son: el sistema mecánico y el sistema eléctrico. El sistema mecánico se lo adaptó de un juguete, del cual se aprovechó parte de la carrocería, el sistema de tracción diferencial y los motores de DC. El sistema eléctrico está constituido por un sistema microprocesado y por el sistema de sensores, para navegación y monitoreo. Tanto el prototipo como el simulador implementado utilizan el mismo sistema de navegación, por lo que utilizando el simulador se puede de una manera muy aproximada conocer el funcionamiento del prototipo en diferentes ambientes de trabajo. Sin embargo cabe mencionar que los simuladores tienen limitaciones ya que no se pueden simular todos los tipos de sensores, y no se podría predecir un funcionamiento erróneo de los mismos, como por ejemplo malas medidas de sensores ultrasónicos debido a reflexiones [4].

## 2. SIMULADOR *PIONERO1*

El simulador está desarrollado en el lenguaje de programación orientado a objetos Visual C++ utilizando las librerías MFC (Microsoft Foundation Class) para sistemas basados en ambiente Windows. Visual C++ utiliza una interfaz gráfica con un gran número de ayudas y herramientas denominado Developer Studio.

Para que la aplicación funcione como el Simulador del Robot Móvil se toma en consideración los siguientes puntos:

- Es necesario representar un ambiente gráfico que conste de un área donde el robot pueda desplazarse y puedan colocarse obstáculos. Estos pueden estar constituidos por formas geométricas básicas. Para tal efecto se crea la clase FORMA la que contiene los datos y funciones de las formas Rectángulo y Elipse que serán los obstáculos del ambiente gráfico del simulador.
- El Robot Móvil debe tener la capacidad de moverse en el medio y evitar los obstáculos con la simulación de los sensores de ultrasonido utilizados. En este caso se utiliza la clase MOVIL que consta de una figura que simula el robot real con funciones que permitan la animación, además incluye una representación de un área gráfica que simula el alcance de los sensores. Dentro de esta clase se define el comportamiento del robot frente a un obstáculo reproduciendo el mismo algoritmo implementado en el robot real.
- Como el prototipo tiene la capacidad de control a distancia por medio de comunicación serial, se implementa un cuadro de dialogo para este fin en el que consta botones para la navegación y comandos para la lectura de sensores.
- Para ciertas características del simulador se incluyen ayudas para el manejo del mismo, en forma de cuadros de diálogo para fijar valores del simulador, como desplazamiento del robot, velocidad de animación y configuración de los sensores.

La figura 2 muestra la arquitectura funcional del software. Básicamente está conformado por una Interfaz Hombre Máquina que abarca todos los objetos gráficos con que el usuario puede interactuar como son: Barra de Herramientas, Cajas de Diálogo con información de datos que pueden ser fijados, Barras de Estado que indican características del simulador y un Menú.

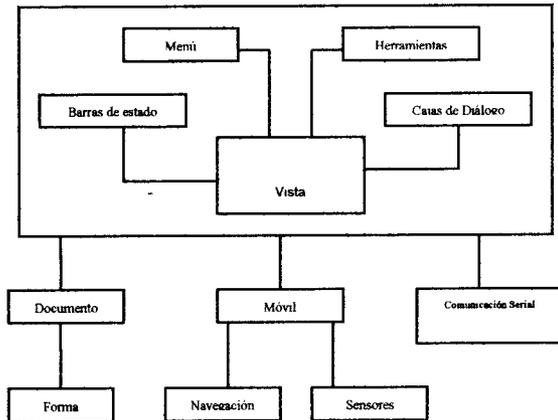


Figura 2. Arquitectura del Software

Por otro lado se encuentra el Documento que guarda la información de los obstáculos que son de la clase Forma. También el objeto Móvil se encarga del proceso de navegación y la simulación de los sensores.

Para la descripción detallada de la arquitectura del simulador se dividirá en dos grupos: Interfaz Gráfica y Objetos del Simulador.

## 2.1. Interfaz Gráfica

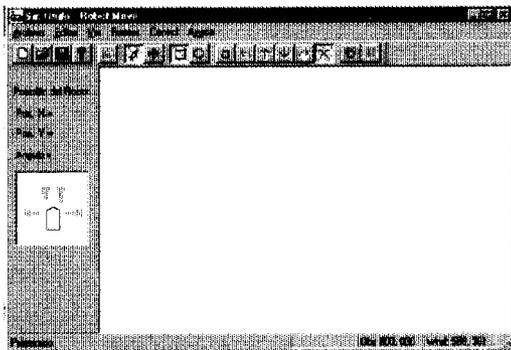


Figura 3 Interfaz Gráfica

Es una ventana que encierra el área de trabajo donde se crea un ambiente para la navegación del móvil, en ella se encuentran:

La Barra de Herramientas, que es un conjunto de botones que ejecutan una función específica:

*Funciones básicas:* crear un nuevo ambiente, abrir o guardar un ambiente e Información del Programa.

