

## RTT: ROBOT PARA NAVEGACIÓN EN TERRENOS IRREGULARES

### RESUMEN

Se describe el diseño y construcción de un prototipo de robot móvil de navegación en terrenos irregulares, concebido para experimentar con plataformas robóticas con capacidad de desplazamiento en terrenos para agricultura. El proyecto comprende: un sistema mecánico con diseño de suspensión tipo Rocker Boggie[1] (seis motores para la tracción del vehículo y cuatro servomotores para la dirección ubicados en las ruedas traseras y delanteras[2]), un sistema de potencia alimentado por dos baterías, un sistema de comunicación y control de movimiento (red de microcontroladores PIC serie 18[6]) y el software de teleoperación desarrollado en LabView[3].

**PALABRAS CLAVES:** Robótica móvil, Microcontroladores, Terrenos irregulares, Suspensión rocker boggie

### ABSTRACT

*Concepts related with design and construction of a mobile robots prototype of navigation in irregular terrain, including its physical and computational structure are present. The project includes: mechanical system with suspension design type Rocker Boggie [1] (six motors for the traction and four actuators for direction [2]), system of power, system of communication and control (microcontroller network, PIC 18xx series [6]) and teleoperation software, developed in Lab View [3].*

**KEYWORDS:** Mobile robotics, Microcontrollers, Irregular terrain, rocker boggie Suspension.

### MARIBEL ARROYAVE

Ing. en instrumentación y control  
Politécnico Colombiano  
Jaime Isaza Cadavid.  
mary\_argi@hotmail.com

### DIANA LUCÍA MAZO

Ing. electrónica  
Universidad Cooperativa de  
Colombia – Medellín  
dianalmazo@gmail.com

### JUAN BERNARDO CANO

Est. ingeniería electrónica  
Universidad de Antioquia  
jbcano@gmail.com

### MAURICIO ARIAS

Ing. en instrumentación y control  
Politécnico Colombiano  
Jaime Isaza Cadavid.  
marias09@une.net.co

### JAIRO MIGUEL VERGARA

Ingeniero Electrónico  
Profesor facultad de ingeniería  
Politécnico Colombiano  
Jaime Isaza Cadavid  
jmvergara@elpoili.edu.co

### NELSON LONDOÑO OSPINA

Ingeniero Electrónico,  
Profesor Depto. Ing. Eléctrica  
Universidad de Antioquia  
nlondono@udea.edu.co

### NELSON DAVID MUÑOZ C.

Ingeniero Electrónico,  
Profesor Depto. Ing. de Sistemas  
Universidad de Antioquia  
nmunoz@udea.edu.co

### CARLOS IGNACIO SIERRA

Ingeniero Mecánico  
Universidad de Antioquia  
gira@udea.edu.co

## 1. INTRODUCCIÓN

Durante las dos últimas décadas, se ha intensificado la investigación en la utilización de robots para la agricultura. La investigación ha estado dirigida a robots para inspección de plantas, transporte, injertos y especialmente cosecha de frutas y vegetales en horticultura [4].

Los robots móviles para agricultura, deben hacer frente a las difíciles condiciones de navegación que presentan los terrenos cultivados, donde se pueden hacer muy pocas suposiciones (al planificar la navegación del robot), debido a la naturaleza del accidentado terreno y a los obstáculos, principalmente plantas[5].

## 2. CARACTERISTICAS GENERALES DEL RTT

El RTT cuenta con una plataforma mecánica cuidadosamente diseñada que permite un correcto desplazamiento en terrenos irregulares[1], un sistema de potencia alimentado por dos baterías, un regulador de voltaje para un adecuado suministro de potencia a los motores de dirección y tracción, un sistema de sensado de corriente para evaluar el consumo total del sistema y el consumo de cada uno de los motores, una red de microcontroladores encargada del manejo de las señales entre el PC y la plataforma móvil y, por último, un software de alto nivel que permite monitorear y manipular el RTT[3].

El diseño actual propicia una fácil implementación de otros subsistemas como por ejemplo: comunicación inalámbrica, sensores que faciliten cierto grado de autonomía y control, sistemas GPS, etc...

El artículo se centra fundamentalmente en el sistema hardware y software encargado del control y comunicación Robot-PC. Se incluye: En el apartado uno, una descripción de la red de microcontroladores; en el segundo se describe la implementación de la comunicación entre la plataforma y el PC mediante un emulador USB; el apartado tres, el software de interfaz de usuario; en el cuarto apartado se presentan los resultados de evaluación y desempeño, en el cinco se analizan la expansibilidad y los trabajos futuros y finalmente, las conclusiones.



