

Desarrollo del robot móvil de arquitectura abierta - RoMAA -

Gonzalo F. Perez Paina

Centro de Investigación en Informática para la Ingeniería, CIII
Universidad Tecnológica Nacional, Regional Córdoba. Córdoba, Argentina

16 de abril del 2014

Resumen

El diseño y desarrollo del Robot Móvil de Arquitectura Abierta –RoMAA– surge ante la necesidad de disponer un vehículo adecuado para el ámbito de la investigación, capaz de adaptarse a distintos experimentos en las áreas de robótica móvil y visión por computadoras. El presente trabajo describe en detalle la experiencia en su desarrollo partiendo desde la idea original, la versión prototipo y la versión final –RoMAA-II–. Se describen las diferentes partes que componen el robot, incluyendo la electromecánica, la electrónica de a bordo y el software de la PC de control de alto nivel, desde el punto de vista del diseño y construcción; y se presentan sus características más relevantes. En la actualidad el CIII cuenta con cuatro robot RoMAA-II completamente operativos, y se utilizan regularmente en experimentos de diferentes proyectos de investigación.

Índice

| | |
|---|-----------|
| 1. Introducción | 2 |
| 1.1. Objetivos | 2 |
| 1.2. Producción académica | 3 |
| 1.3. Presentaciones y transferencias | 4 |
| 2. Descripción general de la plataforma | 4 |
| 2.1. Estructura mecánica | 5 |
| 2.2. Electrónica de a bordo | 6 |
| 3. RoMAA | 6 |
| 3.1. Componentes electromecánicos y bus de alimentación | 7 |
| 3.2. Controlador diferencial | 7 |
| 3.3. Conclusiones | 8 |
| 4. RoMAA-II | 9 |
| 4.1. Modificaciones respecto al prototipo | 9 |
| 4.2. Descripción general | 10 |
| 4.3. Características técnicas | 10 |
| 4.4. Electrónica de a bordo | 11 |
| 4.4.1. Comunicación | 11 |
| 4.4.2. Lazo de control de velocidad | 12 |
| 4.4.3. Cálculo de odometría | 12 |
| 4.5. Software de la PC de a bordo | 14 |
| 4.6. Sensores de ultrasonido | 14 |
| 5. Conclusiones | 16 |

1. Introducción

El diseño y desarrollo del Robot Móvil de Arquitectura Abierta (RoMAA) comienza como proyecto interno del Centro de Investigación en Informática para la Ingeniería, CIII¹ en el año 2008, concluyendo en 2009 con la fabricación del primer prototipo [1]. A partir de la evaluación de las características constructivas, funcionales y de costos de fabricación surgidas durante la realización del prototipo se decide introducir algunas modificaciones en el diseño, tanto en el hardware como en el software, lo que deriva en la nueva plataforma móvil RoMAA-II. El desarrollo de esta nueva versión se realiza en el marco de proyecto homologado de la Universidad Tecnológica Nacional, Facultad Regional Córdoba.² Las principales modificaciones realizadas sobre el prototipo fueron: la mecánica del sistema de tracción, el hardware del controlador embebido, junto con el firmware para el cual se crearon librerías específicas de la arquitectura del microcontrolador utilizado [2]. Además, para el control de alto nivel en la PC de a bordo, se evaluó y optó por utilizar el sistema de desarrollo de robótica Player/Stage³. Para esto fue necesario desarrollar un módulo de software que permita controlar el robot mediante Player, además del modelo de simulación del robot RoMAA-II para el simulador Stage [3].

1.1. Objetivos

El proyecto RoMAA surge ante la necesidad de disponer un vehículo adecuado para el ámbito de la investigación, capaz de adaptarse a distintos experimentos en las áreas de navegación autónoma, y visión por computadoras aplicada a la robótica móvil.

Para esto, las dimensiones del robot deben ser adecuadas para ser utilizado en entornos interiores como oficinas, hogares, laboratorios, etc. permitiendo a la vez que puedan montarse sobre el mismo

¹<http://ciiii.frc.utn.edu.ar/>

² “Robot Móvil de Arquitectura Abierta RoMAA-II” U.T.N. PID 1151, homologado por disposición SCyT-UTN N°34/10. Director: Dr. Ing. Gastón Araguás. <http://ciiii.frc.utn.edu.ar/ProyPID2010RoMAA>

³<http://playerstage.sourceforge.net/>

diferentes tipos de sensores como sonares o escáner láser, actuadores como unidades pan&tilt o manipuladores robóticos, etc.

Desde un punto de vista económico se deben utilizar componentes de fácil acceso en el mercado local o, para el caso de las partes mecánicas, deben ser de bajo costo de maquinado y mínimo mantenimiento.

Desde un punto de vista funcional se debe lograr una autonomía suficiente para una jornada completa de trabajo sin interrupción para recarga o cambio de las baterías.

Como sistema de control de alto nivel se debe utilizar una laptop de uso corriente, la cual cuenta con varias ventajas como sus dimensiones reducidas y su gran flexibilidad en la conectividad tanto para una red de comunicaciones, como a dispositivos de a bordo del robot (sensores o actuadores). La potencia de cálculo de las laptops actuales son adecuadas para el procesamiento online de algoritmos de navegación y/o percepción en general.

En base a estos requerimientos de diseño se adopta la arquitectura de tracción diferencial, que permite gran maniobrabilidad en ambientes reducidos con la mínima cantidad de partes mecánicas móviles, reduciendo así el mantenimiento y la complejidad en la fabricación. La tracción diferencial tiene como ventaja adicional la facilidad en el control y la generación de trayectorias del robot.

Además, se plantea como paradigma fundamental del proyecto el ser completamente abierto, lo que implica disponer libremente del diseño de cada una de las partes que componen al robot. Esto permite al usuario de la plataforma acceder sin restricción a cualquier etapa del sistema. En mayor detalles esto implica tener acceso completo al hardware (circuitos esquemáticos y placas de circuitos impresos) y al firmware del controlador diferencial encargado del control de bajo nivel; además de disponer del código fuente del conjunto de programas incluyendo las librerías de comunicación y los módulos de más alto nivel que se ejecutan en la PC de a bordo. Esta arquitectura abierta es esencial para poder adecuar la plataforma de experimentación a las necesidades particulares de cada usuario/investigador, y es una característica que no ofrecen al mayoría de las plataformas disponibles comercialmente al momento de iniciar el proyecto.

1.2. Producción académica

Las diferentes etapas del desarrollo tanto del prototipo del robot como la versión RoMAA-II se llevaron a cabo como parte de tesis de grado de la carrera de Ingeniería Electrónica, y también de Prácticas Profesionales Supervisadas de nuestra Facultad, cuyos avances han sido presentados en diferentes congresos.

Tesis de grado

- “*Módulo de sensores para robot móvil*”, Fernando Elizondo y David Suarez. Tutor: Gonzalo F. Perez Paina. Tesis de grado para ingeniería electrónica, 2011.
- “*Hardware de control de plataforma robótica móvil con arquitectura ARM7 y RTOS embebido. Caracterización.*”, Santiago Pérez y Martín Baudino. Tutor: Gonzalo F. Perez Paina. Tesis de grado para ingeniería electrónica, 2010.