*Funciones de Edición:* borrar todo, crear formas, cambiar el tamaño y posición de las formas, escoger tipos de formas (rectángulo y elipse).

*Funciones de Navegación:* navegación manual, dirección del movimiento (izquierda, adelante, atrás, derecha, parar), navegación automática y pausa

La Barra de Estado del robot que muestra información de la posición del robot y el estado de los sensores. Además en la parte inferior se muestra la posición actual del cursor, el tamaño del último objeto (obstáculo) y el tamaño de la ventana.

Como interfaz gráfica se muestran también las cajas de diálogo que se despliegan cuando el usuario selecciona una opción del menú control de la ventana principal, estas son:

*Menú: Control – Retardo...*

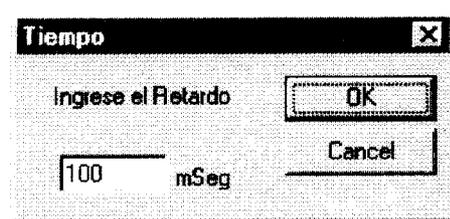


Figura 4 Control Retardo

En este Diálogo se puede establecer el tiempo de retardo para la animación. Este tiempo está comprendido entre una posición y la posición siguiente después de un movimiento. La velocidad de desplazamiento también depende de las capacidades del computador.

Menú: Control – Desplazamiento...

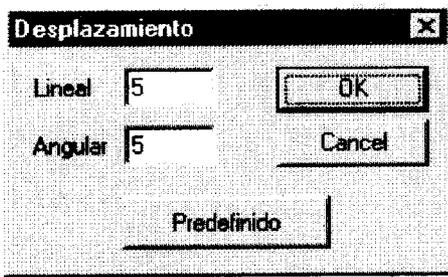


Figura 5 Control Desplazamiento

A través de este Diálogo se puede fijar el desplazamiento, en píxeles, Lineal y Angular por separado dependiendo que movimiento se desea que sea más rápido.

Menú: Control – Aleatorio...

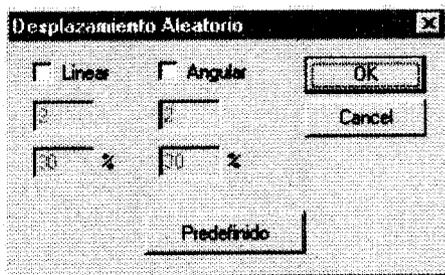


Figura 6 Control Desplazamiento Aleatorio

Como una ayuda para la simulación se puede añadir al desplazamiento del móvil cantidades de movimiento aleatorio (en píxeles), que se añaden al desplazamiento normal. La frecuencia de que ocurra este desplazamiento angular se puede valorizar entre el 0% y 100%. El desplazamiento aleatorio se puede habilitar o deshabilitar independientemente.

Menú: Control – Sensores...

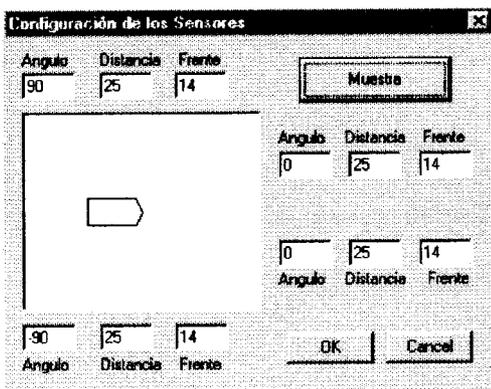


Figura 7 Configuración de Sensores

Es un cuadro de Diálogo un poco más complejo que los anteriores pero de mucha utilidad ya que puede modificar el alcance de los sensores para la simulación y ajustarse de acuerdo al robot real.

Con el menú Formas se puede escoger el modo de Crear o Editar, además de fijar el estilo de las Formas (Rectángulo, Elipse). En el menú control se encuentran todos los Cuadros de Diálogo vistos anteriormente y un cuadro adicional que es para la Comunicación Serial con el prototipo real.

## 2.2 Objetos de Simulación

### • Documento y Formas

El objeto Documento es el encargado de administrar y guardar los datos necesarios para la visualización del ambiente de trabajo. Cada Obstáculo que se crea se guarda en una lista que se incrementa a medida que se crean más obstáculos. También mantiene datos acerca del Estilo del objeto y del color. Existen además funciones que permiten el manejo y enlace de datos entre otros objetos.

La clase *Forma* consta de funciones y variables miembros que encierran las características de los objetos Obstáculo. Para dar formato a estos, guarda datos a cerca del Estilo, Color y Tamaño.