Figura 1. Robot Todo Terreno RTT

## 3. RED DE MICROCONTROLADORES

La arquitectura hardware del sistema está soportada en una red de microcontroladores, como se ilustra en la figura 2.

La arquitectura propuesta pretende optimizar las prestaciones de los microcontroladores de acuerdo a sus

capacidades y limitaciones, de tal forma que el flujo de datos entre ellos sea mínimo [6] [7] [8].

La configuración maestro-esclavo propicia una mejor utilización de los recursos de la red, así:

- *Microcontrolador Maestro*: Gestiona los datos provenientes de la interfaz de usuario (monitor/control manual), procesa la información sensorica de motores izquierdos y genera acciones de control a los motores correspondientes (PWM).
- *Microcontrolador Esclavo 0*: Gestiona la sensorica y actuación de los motores derechos
- *Microcontrolador Esclavo 1*: Realiza el control de los motores de dirección

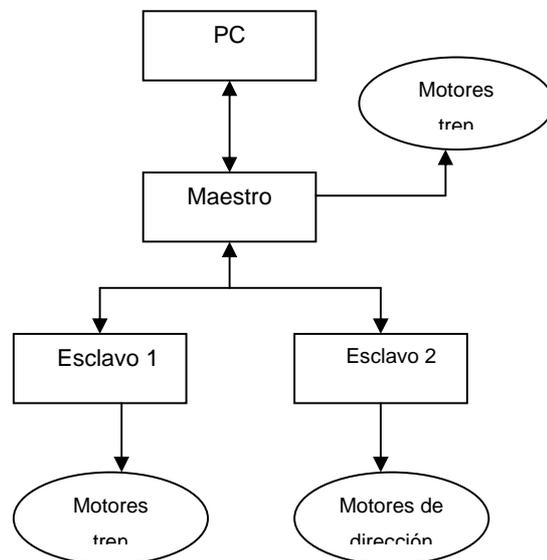


Figura 2. Arquitectura de la red de microcontroladores

### Protocolo de comunicaciones.

La comunicación entre el Sistema de monitoreo y el microcontrolador Maestro se realiza por medio de un bus RS232 a una tasa de 19200bps, mientras que la comunicación interna es una red SPI a una velocidad de 2.5Mhz, lo que garantiza un intercambio rápido de información entre los microcontroladores.

**Comunicación entre el microcontrolador Maestro y Sistema externo:** Se realiza por medio de paquetes, puesto que se decidió implementar una comunicación tipo half duplex, es decir, no se utilizará la comunicación bidireccional del RS232, con el fin de garantizar compatibilidad con transmisores RF.

Los paquetes de datos se conforman de la siguiente forma: un byte de inicio de paquete, una trama de datos de ancho fijo y un byte de finalización. Luego de enviados los datos, el receptor contesta retornando un

byte de confirmación, el cual sirve para informar de errores en la transmisión o cambios en la dirección de la comunicación (modo recepción/modo transmisión). Este protocolo permite implementar fácilmente detección de errores como paridad o suma de verificación.

Visto desde el microcontrolador Maestro, se cuenta con dos modos del bus RS232, modo transmisión y modo recepción. La operación inicia en modo recepción de datos en espera de un comando para iniciar operación, luego de una recepción exitosa, se pasa al modo transmisión donde, la red de microcontroladores envía la información de estado y permanece en este modo hasta que el sistema Interfaz de Usuario solicita cambio de modo/Estado (byte de confirmación) con lo que se conmuta de transmisión a recepción, de acuerdo a los requerimientos del Usuario/Sistema.

El tamaño de los paquetes de transmisión y de recepción, se definen en el microcontrolador maestro mediante la modificación de constantes actualizables fácilmente, las cuales, pueden ser cambiadas para variar el tamaño de los buffer de recepción y transmisión de datos del microcontrolador [6] [7] [8].

La tabla 1, ilustra las características fundamentales de los paquetes de datos.