Trabajos en congresos

- “*Design and implementation of a multi-sensor module for mobile robotics applications*”, Gonzalo F. Perez Paina, Fernando E. Elizondo, David A. Suarez y Luis R. Canali. Congreso Argentino de Sistemas Embebidos (CASE), pp. 269-274, ISBN: 978-987-9374-82-7, UTN-FRBA, Buenos Aires, Argentina, 2012.
- “*Librerías embebidas para microcontroladores LPC2000 de aplicación en robótica*”, Gonzalo F. Perez Paina, David A. Gaydou, Néstor Palomeque y Lucas Martini. Congreso Argentino de Sistemas Embebidos (CASE), ISBN: 978-987-9374-69-6, UTN-FRBA, Buenos Aires, Argentina, 2011.
- “*Extrinsic Calibration of a Camera-Robot System under Non-Holonomic Constraints*”, Gastón Araguás, Gonzalo Perez Paina, Guillermo Steiner, Luis Canali. Simposio Argentino de Tecnología de las Jornadas Argentinas de Informática (AST-JAIIO), pp. 157-167, ISSN: 1850-2806, 2011.

- “Programación y simulación en robótica móvil utilizando Player/Stage”, Gonzalo F. Perez Paina y David A. Gaydou. VI Jornadas Argentinas de Robótica (JAR), pp. 150-155. Buenos Aires, Argentina, 2010.
- “Plataforma móvil de arquitectura abierta”, David Gaydou, Gonzalo Perez Paina, Javier Salomone y Guillermo Steiner. V Jornadas Argentinas de Robótica (JAR), EDIUNS, ISBN: 978-987-655-011-6, Bahía Blanca, Buenos Aires, Argentina, 2008.

Prácticas Profesionales Supervisadas

- Relacionada a herramientas de desarrollo libres para microcontroladores de núcleo ARM7, llevada a cabo por Mario Trangoni, bajo la dirección de Gonzalo F. Perez Paina.
- Relacionada al estudio, implementación y evaluación de lazo de control de robot de tracción diferencial, llevada a cabo por Georgina Albornoz, bajo la dirección de Gastón Araguás.

1.3. Presentaciones y transferencias

Además, el robot móvil RoMAA-II resultó premiado en un concurso nacional, fue presentado en diferentes eventos académicos y tecnológicos, y transferido a un Grupo de Investigación Universitario. En más detalles:

- Premio en la categoría robótica de la Octava Edición del Concurso Nacional de Innovaciones, INNOVAR 2012.
- Participación en el workshop de las VII Jornadas Argentinas de Robótica (JAR) 2012, en la ciudad de Olavarría, Buenos Aires, Argentina.
- Participación en la exposición UNC Innova organizada por la Universidad Nacional de Córdoba, en el Parque Científico Tecnológico.
- Participación en la competencia Innovar 2012 y demostración del robot en la exposición de ciencia y tecnología “Tecnópolis 2012”.
- Presentación del robot RoMAA-II en la “Exposición de la Industria, Electrónica e Informática de Córdoba, Expotrónica 2012”.
- Transferencia de un robot RoMAA-II a la Facultad Regional Resistencia (FRRe) de la Universidad Tecnológica Nacional. Por resolución de Rectorado: 1.282/09. Capacitación de investigadores de la FRRe.

2. Descripción general de la plataforma

La arquitectura del robot es de tracción diferencial, tipo unicycle de tres ruedas. Dos de las ruedas son de tracción controladas individualmente y una rueda castor en la parte trasera del robot que sirve de apoyo. Dicha arquitectura resulta apropiada para ambientes interiores, dado que controlando adecuadamente las ruedas de tracción puede lograrse que el robot gire sobre su centro odométrico.

Un robot de tracción diferencial se controla generalmente mediante comandos de velocidad lineal v y angular ω , representados en un sistema de coordenadas local al robot $\langle o_R, x_R, y_R \rangle$, ver Figura 1. A partir de estos comandos el controlador interno del robot genera las velocidades de cada una de las ruedas, las cuales se relacionan con (v, ω) mediante

$$v = R \left(\frac{\omega_R + \omega_L}{2} \right), \quad \omega = R \left(\frac{\omega_R - \omega_L}{b} \right), \quad (1)$$

donde ω_L y ω_R son las velocidades angulares de las ruedas izquierda y derecha, respectivamente; R es el radio nominal de las ruedas, y b la distancia entre ruedas.

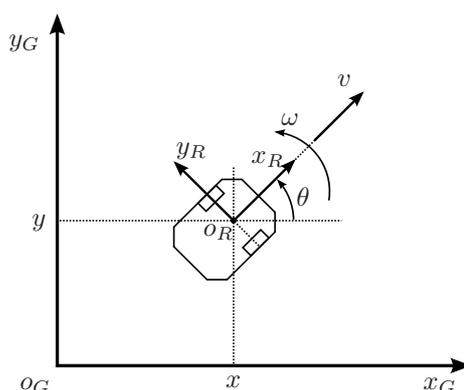


Figura 1: Cinemática de un robot de tracción diferencial.

El movimiento del robot se describe en un sistema de coordenadas global $\langle o_G, x_G, y_G \rangle$ mediante las siguientes ecuaciones cinemáticas

$$\begin{aligned} \dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega. \end{aligned} \tag{2}$$

2.1. Estructura mecánica

La estructura mecánica consiste en dos placas de aluminio de idénticas dimensiones unidas entre sí mediante separadores metálicos, lo que le brinda excelentes características de rigidez y resistencia mecánica a toda la estructura. La placa inferior da soporte a los componentes fundamentales para la operación básica del vehículo: baterías, motorreductores, ruedas y drivers de potencia. La placa superior está dividida en dos partes para poder acceder de forma separada a la sección posterior o anterior del vehículo. Este diseño permite acceder de forma fácil y rápida a los distintos componentes del sistema, como las baterías/cargador, los motorreductores, codificadores ópticos incrementales, y drivers de potencia para el control de tracción del vehículo.

Las dimensiones de la plataforma se muestran en la Tabla 1.

Los componentes de mayor peso están ubicadas en la parte posterior de la plataforma lo cual, en conjunto con el reducido despeje respecto al piso, le confieren al robot gran estabilidad. El área delantera alberga la electrónica para el manejo de los motores de tracción.

La sección posterior de la placa superior está dimensionada para albergar al computador de a bordo (laptop) de control de alto nivel. Además, la placa anterior dispone de una barra con una serie de orificios roscados que permiten el rápido montaje de sensores y actuadores. Las perforaciones se distribuyen regularmente espaciadas a 25 mm y en dos líneas, cada una con rosca estándar M6 particularmente concebida para montar plataformas pan&tilt y cámaras digitales, como se observa en la Figura 2.

| | |
|---------------------------------------|--|
| Dimensiones | Ancho: 520 mm Largo: 570 mm Alto: 200 mm |
| Diámetros de las ruedas | 145 mm |
| Trocha | 455 mm |
| Radio de giro | 0 mm |
| Espacio mínimo necesario para el giro | 410 mm |

Tabla 1: Características mecánicas.



Figura 2: Prototipo del robot móvil de arquitectura abierta RoMAA-II.

2.2. Electrónica de a bordo

La electrónica de a bordo para el control del robot incluye sendos drivers de potencia en configuración llave H para la excitación de cada uno de los motores de tracción y una unidad de procesamiento o *controlador diferencial*.