El usuario interactúa con la interfaz gráfica que está controlada por el objeto Vista (View), y ésta a su vez interactúa con los demás objetos delegando los trabajos específicos para cada uno como por ejemplo la forma de dibujar los obstáculos y la forma de guardar los datos en un archivo.

### • Móvil

Esta clase contiene las características operativas del objeto que representa al prototipo en forma de silueta que se desplaza por el área cliente del ambiente de trabajo. También contiene las funciones con el mismo algoritmo implementado en el robot real para la navegación. Además se encarga de la simulación del comportamiento de los sensores de ultrasonido implementado en el prototipo.

**EL MOVIL:** está constituido por una región definida por cinco puntos que representa la silueta del robot móvil. Existe concordancia entre las unidades reales, en centímetros, con los píxeles. Un píxel equivale a un centímetro como se muestra en la figura 8.

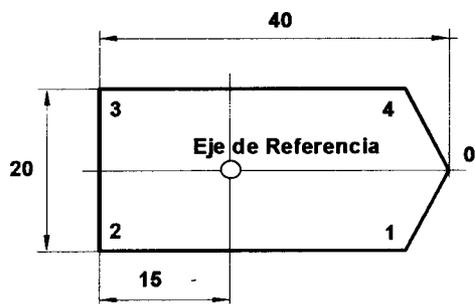


Figura 8. Estructura del Móvil

Cada uno de los puntos (0, 1, 2, 3, 4) que conforman el móvil está referido a un eje que representa el eje de rotación del prototipo cuando éste gira. Este eje sirve de referencia para el desplazamiento del móvil, por lo tanto la posición del mismo está dado por el punto del eje de referencia y la dirección con un ángulo como se muestra en la figura 9.

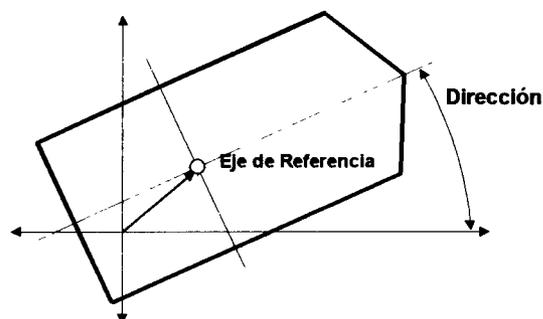


Figura 9. Desplazamiento del Móvil

Las posiciones de los puntos que conforman el móvil están guardadas en un arreglo o vector de datos.

**LOS SENSORES:** están representados por una figura triangular que muestra el alcance de los sensores de ultrasonido que están implementados en el robot real. Los sensores reales están acondicionados de tal manera que tienen la capacidad de ajuste de sensibilidad, que es la distancia de alcance de los sensores, y un ángulo de enfoque que puede ser fijado mecánicamente por tornillos.

De esta manera y por facilidad para el usuario, los datos que ingresan al sensor en el simulador están dados por tres magnitudes: distancia, ángulo y frente.

Para simular el comportamiento del sensor real se aprovecha de la diferencia de colores que existe entre el área de trabajo (por donde puede desplazarse el móvil) y los obstáculos (por donde no puede pasar). Cada vez que se realiza un desplazamiento, el programa llama a la función *LeeSensores()* que informa si alguno de los cuatro sensores ha encontrado un obstáculo haciendo una comparación del color leído en la posición del sensor con el color de fondo (área de trabajo) y si es diferente entonces señala obstáculo. Esta señalización es aprovechada por el algoritmo de control que toma una acción dependiendo que sensores están activados.

**EL CONTROL:** para el desplazamiento del móvil se considera el mismo control de navegación que en el prototipo real tiene implementado.

En cada desplazamiento del robot, se llama a la función *Control()* que es parte del objeto Móvil. Esta función es la encargada de decidir hacia donde se moverá el robot dependiendo si un sensor está activo o no.

El algoritmo de la función *Control()* lee los sensores y realiza las siguientes tareas:

- Comprueba si uno de los sensores delanteros está activo (encontró un obstáculo), si es así, entonces llama a la función *Control\_B()*.
- Comprueba si existe problema (cuando entra en un callejón), y llama a la función *Control\_C()* mientras no exista salida (no puede girar a ningún lado).

*Control\_B():* Si puede girar a cualquier lado, escoge uno alternativamente. Si el sensor izquierdo está activo gira a la derecha, si el sensor derecho está activo gira a la izquierda y si ambos sensores están activos entonces existe problema (entró en un callejón).

*Control C()*: Si el sensor izquierdo no está activo, entonces hay salida y gira a la izquierda. Si el sensor derecho no está activo hay salida y gira a la derecha. Si ambos sensores laterales están activos no hay salida y el desplazamiento es hacia atrás.