:Dato de inicio de paquete	(250)
Paquete de datos	Datos para el maestro
	Datos para el esclavo 0
	Datos para el esclavo 1
Dato de finalización	(251)

Tabla 1. Paquete de datos recibidos

:Dato de inicio de paquete	(250)
Paquete de datos	Datos desde el maestro
	Datos desde el esclavo 0
	Datos desde el esclavo 1
Dato de finalización	(251)

Tabla 2. Paquete de datos enviados

**Comunicación entre microcontroladores.** Como se comentó anteriormente, la red de microcontroladores se implementó bajo la estructura maestro-esclavos, donde el maestro maneja la temporización e inicia y termina todas las transacciones. Al igual que la comunicación anterior, se utiliza una estructura de paquetes con la diferencia de que no se utiliza byte de confirmación, dado que es una comunicación entre integrados que tiene una baja probabilidad de error. También, la longitud de los paquetes se controla por medio de constantes que indican cuantos datos deben ser enviados y recibidos desde y hacia cada microcontrolador. Los datos a ser enviados se almacenan en el maestro en un buffer en el cual, de acuerdo a su posición, el microcontrolador discrimina a

cual debe enviarse los datos, igualmente se destinó un buffer de recepción donde se almacenan los datos recibidos de los esclavos.

Cuando el maestro inicia una transmisión, direcciona los datos del buffer correspondiente al primer esclavo y comienza a transmitir el paquete de datos hasta que envía el dato de fin de paquete; luego se direcciona el segundo esclavo y comienza el proceso de envío correspondiente.

La recepción se realiza por medio de un proceso similar, en este caso los datos se llevan a un buffer de recepción ordenados según el microcontrolador de procedencia.

#### 4. COMUNICACIÓN ENTRE LA PLATAFORMA Y EL PC MEDIANTE UN EMULADOR USB

Tradicionalmente la solución para enlazar un sistema embebido con un PC ha sido la comunicación por medio del estándar RS232, sin embargo, con el auge de la tecnología se hace necesaria una alternativa que permita migrar rápidamente hacia otros estándares más modernos como lo es el puerto USB, provocando un mínimo impacto en la programación o el hardware existente.

Se utilizó por lo tanto una solución suministrada por la MICROCHIP consistente en un conjunto de librerías y drivers para el microcontrolador 18F4455 denominada CDC EMULATION, las cuales permiten utilizar el puerto USB emulado sobre el puerto serial convencional, así:

- Desde el punto de vista del PC, al conectar el dispositivo se instala un puerto serial que se puede manejar de la manera convencional como se manejaban estos dispositivos excepto que los parámetros como velocidad y paridad son manejados por el driver USB,
  - Desde el punto de vista del microcontrolador, se suministran unas rutinas que permiten un manejo similar a un módulo de recepción serial asíncrona (USART). De esta forma se tiene una transición fácil a USB aunque no se obtienen todas las ventajas en cuanto a velocidad que permite el manejo directo del puerto USB.
- En otras palabras, en la plataforma el USB se comporta como un convertidor USB/RS232.

#### 5. SOFTWARE DE INTERFAZ DE USUARIO

La interfaz de usuario fue desarrollada bajo la plataforma software de control y diseño gráfico LabView [3]. La figura 3 ilustra la interfaz desarrollada, en la cual se muestra la estructura mecánica del robot, con dos trenes de movimiento: Tren izquierdo formado por los motores de tracción 1,2 y 3 y el tren derecho, formado por los motores de tracción 4, 5 y 6. En la parte superior de los motores 1, 3, 4 y 6, se encuentran ubicados los motores de dirección.

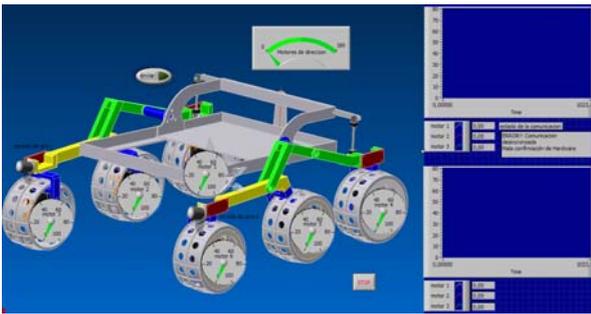


Figura 3. Interfaz gráfica en Labview.

### Etapas del programa

El programa comienza con un protocolo de inicialización donde se transmite un dato desde el PC (Se envía un 55). Para comprobar la sincronización, el microcontrolador envía la confirmación de la recepción del dato para dar inicio al programa (se espera a recibir un 56). Después de esto, el PC queda en estado de recepción.