El controlador diferencial es el encargado de establecer la comunicación con la PC de a bordo y de cerrar los lazos de control de velocidad para cada uno de los motores de tracción, también realiza el cálculo de odometría del robot a partir de las lecturas de los codificadores ópticos incrementales.

Este controlador implementa tres diferentes modos de funcionamiento para operar el robot. El primer modo permite controlar al robot mediante comandos de velocidad lineal y angular, el segundo permite controlar de forma independiente cada uno de los motores de tracción con su respectivo lazo de control mediante comandos de velocidad, y en el tercer modo se controlan los motores directamente con comandos de señal de PWM (Pulse Width Modulation), funcionando a lazo abierto.

La comunicación con la PC de a bordo permite recibir comandos de control como también acceder a variables internas del lazo, lecturas de los codificadores ópticos incrementales (ya sea en forma de cantidad de pulsos, velocidad angular o desplazamiento lineal recorrido), ajustar los valores de las constantes de los controladores digitales, leer diferentes parámetros del robot, etc.

La energía para el funcionamiento del robot es provista por baterías recargables de plomo calcio, libres de mantenimiento, cuyas características principales se muestran en la [Tabla 2](#).

| | |
|---|--|
| Tensión nominal | 12 V |
| Capacidad nominal @ 25°C, 10hs (2.5A, 1.75V) | 25 Ah |
| Dimensiones | Ancho: 166 mm Largo: 175 mm Alto: 126 mm |
| Peso aprox. | 8.9 kg |

Tabla 2: Características de las baterías Probattery BSLA-12260-CPB.

3. RoMAA

El diagrama en bloques del prototipo del robot móvil RoMAA se muestra en la [Figura 3](#).

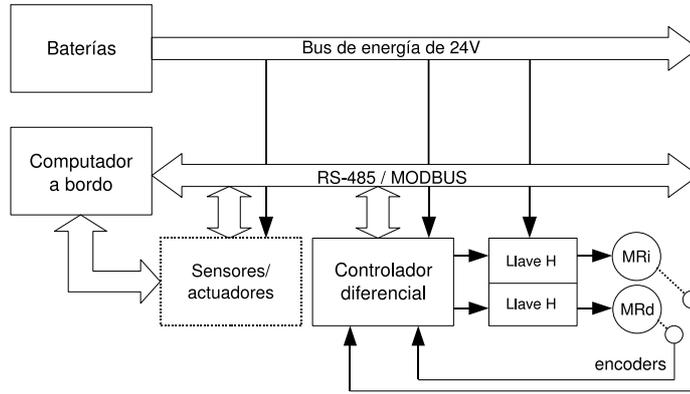


Figura 3: Diagrama en bloques del prototipo del robot RoMAA.

3.1. Componentes electromecánicos y bus de alimentación

El sistema de tracción diferencial del RoMAA está formado por un motorreductor y un codificador óptico incremental por cada rueda, específicamente:

- Motorreductor marca Ignis modelo MR103-400, el cual cuenta con un motor de 6400r.p.m. y 72W, y caja reductora de 16:1.
- Codificador óptico incremental marca Autonics modelo E40H-6-100-2-1-24, de 100p.p.r.

La energía para el funcionamiento del robot es provista por dos baterías recargables de plomo calcio de la Tabla 2. Las baterías se conectan en serie para formar un bus de alimentación de 24V.

Los componentes internos se comunican por medio un bus RS-485 sobre el que se implementa el protocolo de comunicaciones MODBUS.

3.2. Controlador diferencial

El controlador diferencial se encarga de generar las referencias de velocidad para cada motor del robot (izquierdo y derecho) según los comandos de velocidad lineal y angular (v, ω) recibidos como referencia. Dichos comandos son enviados desde la PC de a bordo a través de una conexión serie (RS232). El lazo de control implementado en la versión prototipo se muestra en la Figura 4.

El firmware de este controlador utiliza números enteros para representar todas las mediciones y comandos, y la trama de comunicación con la PC de a bordo está basada en caracteres ASCII, lo que permite depurar el firmware utilizando una terminal de comunicación desde la PC de a bordo. Para enviar órdenes desde la PC se utiliza un joystick, para lo cual se desarrollaron aplicaciones específicas que genera las referencias enviadas al controlador por el puerto serie.

La Figura 5 muestra una versión preliminar del controlador diferencial, donde se utilizó una placa multipropósito desarrollada para evaluar el μC ARM7TDMI (Figura 5a), conectada sobre una placa de soporte que contiene además la fuente de alimentación y los conectores (Figura 5b).

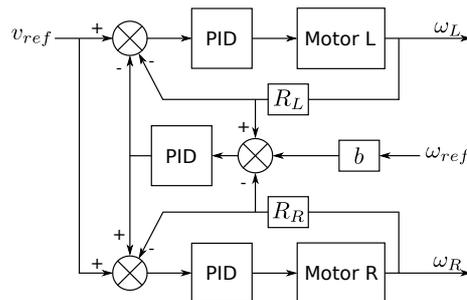


Figura 4: Lazo de control del robot móvil prototipo.

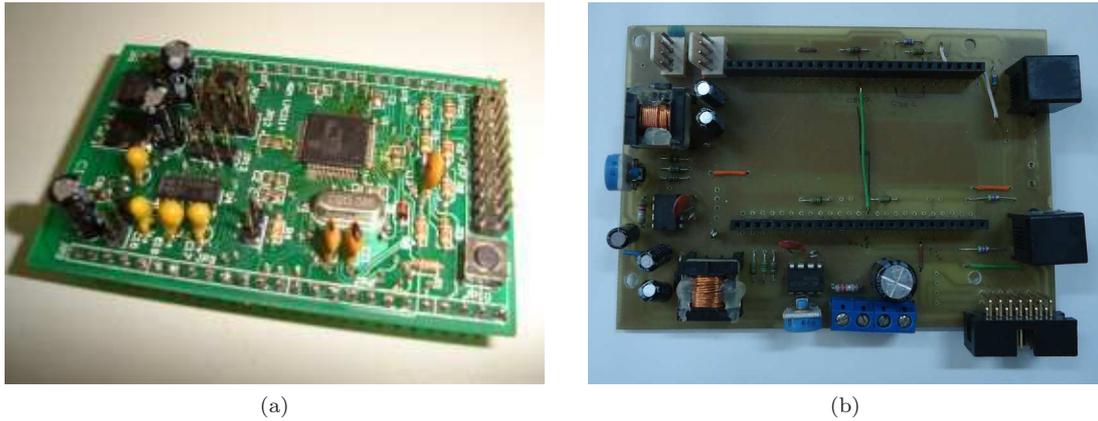


Figura 5: Versión preliminar de placas de circuito impreso (PCB) del controlador embebido utilizadas en el robot RoMAA. (a) PCB multipropósito del μC ARM7. (b) Soporte al PCB del μC , conectores y fuente.

Las versión final del controlador diferencial utilizada en el robot RoMAA se muestran en la Figura 6a. La placa del microcontrolador consta de un único circuito impreso con dos fuentes de alimentación conmutadas para reducir la tensión del bus de alimentación de 24V a 12V y 5V. La Figura 6b muestra el circuito impreso de la llave de potencia en configuración H utilizada para excitar los motores del robot RoMAA.