#### • Vista

El objeto Vista es el que mayor cantidad de funciones tiene ya que implementa el manejo de la Interfaz de Usuario y todos los comandos para que el programa funcione de acuerdo a nuestras necesidades.

#### • Comunicación Serial

Una de las capacidades del prototipo es que puede comunicarse con el computador para recibir comandos de navegación y enviar lectura de los sensores. Se utiliza la comunicación serial a 9600 bps a través de un cable que conecta al computador con el prototipo. Los comandos son enviados por medio de acción de botones, y se recibe datos que son mostrados a través de una Caja de Dialogo como se muestra en la figura 10.

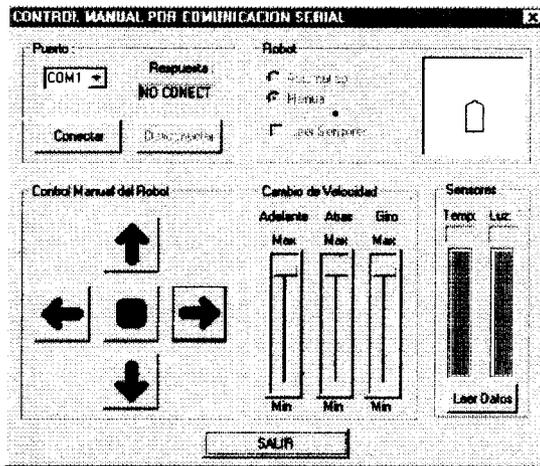


Figura 10. Dialogo para la Comunicación

La caja de dialogo se divide en áreas que agrupan ciertas características:

#### *Puerto*

Permite escoger el puerto por donde se desea hacer la comunicación, además de botones para la conexión y desconexión del mismo.

*ControlBotones* con flechas que indican la dirección a la que se desea mover el prototipo.

#### *Velocidad*

El robot real posee la capacidad de cambiar la velocidad de los motores. Cuando se envía un comando para mover el prototipo se envía también un parámetro con el que se puede variar la velocidad de desplazamiento. La caja de diálogo permite escoger una velocidad hacia delante, otra hacia atrás y otra para los giros.

#### *Sensores*

Son indicadores que muestran el nivel de temperatura y de luz de sensores incorporados en el robot real que sirven para medir parámetros del ambiente.

#### *Móvil*

Existe un control manual, donde el usuario tiene un total control de la navegación, y un control automático donde el usuario puede controlarlo pero el prototipo reacciona si encuentra un obstáculo. También puede visualizarse en la pantalla que sensores han sido activados.

### 3. PLATAFORMA MOVIL PIONERO1

La plataforma móvil se implemento en un juguete, del cual se aprovecho parte de la carrocería, el sistema de transmisión y los motores de DC. La parte inteligente de la misma se la realizó con un sistema microprocesado, el cual recibe los datos de los sensores de ultrasonido y sobre la base de esa información se comanda a los motores de DC para que navegue con facilidad en su ambiente de trabajo. Los comandos también pueden ser enviados desde un computador central para realizar un control manual del robot.

#### 3.1 Sistema Microprocesado

El sistema microprocesado esta constituido por la tarjeta de desarrollo para microcontroladores de la familia MCS-51, denominada MCPD51 [5]. En esta tarjeta aprovechando su capacidad de direccionar 64K localidades de memoria externa a través del bus de direcciones de 16 bits, se ha incluido toda la circuitería para direccionar independientemente hasta 8 dispositivos de

entrada y 8 dispositivos de salida, los dispositivos utilizados se muestran en la tabla 1.

Tabla 1 Mapa de Memoria del Microcontrolador

Direcciones	Dispositivo de Entrada	Dispositivo de Salida
0000H - 1FFFH	Pórticos SW0-SW7	Pórticos OUT0-OUT7
2000H - 3FFFH	Pórticos EXT-INT0 a EXT-INT7	Pórticos OUT8 - OUT15
4000H - 5FFFH	Sensores de Ultrasonido	Convertor D/A
6000H - 7FFFH	Convertor A/D	DISPONIBLE
8000H - 9FFFH	Memoria Eprom	DISPONIBLE
A000H - BFFFH	DISPONIBLE	DISPONIBLE
C000H - DFFFH	DISPONIBLE	DISPONIBLE
E000H - FFFFH	DISPONIBLE	DISPONIBLE

Los pórticos mencionados en la tabla, se encuentran disponibles en los conectores H2, H6 y H5 de la Tarjeta MCPD51, mayor información sobre los conectores se la encuentra en la referencia [5].

### 3.2 Sensores Utilizados

Para entregar la información del medio ambiente al microcontrolador, y realizar una buena navegación se utilizaron sensores de ultrasonido, cuya información fue acondicionada a los requerimientos del software de navegación. Además se colocaron sensores adicionales, de nivel de luz y temperatura, que se utilizarán para realizar monitoreo de estos parámetros en el medio ambiente de trabajo del robot.