#### 5.1 Módulo de recepción

El programa en estado de recepción recibe 25 datos, clasificados de la siguiente manera:

- Dato 0: Inicio de recepción (250).
- Dato1 - Dato6: Datos del período del encoder de los motores 1, 2 y 3 (tren izquierdo). Cada dato se recibe en dos bytes, uno con la parte baja del dato y otro con la parte alta.
- Dato7 - Dato11: datos del consumo de corriente tanto de los motores de tracción (1, 2 y 3), como de los motores de dirección del tren izquierdo.
- Dato12 – 17: Datos del período del encoder de los motores 4, 5 y 6 (tren derecho). Cada dato se recibe en dos bytes, uno con la parte baja del dato y otro con la parte alta.
- Dato 18 – 22: Datos del consumo de corriente, tanto de los motores de tracción (4, 5 y 6), como de los motores de dirección del tren derecho.
- Dato 23: Señal de estado de baterías.
- Dato 24: Fin de recepción (251).

La información del movimiento y corriente de los motores, se convierte a velocidad y valores en amperios respectivamente, con el fin de ser visualizada en gráficos.

Dentro del módulo de recepción se pregunta si se continúa en estado de recepción o se pasa al modo de transmisión. Se pasa al estado de transmisión, enviando

un dato 253 al microcontrolador para que este comience a recibir los datos del PC y permanece en estado de recepción enviando un 252 al microcontrolador.

#### 5.2 Módulo de transmisión

El programa en estado de transmisión, envía 11 datos clasificados de la siguiente forma:

- Dato 0: Inicio de recepción (250).
- Dato1 – Dato3: Datos de PWM de los 3 motores de tracción del tren izquierdo, se envía la señal en porcentaje.
- Dato4: Sentido de giro de los motores de tracción del tren izquierdo.
- Dato5 – Dato7: Datos de PWM de los 3 motores de tracción del tren derecho, se envía la señal en porcentaje.
- Dato8: Sentido de giro de los motores de tracción del tren derecho.
- Dato9: Grados de giro de los motores de dirección.
- Dato 10: Fin de recepción (251) y vuelve al estado de transmisión.

Puede ocurrir que el último dato que se envía sea un 252 en lugar de un 251, lo que significa que el PC continuará enviando y aún no se debe habilitar la recepción. Esto se hace por medio del botón de envío o a través de la tecla "Enter".

Después de transmitir todos los datos al microcontrolador, el PC espera recibir un 252 del microcontrolador con el fin de verificar que el paquete transmitido llegó correctamente, en caso de no recibir este dato, el PC muestra un mensaje avisando que hay error en la comunicación.

## 6. RESULTADOS DE EVALUACIÓN Y DESEMPEÑO

### 6.1. Las comunicaciones:

En sus diversas etapas han demostrado ser confiables en la medida que no se dan bloqueos ni fallas en los paquetes transmitidos y recibidos, además el flujo de datos es correcto, es decir, cada dato llega correctamente a su destino. El diseño proporciona flexibilidad para la implementación de futuros módulos.

Los periféricos del microcontrolador funcionan correctamente lo cual se comprueba al realizar mediciones de señales y observarlas en el osciloscopio.



Figura 4. RTT en terrenos irregulares

### 6.2. Desempeño mecánico

Para evaluar su desempeño en el desplazamiento, el RTT se probó en diferentes terrenos irregulares (arenosos, rocosos, entre otros), demostrando un buen desempeño en su estructura mecánica puesto que mantiene una correcta nivelación del chasis protegiendo las tarjetas electrónicas y evitando desconexiones en el cableado.

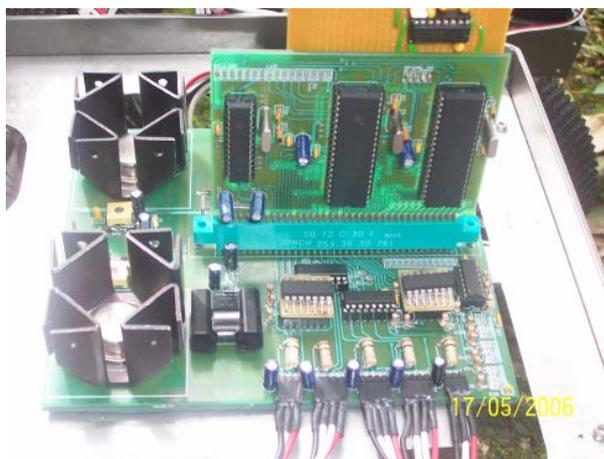


Figura 5. RTT Sistema electrónico

### 6.3. Sistema de potencia

El sistema de potencia presentó un correcto funcionamiento y cumple con los requerimientos necesarios para un buen funcionamiento de los actuadores. En lugares con inclinaciones por el orden de los 45°, el sistema de potencia suministró la regulación necesaria para desplazarse.