3.3. Conclusiones

Como resultado de la primer versión del robot móvil RoMAA se obtuvo un prototipo completamente funcional que permitió evaluar el desempeño de las partes componentes del robot, tanto la parte mecánica, eléctrica y de tracción, como la electrónica de a bordo del controlador diferencial y electrónica de potencia.

En cuanto a la parte mecánica se pudo evaluar el desempeño de los motorreductores empleados y la mecánica de acoplamiento entre motor, caja reductora, codificadores ópticos y rueda; a partir de la cual surgieron modificaciones para la versión RoMAA-II. Algunos de los puntos claves fueron:

- Debido a que el eje del motorreductor no soporta el peso del robot, es necesario utilizar un soporte auxiliar con rodamiento haciendo de apoyo al eje de la rueda, Figura 7a
- El acoplamiento entre el motorreductor y el codificador óptico resultó demasiado complejo, demandando un mecanizado especial. Además se encontró mucha dificultad en la alineación de los ejes del motorreductor y el codificador óptico, Figura 7b

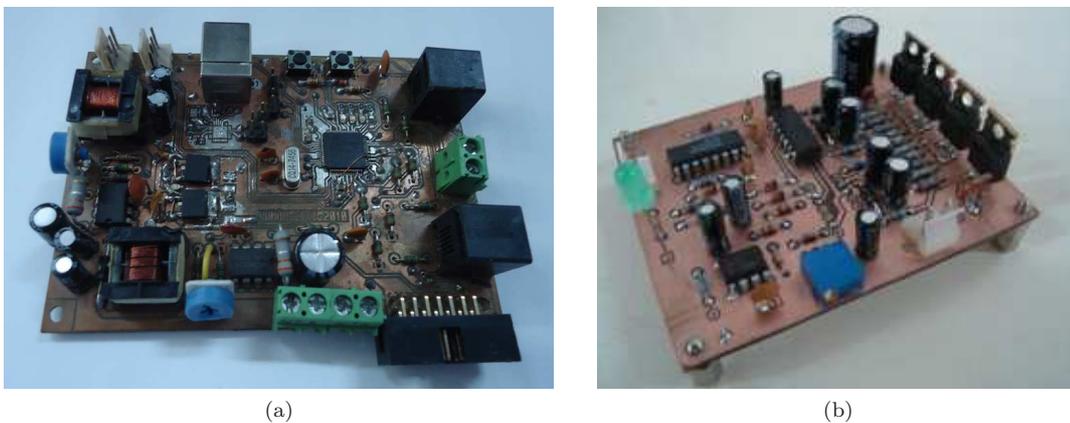


Figura 6: Placas de circuito impreso (PCB) utilizadas en el robot RoMAA. (a) Controlador diferencial con fuentes conmutadas. (b) Llave H utilizada como driver de potencia de los motores.

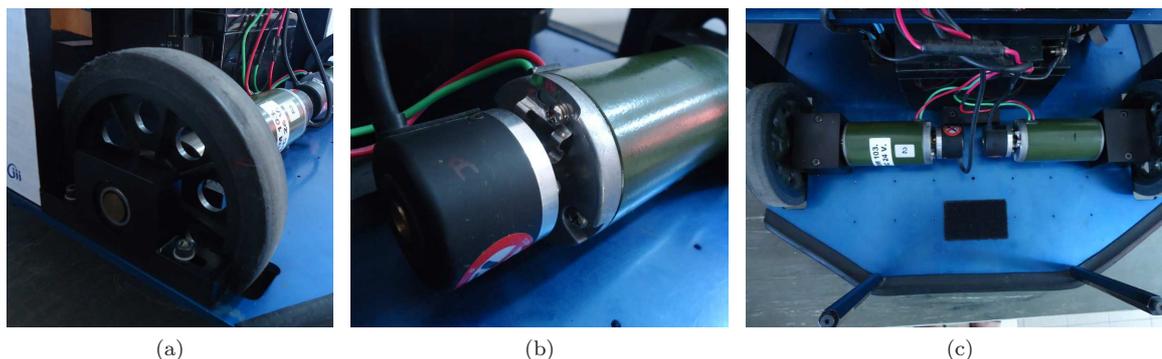


Figura 7: Partes mecánicas modificadas de la versión prototipo. (a) Pieza de apoyo del eje de la rueda. (b) Acople entre los ejes del motor y el codificador óptico. (c) Sujeción de los motorreductores.

- Debido al tipo de sujeción entre el motorreductor y la base del robot, no resultó práctico ajustar la alineación entre los ejes de las ruedas (o paralelismo de las mismas), Figura 7c.

Como consecuencia de estas observaciones realizadas en el RoMAA se plantearon una serie de modificaciones que además de resolver estas dificultades permitieron reducir la cantidad de piezas y la complejidad de mecanizado, con la consiguiente reducción de material y costo.

Respecto al firmware, esta versión era bastante limitada, debido a que las operaciones del microcontrolador eran realizadas solamente con tipo de datos enteros, y el control de tracción se basaba únicamente en un controlador proporcional. El controlador embebido mantenía un registro de los pulsos de codificadores ópticos utilizados por la PC de a bordo para realizar el cálculo de la odometría.

El software de la PC de a bordo se desarrolló en base a una clase de comunicación escrita en lenguaje C++, y se escribió un driver para el entorno de desarrollo de robótica Player (v2.0).

4. RoMAA-II

El robot móvil RoMAA-II pretende mejorar las características tanto constructivas como de prestaciones del prototipo RoMAA.

4.1. Modificaciones respecto al prototipo

Algunas de las modificaciones mecánicas son:

- Reemplazo del motorreductor por un conjunto motor/caja reductora más robusto. La caja es de tipo sinfín y corona y soporta una fuerza radial en el eje suficiente para el peso del robot.
- El motor es de 12V y de fabricación nacional.
- Este nuevo sistema de tracción permite acoplar directamente el codificador óptico al eje de la rueda, permitiendo registrar el movimiento de la rueda debido a huelgos de la caja reductora.

Los principales componentes electromecánicos que forman parte del robot RoMAA-II son:

- Motores marca Remssi modelo 551202 de 2000r.p.m. y 60 W.
- Caja reductora tipo sinfín y corona NMRV 025 de 1:7,5.
- Codificadores ópticos incrementales marca Autonics modelo E40HB10-1000-3-T-24 de 1000p.p.r.
- Baterías (1x) marca Probattery modelo BSLA-12260-CPB (Tabla 2).

El sistema de alimentación se cambió por una única batería de 12V para adecuarse al nuevo motor, resultando también en una simplificación de la parte electrónica. En cuanto al controlador diferencial se rediseñaron tanto la placa del microcontrolador como las llaves H de potencia para la



Figura 8: Robot Móvil de Arquitectura Abierta versión II, RoMAA-II.

excitación de los motores. El nuevo firmware del controlador diferencial del robot RoMAA-II está basado en un sistema operativo en tiempo real (FreeRTOS). Se diseñó e implementó un nuevo lazo de control que resulta de más fácil ajuste que el anterior.