Los sensores de ultrasonido utilizados para darle visión al robot, están constituidos por un par, transmisor - receptor (figura 11), que tienen las siguientes características:

Frecuencia: 40kHz  $\pm$  1.0kHz  
 Capacitancia: 2000 pF  $\pm$  20%

#### Transmisor:

Ancho de banda: 5.0 kHz/100dB  
 Nivel de presión: 112dB/40  $\pm$  1kHz  
 (0dB+0.0002  $\mu$ bar)/10V SN/30cm/min

#### Receptor:

Ancho de banda: 5.0kHz @ -75dB  
 Sensibilidad min: 67dB/40 kHz  $\pm$  1.0kHz  
 (0dB vs 1V  $\mu$ bar) R=3.9k $\Omega$

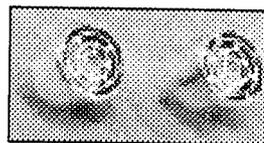


Figura 11 Sensores de ultrasonido, par transmisor - receptor

La información de los sensores proporcionada por el fabricante es muy escasa, por lo que fue necesario experimentar con ellos para obtener información de alcance, ángulo de visión y la separación óptima entre el transmisor y el receptor.

De acuerdo a estos experimentos se optó por colocar dos pares transmisor - receptor de estos sensores en la parte delantera del robot, con el objetivo de darle una visión de todo el frente del robot y evitar de esta manera posibles choques. Por la forma del montaje de los sensores, se los puede colocar a diferentes ángulos respecto al eje de las ruedas, con el objetivo de experimentar en el prototipo diferentes configuraciones hasta encontrar la óptima.

La información de los sensores de ultrasonido se la acondicionó para obtener señales TTL las cuales ingresan como memoria externa al microcontrolador, con dicha información él comanda a los motores de acuerdo al sistema de navegación que tiene programado.

Además de los sensores de ultrasonido utilizados para navegación, en el robot móvil se montaron dos sensores para monitoreo del medio ambiente, un sensor de temperatura, implementado con el LM355 y un sensor de nivel de luz, dichos sensores ingresan al microcontrolador a través del convertor A/D ADC0804, que se encuentra disponible en la tarjeta MCPD51. Como la tarjeta cuenta con una sola entrada, fue necesario usar un mux 4052, el mismo que cuenta con dos multiplexers de 4 a 1, simultáneos, de los cuales se utilizan únicamente dos entradas, quedando libres dos del primer multiplexer y las 4 del segundo, las entradas y salidas libres de este elemento se encuentran disponibles en el conector JP4 de la placa de acondicionamiento de sensores, para que en un futuro sea posible añadir más sensores a la plataforma móvil, a través del convertor A/D. Los los circuitos implementados se muestran a continuación:

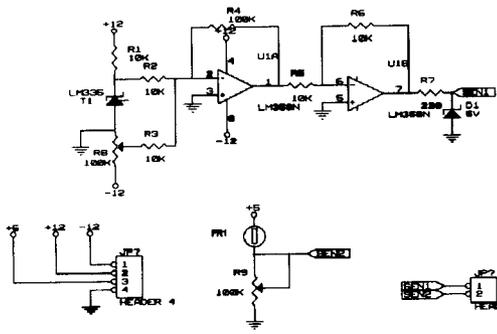


Figura 12. Sensores de Temperatura y Luz

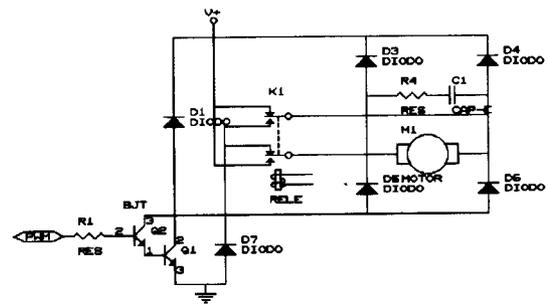


Figura 13. Control de Velocidad de Motores DC

### 3.3 Control de Motores

En el robot móvil desarrollado, era necesario realizar el control de velocidad e inversión de giro de los motores de DC. Para el control de velocidad se utilizó la configuración de un chopper reductor mostrado en la figura 13, cuyo control se lo realiza con un Modulador por Ancho de Pulso (PWM), generado por el microcontrolador, y que sale a través pin 0 del puerto 1 (P1.0). Se optó por la configuración del chopper indicado, ya que la misma señal de control PWM se la utiliza para los dos motores de DC, que deben funcionar independientemente uno del otro. La inversión de giro se la realizó con relés cuyos contactos alimentan a los motores, la conexión y desconexión de los mismos es controlada por el micro a través del pin 2 y 3 del puerto 1 (P1.2; P1.3), como se muestra en la tabla 3, ya que la inversión de giro se la realiza con corriente circulante se colocó diodos de rápida recuperación en paralelo con los contactos del relé, con el objeto de dar a la corriente un camino alternativo en el momento de la conmutación. Debido a que la bobina de los relés utilizados se excita con 24 voltios, se utilizó optoacopladores para cambiar el nivel de referencia de las bobinas, ya que para la señal de control se utiliza la fuente del micro que es de 5V.