### 6.4. Componentes electrónicos

Cada uno de los componentes electrónicos mantiene una temperatura estable que le permite funcionar sin ningún riesgo.

### 6.5. Arquitectura Hard/Sofá

La configuración maestro-esclavo, presentó un correcto funcionamiento del sistema y propició, sin conflictos, la comunicación Usuario/Robot y Microprocesadores maestro/esclavo con excelentes resultados.

Igualmente la interfaz desarrollada permite al usuario controlar el robot de manera fácil y confiable.

En la siguiente dirección se pueden descargar videos donde se aprecia el desempeño del vehículo en diferentes entornos:

[http://ingenieria.udea.edu.co/giraa/robot\\_TT/videos/](http://ingenieria.udea.edu.co/giraa/robot_TT/videos/)

### 7. EXPANSIBILIDAD Y TRABAJOS FUTUROS

La evaluación de tiempos y recursos utilizados en la comunicación utilizada en la arquitectura empleada garantizan la posibilidad de implementar sistemas de control y navegación acordes con las necesidades del sistema y tareas concebidas.

La arquitectura es suficientemente capaz de soportar ampliaciones tanto en algoritmos de control como en dotación de sensorica y/o actuadores que permitan la implementación de mapas y estrategias de navegación mas elaboradas.

El Hardware y Software del RTT se diseño con la posibilidad de implementar comunicaciones inalámbricas, necesarias dadas las irregularidades y la extensión de los terrenos en que deberá desempeñarse.

La capacidad de los micros es suficiente para soportar ampliaciones. El esclavo 1 esta en capacidad de realizar el control de los sensores que se implementen para navegación.

### 8. CONCLUSIONES

La arquitectura propuesta en configuración Maestro-Esclavo, hace más eficiente el proceso de gestión de sensores y actuadores y facilita desarrollar un sistema de control de los dispositivos utilizados en una plataforma móvil concebida para operar en terrenos irregulares que, por su característica, exige un buen desempeño del sistema de procesamiento a bajo nivel.

El prototipo RTT desarrollado, facilita futuras ampliaciones tanto en Hardware como en Software garantizando extender su desarrollo y campos de aplicación a labores de agricultura que requieren implementar sistemas de visión, identificación y movilidad muy especializados.

Otro de los propósitos previstos en el la concepción del proyecto, es que sirviera como plataforma de investigación y aplicaciones educativas, objetivo que se

cumplió, pues permite proponer proyectos de investigación y actualización en un amplio campo de aplicaciones en: Mecánica, Electrónica, energética, robótica, mecatrónica, etc.

Con la elaboración de proyectos como estos se logra no sólo resaltar el potencial de los grupos de investigación en las universidades sino sensibilizar a los estudiantes de lo importante de realizar proyectos que miren necesidades específicas en nuestro entorno, con el fin de generar soluciones que permitan un desarrollo sostenible.

## 9. RECONOCIMIENTOS

Este trabajo fue parcialmente financiado por el convenio de investigación Politécnico Colombiano Jaime Isaza Cadavid - Universidad de Antioquia, en el marco del proyecto CNV POLI-UDEA03-1-02 desarrollado por el grupo de investigación en Robótica y Mecatrónica – adscrito al GIMEL, [<http://ingenieria.udea.edu.co/giraa/>], y el grupo GRIBOT.

## 10. BIBLIOGRAFIA

[1] M. THIANWIBOON, V. SANGVERAPHUNSIRI AND R. CHANCHAROE. Rocker-Bogie Suspension Performance. Robotics and Automations Laboratory, Chulalongkorn University, Bangkok, Thailand 10330. 2001.

[2] CUENCA, Eugenio Martín. ASENSIO, Luis Javier. Servomotores y su control, PC Práctico, Robótica bioinspirada. 2000.

[3] LÁZARO, Antonio Manuel. DEL RIO FERNANDEZ, Joaquín. LabView 7.1 Programación Gráfica para el control de instrumentación. Editorial Thomson, 2005.

[4] GARCÍA, P. Lía. Navegación autónoma de robots en agricultura: un modelo de agentes. Tesis doctoral. CSIC. España. 2003.

[5] MANDOW A, y otros. The autonomous mobile robot aurora for greenhouse operation. IEEE Robotics and Automation Magazine, Vol 3 N° 4, pp. 18-28. 1996.

[6] MICROCHIP. PICmicro 18C MCU Family Reference manual. 2000.

[7] MICROCHIP. Configuration Settings Addendum. 2005.

[8] MICROCHIP. IDE User's Guide. 2006.