La placa de circuito impreso del controlador diferencial incluye conectores para los codificadores ópticos incrementales, un conector a un panel de control y un adaptador para comunicación USB. Además, el firmware del RoMAA-II opera en punto flotante, realizando internamente el cálculo de la odometría, junto con los lazos de control implementados mediante controladores PID (Proporcional-Integral-Derivativo) digitales para el control de tracción. Este lazo de control permite comandar al robot mediante velocidad lineal y angular en unidades de metros/seg. y radianes/seg., respectivamente.

El software de la PC de a bordo utiliza una librería de comunicación implementada como clase C++ similar a la versión anterior, pero adaptada al nuevo controlador; junto con el driver de entorno de desarrollo de robótica Player (v3.0).

Por último, se diseñó y construyó un sistema de sensores compuesto por un anillo de sensores de ultrasonido y una unidad inercial.

4.2. Descripción general

El robot móvil RoMAA-II se muestra en la Figura 8, y su diagrama en bloques en la Figura 9.

4.3. Características técnicas

- Motores de tracción: 2000 r.p.m. - 60W.

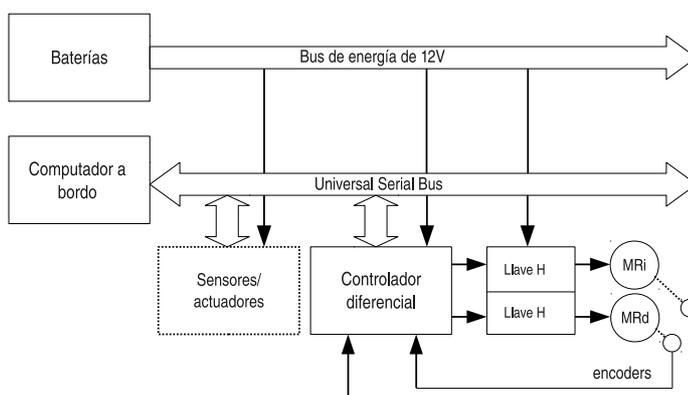


Figura 9: Diagrama en bloques del robot RoMAA-II.

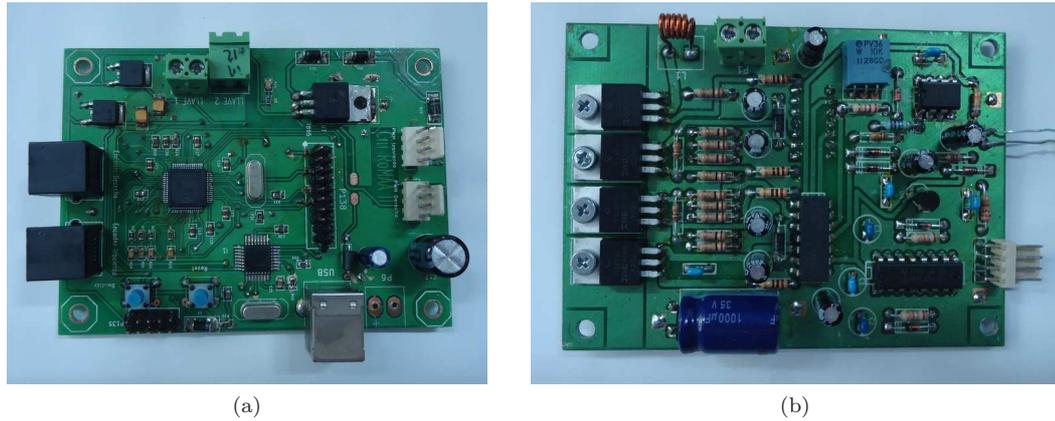


Figura 10: Placas de circuito impreso (PCB) del controlador embebido utilizadas del RoMAA-II. (a) PCB del microcontrolador. (a) Llave H utilizada como driver de potencia de los motores.

- Caja reductora: 7,5 : 1.
- Diámetro de las ruedas: 147mm.
- Distancia entre ruedas: 455mm.
- Peso aproximado: 3Kg.
- Velocidad máxima: 2m/s.

4.4. Electrónica de a bordo

La electrónica de a bordo para el control del robot incluye los dos drivers de potencia en configuración llave H para la excitación de cada uno de los motores de tracción y el controlador diferencial.

El controlador diferencial consiste en un microcontrolador con núcleo ARM7TDMI de 32bits y 60MHz de frecuencia de reloj, específicamente el modelo LPC2114 de NXP. La Figura 10 muestra la electrónica del robot RoMAA-II.

El firmware del controlador diferencial del robot RoMAA-II está basado en un sistema operativo en tiempo real (FreeRTOS) [4], que administra las diferentes tareas llevadas a cabo por el μC . Estas tareas incluyen los lazos de control en velocidad para los motores de tracción, cálculo de odometría del robot a partir de las lecturas de los codificadores ópticos incrementales y la comunicación con la PC de control de alto nivel a bordo del vehículo.

4.4.1. Comunicación

La comunicación entre el controlador diferencial y la PC de a bordo se realiza a través de un puerto USB (Universal Serial Bus) utilizando una trama de comunicación RS-232. En la Figura 11 se muestra la estructura del paquete utilizado en la comunicación.

Un paquete de comunicación de comandos comienzan con un *inicio de paquete* (INIT), representado por un valor ASCII particular (concretamente el caracter '*'). A continuación se indica la cantidad de bytes que conforman el paquete. El sector de datos (DATA) consiste en un byte de identificación de comando seguido de un campo de datos no obligatorio, dependiendo del comando específico.

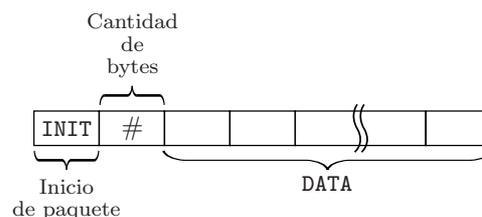


Figura 11: Paquete de comunicación del controlador diferencial.

Los comandos implementados permiten escribir y leer parámetros de configuración, enviar comandos de movimiento y leer valores obtenidos a partir de la lectura de los codificadores ópticos incrementales. También se dispone de comandos para leer y ajustar valores límites o umbrales bajo y alto del sistema energético. El umbral alto permite fijar el valor de tensión que activa un indicador luminoso cuando la tensión de la batería es inferior a un valor prefijado. Mientras que el umbral bajo fija un valor de tensión de batería por debajo del cual se produce la deshabilitación de la tracción del robot (este bloqueo automático por tensión baja de la batería puede ser activado o desactivado con sus respectivos comandos).

La Tabla 3 contiene los comandos implementados con su descripción.

4.4.2. Lazo de control de velocidad

La velocidad de giro de las ruedas se determina midiendo el ancho de los pulsos de los codificadores incrementales acoplados a sus ejes. Para esto se utiliza el módulo de captura del μC además de una señal de entrada/salida general para determinar el sentido de giro de los codificadores ópticos, mediante las señales de salida (fases A y B). Como base de tiempo se usa el reloj interno del μC .

El diagrama en bloques del lazo de control de velocidad implementado se muestra en la Figura 12, el cual hace uso de las relaciones entre las velocidades lineal y angular (v, ω) del robot, con las velocidades angulares de las ruedas izquierda y derecha (ω_L, ω_R), dadas en (1).