Tabla 3. Asignación de pines del puerto 1 para control de motores

Pin	Acción
P1.0	PWM
P1.2	Señal de control M1
P1.3	Señal de control M2

### 3.4 Circuito de Vigilancia

La función del circuito de vigilancia, más conocido como Watch Dog Timer (WDT), es reinicializar el programa implementado en el microcontrolador, cuando éste por cualquier circunstancia interna o externa deja de responder adecuadamente. Este circuito informa al usuario el estado de reinicialización a través de un indicador luminoso colocado en la tarjeta de sensores.

Para este circuito se tomó como referencia un monoestable multivibrador [7] con el CI LM555, en el cual se utiliza la señal PWM generada por el microcontrolador como señal de control para el WDT, el circuito implementado y su diseño se muestran a continuación.

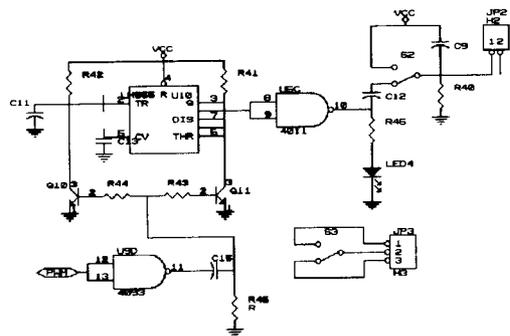


Figura 14. Circuito de Vigilancia

Para el diseño, se sabe que la frecuencia del PWM es 2KHz, por lo tanto el periodo es  $T_{PWM} = 0.5$  ms. Es necesario dar un tiempo mayor al circuito WDT para que genere el RESET, se asume que:  $T_{WDT} = 1$ ms.

Se conoce que para el LM555, el periodo es:

$$T = 1.1 RC$$

$$\text{Entonces: } R = T / 1.1 C$$

$$\text{Sea } C = 0.1 \mu\text{F, entonces}$$

$$R = 1\text{ms} / 1.1 * 0.1 \mu\text{F} = 9000 \Omega$$

Por lo que se coloca una resistencia de 10K

El usuario puede hacer que el circuito de vigilancia esté conectado al reset del microcontrolador, a través de JP2, o simplemente sea un indicador, utilizando para este efecto el selector JP3, que se encuentra conectado al conector JP4 de la tarjeta MCPD51.

## 4. RESULTADOS

### 4.1 Interfaz de Usuario

Se trata de dar al usuario facilidades en el uso de herramientas, por este motivo se utilizan una barra con botones gráficos que muestran, si se pasa con el ratón sobre ellos, la función que realizan. Además se muestra en la parte inferior (Barra de Estado) una mayor explicación de la función de los botones.

La Barra de Diálogo (a la izquierda de la ventana) es una ayuda visual que muestra la posición del móvil en las coordenadas X e Y, relativas al área de trabajo, y la dirección del frente que está dada en grados.

Otras ayudas son los datos de posición del cursor, el tamaño del último obstáculo señalado y el tamaño de la ventana.

Para fijar ciertos valores como tiempo, desplazamiento, modificación de los sensores se utilizan cuadros de Diálogo.

Todas las herramientas mostradas son sencillas de utilizar si se conoce el funcionamiento de ventanas en un ambiente Windows.

### 4.2 Editor de Ambiente

Las herramientas para crear obstáculos son básicas: rectángulos y elipses, pero con la combinación de los mismos se pueden recrear ambientes relativamente complejos. Además se

cuenta con la capacidad de guardar el ambiente de trabajo creado, con lo que se evita tener que reconstruirlo nuevamente.

Existe la capacidad de modificar el tamaño y la localización de los obstáculos a través de un recuadro llamado *tracker* que consta de 8 puntos que se encuentran en las esquinas y los puntos medios de los lados permitiendo una fácil modificación de un obstáculo. Para cambiar de posición un obstáculo solo necesita arrastrarlo al lugar deseado.

Los límites del área de trabajo son considerados obstáculos, por lo que no es necesario crear paredes que detengan al móvil cuando éste quiera sobrepasar los límites de la ventana.