4.4.3. Cálculo de odometría

El cálculo de odometría permite estimar la pose del robot (x_k, y_k, θ_k) en el paso temporal k , partiendo de la odometría en el paso temporal anterior ($x_{k-1}, y_{k-1}, \theta_{k-1}$) y las distancias recorridas por cada una de las ruedas entre dichos intervalos. Dadas las cantidades de pulsos entre pasos temporales del codificador óptico acoplado a la rueda izquierda N_L y derecha N_R , se convierten en las distancias recorridas utilizando el factor de conversión

$$resolucion_distancia_rueda = \frac{\pi \cdot diametro_rueda}{resolucion_cod_incremental \cdot reduccion_motor} \quad (3)$$

Por lo que se tiene

$$l_L = N_L \cdot resolucion_distancia_rueda \quad (4)$$

$$l_R = N_R \cdot resolucion_distancia_rueda \quad (5)$$

donde l_L y l_R son las distancias recorridas por las ruedas izquierda y derecha respectivamente en cada paso temporal. La distancia recorrida por el centro odométrico y el cambio en orientación del robot están dados por

$$\Delta l = \frac{l_L + l_R}{2}, \quad \Delta \theta = \frac{l_R - l_L}{b} \quad (6)$$

En [5] se realiza el cálculo detallado de las ecuaciones de odometría basados en geometría, cuyas

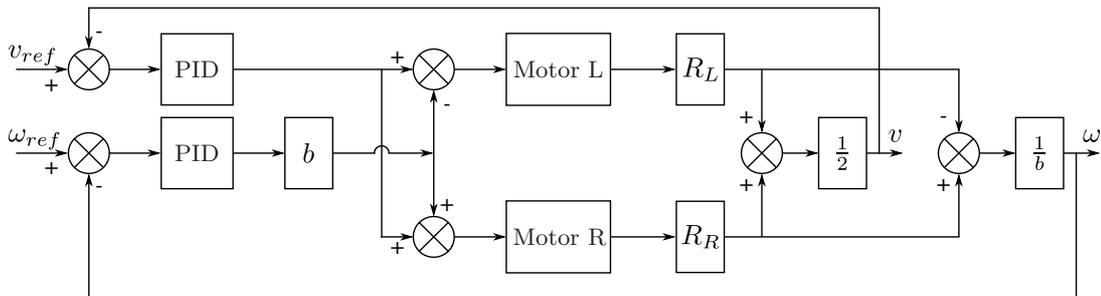


Figura 12: Lazo de control del robot móvil RoMAA-II.

| Comandos generales | |
|---|--|
| CMD_RESET | Inicialización general del controlador |
| CMD_SET_PWM | Fijar los valores de de los PWM [%] de las llaves H |
| CMD_GET_PWM | Leer los valores de de los PWM [%] de las llaves H |
| Comandos del lazo de control | |
| CMD_OPEN_LOOP_MODE | Fija modo de lazo abierto |
| CMD_CLOSE_LOOP_MODE | Fija modo de lazo cerrado (v, ω) |
| CMD_GET_LOOP_MODE | Petición del modo de lazo |
| CMD_SET_V_PID | Fijar las constantes (Kp, Ki, Kd) de PID del lazo de velocidad lineal |
| CMD_GET_V_PID | Leer las constantes (Kp, Ki, Kd) de PID del lazo de velocidad lineal |
| CMD_SET_W_PID | Fijar las constantes (Kp, Ki, Kd) de PID del lazo de velocidad angular |
| CMD_GET_W_PID | Leer las constantes (Kp, Ki, Kd) de PID del lazo de velocidad angular |
| Comandos movimiento | |
| CMD_ENABLE_MOTORS | Habilitación de los motores a través de las llaves H |
| CMD_DISABLE_MOTORS | Deshabilitación de los motores a través de las llaves H |
| CMD_SET_SPEED | Fijar la velocidad lineal [m/s] y angular [rad/s] del robot |
| CMD_GET_SPEED | Leer la velocidad lineal [m/s] y angular [rad/s] del robot |
| CMD_SET_WHEEL_SPEED | Fijar la velocidad lineal [m/s] de de las ruedas |
| Comandos odometría | |
| CMD_RESET_ODOMETRY | Inicializa los valores de odometría (0, 0, 0) |
| CMD_SET_ODOMETRY | Fijar los valores de odometría (x, y, θ) [m,m,rad/s] |
| CMD_GET_ODOMETRY | Leer los valores de odometría (x, y, θ) [m,m,rad/s] |
| CMD_GET_ENC_COUNTER | Leer los valores de los contadores de los codificadores ópticos |
| CMD_GET_WHEEL_V | Leer la velocidad lineal [m/s] de las ruedas |
| CMD_GET_WHEEL_W | Leer la velocidad angular [rad/s] de las ruedas |
| CMD_GET_WHEEL_D | Leer la distancia recorrida de cada rueda en cada ciclo de μC |
| Comandos de logs de datos | |
| CMD_LOG_ODOM | Log de datos de odometría |
| CMD_LOG_SPEED | Log de datos de velocidad lineal y angular |
| CMD_LOG_PWM | Log de datos de valores de PWM |
| CMD_LOG_ENC_COUNTER | Log de datos de los contadores de los codificadores ópticos |
| CMD_LOG_WHEEL_V | Log de velocidad lineal de las ruedas |
| CMD_LOG_WHEEL_W | Log de velocidad angular de las ruedas |
| CMD_LOG_WHEEL_D | Log de distancia recorrida por las ruedas |
| CMD_LOG_STOP | Detiene el log de datos |
| Comandos de parámetros | |
| CMD_SET_KINEMATIC | Fijar los parámetros cinemáticos del robot |
| CMD_GET_KINEMATIC | Leer los parámetros cinemáticos del robot |
| CMD_GET_BODY_GEOM | Leer los valores de dimensiones geométricas del robot |
| Comandos de nivel y control de batería | |
| CMD_GET_BATTERY_LEVEL | Leer tensión de la batería |
| CMD_SET_BATTERY_L_THR | Fijar el umbral mínimo de aviso de nivel de batería |
| CMD_GET_BATTERY_L_THR | Leer el umbral mínimo de aviso de nivel de batería |
| CMD_SET_BATTERY_H_THR | Fijar el umbral máximo de aviso de nivel de batería |
| CMD_GET_BATTERY_H_THR | Leer el umbral máximo de aviso de nivel de batería |
| CMD_ENABLE_CHECK_BATTERY | Habilitar el control de nivel de batería |
| CMD_DISABLE_CHECK_BATTERY | Deshabilitar el control de nivel de batería |
| CMD_SET_ADC_CONV_CTE | Fijar la constante de conversión de ADC para medición de batería |

Tabla 3: Comandos implementados en el controlador embebido del RoMAA-II.

ecuaciones finales son

$$x_k = x_{k-1} + \Delta x_k \approx x_{k-1} + \frac{\sin(\Delta\theta/2)}{\Delta\theta/2} \Delta l \cos\left(\theta_{k-1} + \frac{\Delta\theta}{2}\right) \quad (7)$$

$$y_k = y_{k-1} + \Delta y_k \approx y_{k-1} + \frac{\sin(\Delta\theta/2)}{\Delta\theta/2} \Delta l \sin\left(\theta_{k-1} + \frac{\Delta\theta}{2}\right) \quad (8)$$

$$\theta_k = \theta_{k-1} + \Delta\theta_k \approx \theta_{k-1} + \Delta\theta \quad (9)$$

Notar que como $\sin\phi/\phi \rightarrow 1$ cuando $\phi \rightarrow 0$ el término $\sin(\Delta\theta_k/2)/(\Delta\theta_k/2)$ tiende a la unidad cuando el robot realiza movimientos con poca curvatura, por lo tanto este término se puede ver como un *factor de ajuste* para la localización del robot en movimientos circulares.

La aproximación implementada en el firmware del controlador diferencial del robot RoMAA-II, es

$$x_k = x_{k-1} + \Delta l \cos\left(\theta_{k-1} + \frac{\Delta\theta}{2}\right) \quad (10)$$

$$y_k = y_{k-1} + \Delta l \sin\left(\theta_{k-1} + \frac{\Delta\theta}{2}\right) \quad (11)$$

$$\theta_k = \theta_{k-1} + \Delta\theta \quad (12)$$

4.5. Software de la PC de a bordo

El software de la PC de a bordo del robot RoMAA-II está implementado para correr en sistemas operativos GNU/Linux. Incluye una librería de comunicación implementada como clase POO (Programación Orientada a Objeto) en C++ que permite enviar y recibir datos al controlador diferencial del robot mediante métodos análogos a cada uno de los comandos propios de bajo nivel listados en la Tabla 3. Además, se incluye la posibilidad de utilizar el entorno de desarrollo de robótica de código abierto Player/Stage, para lo cual se dispone del driver correspondiente para el robot RoMAA-II. Por lo tanto, el desarrollo de aplicaciones para utilizar el robot se pueden llevar a cabo de dos maneras diferentes:

- Utilizando la librería de comunicación escrita en C++,
- o utilizando el entorno de desarrollo de robótica de código abierto Player [3].

La utilización de Player permite controlar el robot a través de una interfaz estándar, además de diferentes sensores y actuadores a los que Player brinda soporte. Player es un servidor de dispositivos utilizados en robótica [6] basado en sockets que proporciona una interfaz simple a sensores y actuadores en redes TCP/IP, la abstracción de los sockets posibilita la independencia del lenguaje de programación y de la plataforma de trabajo. Player consta de dos partes, una parte funciona como servidor de red para el control de robots y corre a bordo del robot. Actúa como una capa de abstracción de hardware (HAL) para dispositivos robóticos en el caso de robots reales, Figura 13a; o bien sirve de interfaz al simulador Stage, Figura 13b. La otra parte son las librerías clientes que brindan acceso a los dispositivos remotos (reales o simulados) en el lado del servidor.

Stage es un simulador de múltiples robots que simula una población de robots, sensores y objetos en un entorno 2D; dispone de robots virtuales de modo que Player interactúe con el entorno simulado en lugar de los dispositivos físicos, además de varios modelos de sensores incluyendo sonares, sensores de barrido láser, cámaras pan-tilt-zoom, odometría, etc. Stage es adecuado en investigación de sistemas autónomos multi-agente, debido a que se basa en un modelo simple de bajo requerimiento computacional de múltiples dispositivos en lugar de emular cada dispositivo con gran fidelidad. Stage se puede utilizar también como un simulador independiente de Player, mediante la programación de controladores que se asocian a un modelo del entorno simulado, el cual se carga en tiempo de ejecución [7].

4.6. Sensores de ultrasonido

Uno de los sensores que normalmente se encuentran disponibles en los robots comerciales son los anillos de ultrasonido, ya que su uso es muy común en tareas de navegación. En la Figura 15

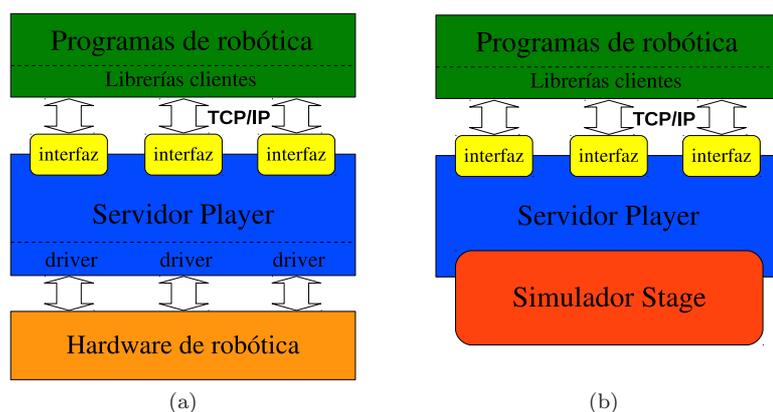


Figura 13: Formas de utilizar el entorno de desarrollo de robótica Player/Stage. (a) Player con hardware de robótica. (b) Player con el simulador Stage.

se muestra el anillo de ultrasonidos o sonar (SOund Navigation and Raging) desarrollado para el robot RoMAA-II [8]. El módulo de sonares es un desarrollo independiente que puede ser adaptado a diferentes robots. Está basado en el mismo microcontrolador ARM7TDMI de 32bits que utiliza el controlador del robot. El anillo está compuesto por nueve transductores electrostáticos de ultrasonido, los cuales se encuentran multiplexados para poder operar con un único módulo de medida. El microcontrolador maneja todas las señales del módulo de medida como también del multiplexor para generar un barrido de mediciones de todos los transductores. El módulo de sonares se comunica con la PC de a bordo del robot a través de un puerto USB con trama RS232.

La Figura 16a muestra el tipo de transductor utilizado y la Figura 16b el módulo de medición, ambos de la firma SensComp Inc.

El módulo de sensores tiene además una unidad inercial compuesta de sensores tipo MEMS (Microelectromechanical System) que incluye acelerómetro, giróscopo y magnetómetros, todos de tres ejes. La Figura 17a muestra el diagrama en bloques del módulo de sensores, y la Figura 17b la electrónica del módulo terminada.

De forma similar al controlador diferencial del robot RoMAA-II, la utilización del módulo sensor por parte de la PC de a bordo puede hacerse por medio de en una librería (clase de C++) de comunicación o de su correspondiente driver para el entorno de desarrollo de robótica Player.

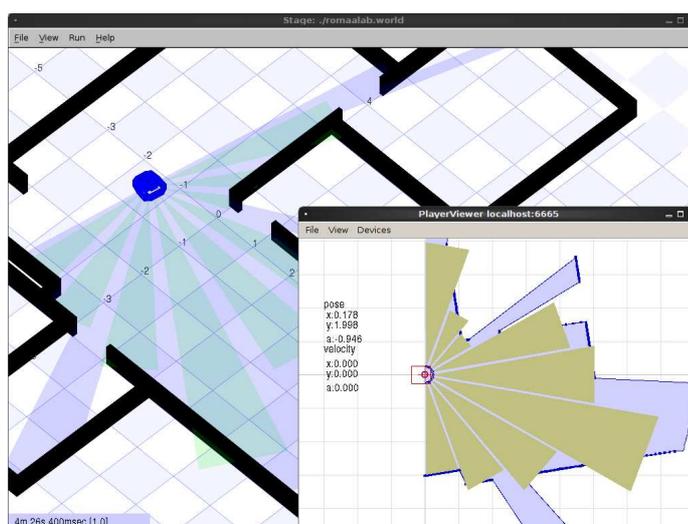


Figura 14: Simulador Stage con el modelo del robot RoMAA y sensores.