### 4.3 Animación

Gracias a la capacidad de compilación que tiene Visual C++, se puede tener rutinas relativamente complejas a un costo muy pequeño en tiempo. A pesar de la forma como se dibuja el móvil y los sensores cada vez que realiza un movimiento, el programa lo muestra de una forma fluida y rápida, dependiendo también de las características del computador.

Para el comando en modo manual, cada uno de los puntos que conforman el móvil es sensible a los obstáculos, es decir, si encuentra un obstáculo choca y se detiene.

### 4.4 Comunicación con el Robot

Además de los comandos básicos que son los de navegación, se muestran los datos de los sensores de temperatura y de luz en forma numérica y también en forma gráfica.

Existen controles más avanzados que son barras deslizantes que tienen la capacidad de fijar la velocidad con que se desplaza el prototipo.

Es interesante poder observar el comportamiento de los sensores de ultrasonido reflejándose la respuesta de ellos tanto en el robot real como en la consola de comunicación.

## 4.5 Plataforma Móvil

Se cuenta con un prototipo de plataforma móvil de tracción diferencial, que puede cumplir tareas de navegación simples tales como: marcha adelante, marcha atrás y movimientos de izquierda y derecha, además tiene la capacidad de evadir obstáculos que se encuentren al alcance de sus sensores, por lo que puede navegar libremente por su ambiente de trabajo.

Ya que para su sistema microprocesado se utilizó la tarjeta MCPD51, se puede utilizar el interfaz de comunicación serial que ella posee, por lo tanto el prototipo puede comunicarse con un computador central. Con dicha comunicación se puede realizar un control directo del prototipo, es decir se puede suspender su autonomía y realizar un control manual desde un PC. Esta comunicación se la realiza utilizando un cable serial que tiene una longitud de 10 metros.

Se montó en el prototipo sensores de temperatura y nivel de luz, con los que se puede conseguir información del ambiente de trabajo de la robot móvil. Dichos sensores ingresan al sistema a través del conector JP4 de la tarjeta de acondicionamiento de sensores, quedando libres en el mismo conector dos entradas en las que se podrían colocar dos sensores más para realizar tareas de monitoreo del medio ambiente.

El prototipo de plataforma móvil utiliza para su alimentación dos baterías recargables, una de 7,5 V utilizada para el sistema de control y una de 6 Voltios que alimenta a los motores. Ambas baterías pueden ser recargadas sin ser desmontadas de la plataforma, a través de los conectores R1 y R2 respectivamente, dichos conectores se encuentran montados en las partes laterales de la plataforma.

## 5. CONCLUSIONES

El trabajo realizado abarca mucho más de los objetivos planteados, es así que ahora se cuenta con un prototipo, al cual se le puede añadir más dispositivos para navegación y monitoreo, sin hacer modificaciones en el mismo, ya que la tarjeta MCPD51 utilizada, tiene dividido el mapa de memoria externa en dieciséis espacios de memoria, para colocar hasta ocho dispositivos de

entrada y ocho dispositivos de salida, que se tratarán como memoria externa del microcontrolador.

Cuando se trabaja con motores de DC pequeños, como los de los juguetes hay que tener mucho cuidado con el ruido que este tipo de motores introducen al sistema.

Existen algunas ayudas para la simulación, como la variación en el tiempo entre cada movimiento del móvil. Esto permite variar la velocidad de la animación dependiendo de las características del microprocesador y el computador en el que se ejecute el programa. Una capacidad adicional comprende el uso de un factor aleatorio en el movimiento del móvil que simula deslizamientos de las llantas y variaciones bruscas de giro en el robot real. Existe también un cuadro de dialogo que permite simular el ajuste y calibración de los sensores, que es de gran ayuda para el estudio del comportamiento de los mismos.

Existe cierta limitación para modelar los sensores reales de ultrasonido en el simulador, por ejemplo las reflexiones múltiples o no reflexiones que requerirían un algoritmo complejo. Otro tipo de limitación es que no se puede modelar exactamente el ambiente de operación, como son el tipo de material, textura de los objetos, etc.

La herramienta para la creación de ambientes de trabajo son simples, pero pueden crear ambientes relativamente complejos.

El campo de la Robótica Móvil, en nuestro país debería ser desarrollado y adaptado a las necesidades de nuestra industria, para mejorar su rendimiento y abaratar los costos de producción. Pero no sólo en la industria puede ser utilizado, es así que este tipo de robots podrían utilizarse como exploradores en lugares nocivos para los humanos, o en lugares a los que no se tiene un libre acceso como por ejemplo el interior de tuberías, en las cuales el robot móvil podría monitorearlas, detectar fallas y realizar arreglos de las mismas.