Figura 15: Robots móviles RoMAA-II disponibles en el CIH con diferentes sensores abordo. El primero de la izquierda cuenta con el anillo de ultrasonido.

5. Conclusiones

El diseño y desarrollo del Robot Móvil de Arquitectura Abierta –RoMAA– resultó en una experiencia altamente enriquecedora tanto para los principales involucrados, miembros activos del Centro de Investigación en Informática para la Ingeniería, como así también para los diferentes estudiantes de grado de la carrera de ingeniería electrónica que participaron en el mismo, ya sea en concepto de becarios de investigación, tesistas, o en la realización de la Práctica Profesional Supervisada exigida por la carrera de grado de ingeniería en electrónica de la UTN. La experiencia adquirida en el desarrollo del presente proyecto, permite fortalecer el vínculo entre nuestro Centro de Investigación y la carrera de grado, resultando en conocimientos y experiencia adicional de los estudiantes a la obtenida por la curricula obligatoria de las diferentes cátedras. Además del conocimiento técnico y experiencia profesional realizada en la construcción del robot RoMAA, resultó en un trabajo en equipo que incluyó continuamente más de una decena de personas.

Se obtuvo una plataforma de experimentación para ser aplicado a la investigación en robótica móvil y visión por computadoras de prestaciones similares a las disponibles comercialmente. Además, el hecho de contar con el conocimiento de cada una de las partes que componen al robot RoMAA-II, desde la mecánica, electrónica de a bordo y componentes de software para el desarrollo, permite al usuario tener acceso al diseño de cada parte del robot para ser modificado y adaptado a sus necesidades. Por otro lado, es importante mencionar que en el diseño y construcción del robot RoMAA se utilizó mayormente software libre, incluyendo herramientas de diseño electrónico (esquemáticos y PCB), programación y depuración del firmware del microcontrolador, como también las herramientas utilizadas para desarrollar el software para el control del robot.

Por último, la experiencia adquirida en el desarrollo de este proyecto permite poder adaptar el diseño del robot para ser aplicado a tareas específicas, las cuales esta teniendo cada vez mayor impacto de aplicación de la robótica móvil.

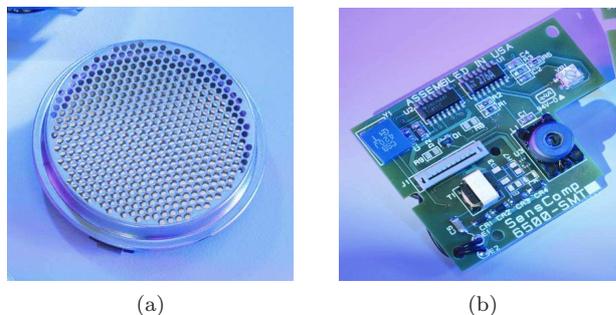


Figura 16: Componentes del anillo de ultrasonido de SensComp Inc. (a) Transductor serie 600. (b) Módulo de medida 6500.

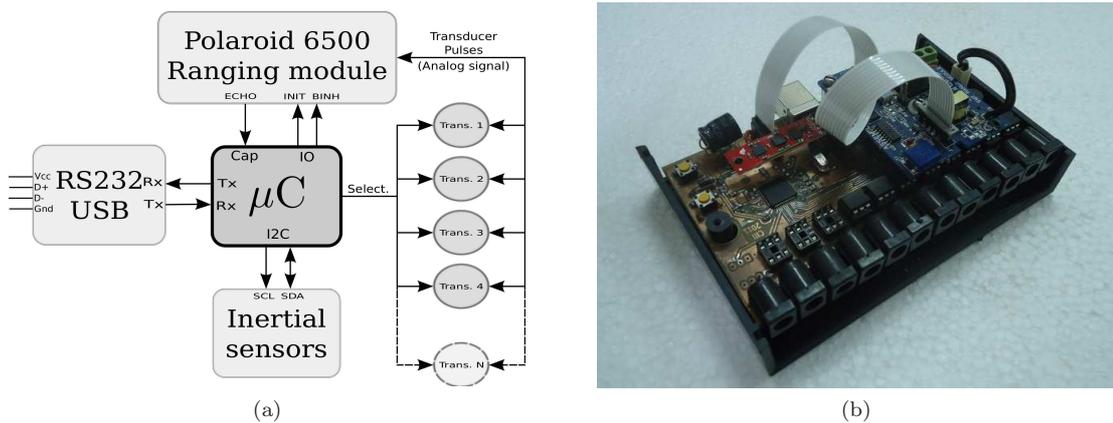


Figura 17: Módulo de sensores. Anillo de ultrasonido y unidad inercial. (a) Diagrama en bloques. (b) Sistema embebido.

Referencias

- [1] D. A. Gaydou, G. F. Pérez Paina, G. M. Steiner, and J. Salomone. Plataforma móvil de arquitectura abierta. In *Proceedings of the V Jornadas Argentinas de Robótica (JAR 2008)*. Ediuns, November 2008. 2
- [2] Gonzalo F. Perez Paina, David A. Gaydou, Néstor L. Palomeque, and Lucas A. Martini. Librerías embebidas para microcontroladores lpc2000 de aplicación en robótica. In *Proceedings of the Argentine Conference on Embedded Systems (CASE 2011)*, 2011. 2
- [3] Gonzalo F. Perez Paina and David A. Gaydou. Programación y simulación en robótica móvil utilizando player/stage. In *Proceedings of the VI Jornadas Argentinas de Robótica (JAR 2010)*, pages 150–155, November 2010. 2, 14
- [4] R. Barry. Freertos reference manual. <http://www.freertos.org/>. 11
- [5] C.M. Wang. Location estimation and uncertainty analysis for mobile robots. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 1231–1235, apr 1988. 12
- [6] B.P. Gerkey, R.T. Vaughan, K. Stoy, A. Howard, G.S. Sukhatme, and M.J. Mataric. Most valuable player: a robot device server for distributed control. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 3, pages 1226–1231 vol.3, 2001. 14
- [7] Richard Vaughan. Massively multi-robot simulation in stage. *Swarm Intelligence*, 2(2-4):189–208, 2008. 14
- [8] Gonzalo F. Perez Paina, Fernando E. Elizondo, David A. Soares, and Luis R. Canali. Design and implementation of a multi-sensor module for mobile robotics applications. In *Proceedings of the Argentine Conference on Embedded Systems (CASE 2012)*, pages 269–274, 2012. 15

El documento fue editado por Gonzalo F. Perez Paina con revisión de R. Gastón Araguas.

En el diseño y desarrollo tanto del prototipo como de la versión final (RoMAA-II) participaron en general personal permanente de CIII, como también diferentes becarios (rentados y ad-honorem) tanto de grado como graduados. Ellos son:

- Gastón Araguás, Gonzalo Perez Paina, Guillermo Steiner, David Gaydou, Javier Salomone, Claudio Paz, Diego Gonzalez Dondo, Lucas Martini, Daniel Marchetti, Martín Baudino, Santiago Pérez, Ariel Delfino, Estefanía Pereyra, Mauro Pereira, Guillermo Bustos, Nicolás Rey, Ignacio Bevacqua, Néstor Palomeque, Jeremías de la Cruz, Georgina Albornoz, Mario Trangoni.