La realización de herramientas de simulación que tengan el mismo sistema de control de un robot móvil ayuda para detectar fallas y realizar correcciones del sistema de control, ya que ésta

herramienta permite simular muchos ambientes y verificar el funcionamiento del robot en poco tiempo y a un costo menor comparado con el mismo estudio en el robot real.

## 6. BIBLIOGRAFIA

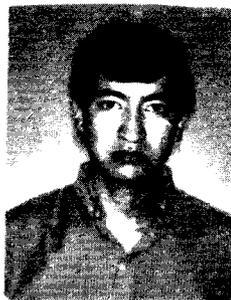
- [1] J. Borenstein, N. Everett and L. Feng, *Navigating Mobile Robots System and Techniques*, Wellestey, MA: AK Peters, 1996
- [2] P. J. McKerrow, *Introduction to Robotics*, Sidney: Addison-Wesley Publishing Company, 1993.
- [3] Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, "A Stable Tracking Control Method for an Autonomous Mobile Robot," *Proc. IEEE Int. Conf. on Robot. and Automat.*, vol. 1, May 1990, pp. 384-389.
- [4] Y. F. Zheng, *Recent Trends in Mobile Robots*, Singapore: World Scientific Publishing Co. Pte. Ltd., 1993.
- [5] B. Ledesma, *Tarjeta MCPD51*: E.P.N. Quito, Septiembre 1992.
- [6] J. Hermann, S. Eduard and L. Rolf, *Tablas para la Industria Metalúrgica*, Barcelona: Reverté, S.A., 1984.
- [7] J. Carr, "Desinging with Digital IC's," *Proc. Radio Electronics*, vol. 57, Jan. 1986, pp. 71-99.
- [8] Ediciones Nueva Lente, "Montaje de un Sistema Emisor - Receptor de Ultrasonidos," *Proc. Enciclopedia Práctica ELECTRONICA*, vol 1, 1982, pp. 208-216.
- [9] M. Rashid, *Electrónica de Potencia*, Mexico: Prentice Hall Hispanoamericana, S.A., 1993.
- [10] P. Rivera, *Control de Máquinas Eléctricas*, E.P.N. Quito, 1997.
- [11] J. González, *Introducción a los Microcontroladores*, España: McGraw-Hill / Interamericana de España, S.A., 1992
- [12] B. Kernigham and D. Ritchie, *El Lenguaje de Programación C*, México, D.F.: Prentice-Hall Hispanoamericana, S.A., 1992.
- [13] H. Schildt, *Aplique Turbo C++ para Windows*, España: McGraw-Hill Interamericana S.A., 1993.
- [14] M. Andrews, *Aprenda Visual C++ Ya*, España: McGraw-Hill Interamericana S.A., 1996.
- [15] Manual de Usuario, *Microsoft Visual C++ Versión 4.0*, Microsoft Corporation, 1995

## BIOGRAFÍAS



**LENIN ANDRANGO.**

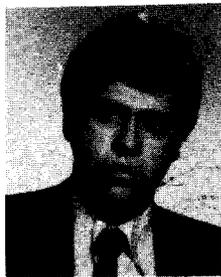
Nació en Quito el 8 de Mayo de 1971. Obtuvo el Título de Bachiller Técnico Industrial especialidad electricidad en el Colegio Técnico Don Bosco. Culminó sus estudios superiores en la Facultad de Ingeniería Eléctrica de la Escuela Politécnica Nacional en 1999. Actualmente está encargado del área de mantenimiento en la empresa FADERAZA, sus principales áreas de interés son: robótica, automatización, instrumentación virtual y programación.



**NELSON SOTOMAYOR.**

Nació en Quito el 9 de Septiembre de 1971. Realizó sus estudios secundarios en el Instituto Nacional Mejía. Se graduó en la Escuela Politécnica Nacional como Ingeniero en Electrónica y Control en 1999. Actualmente desempeña el cargo de Profesor Asistente en el Departamento de Automatización y Control Industrial de la Facultad de Ingeniería Eléctrica en la Escuela Politécnica Nacional. Además colabora con el Proyecto de Investigación PBID-085 en calidad de Investigador Asistente.

Áreas de interés: automatización, robótica móvil, microcontroladores, informática y redes.



**RAFAEL FIERRO.**

Nació en Riobamba el 19 de Enero de 1963. Estudios: Ing. en Electrónica y Control, EPN, 1987. MSc. In Control Engineering, University of

Bradford, Inglaterra, 1990. Ph.D, in Electrical Engineering, University of Texas at Arlington, USA, 1997. Becas: Consejo Británico, Comisión Fulbright, ONU. Cargos: Profesor FIE-EPN, 1987-1999, Research Assistant, The Automation and Robotics Research Institute, UTA, USA, 1993-1997. Actualmente Post-doctoral Researcher, University of Pennsylvania, GRASP Lab.

Áreas de interés: Sistemas híbridos, robótica, control.