

“Hermes”: Sistema Robótico Embebido para la Educación

Leandro Federico Lichtensztein^{#1}, Orlando Micolini^{#2}, Marcelo Cebollada^{#3}

^{#1}Laboratorio de Arquitectura de Computadoras,
Facultad de Ciencias Exactas, Físicas y Naturales,
Universidad Nacional de Córdoba
Av. Vélez Sarsfield 1611, Córdoba, Argentina

¹leandrosnm@gmail.com, ²omicolini@compuar.com, ³mcebollada@gmail.com

Abstract— Currently, using robots for teaching is widely accepted. Several studies have shown how this sort of device eases learning.

This paper addresses the design and development of a robot called “Hermes”, devised for high schools and universities to be programed, manufactured and further developed.

Component selection and design followed ease of use, functionality and affordability criteria.

The robot compatibility with software tools presently used, such as the Lego Mindstorms system, was taken into account.

The project resulted in a robot which can be programed and built by students and allows inclusion of new features.

This paper demonstrates that designing and making robotic devices for teaching is possible.

Resumen— En la actualidad el uso de robots en la enseñanza es ampliamente aceptado. Se ha probado en numerosos estudios cómo el uso de estos dispositivos facilita el aprendizaje.

En este trabajo se presenta el diseño y construcción de un robot denominado “Hermes”, cuya finalidad es ser utilizado en la educación a nivel secundario y universitario tanto en su programación, ampliación y construcción.

Se siguieron criterios de funcionalidad, costo y simplicidad para el diseño y la selección de los componentes.

Se tuvo en cuenta la compatibilidad del robot aquí diseñado con las herramientas actualmente usadas, como el sistema Lego Mindstorms y otros.

Como resultado de este trabajo se obtuvo un robot con capacidad de ser programado y construido por los alumnos, con posibilidades de incorporar nuevas características.

Este trabajo muestra que es posible diseñar y construir dispositivos robóticos con fines de enseñanza.

I. INTRODUCCIÓN

Se llama constructivismo a la corriente pedagógica desarrollada por Jean Piaget. En esta se propone que el conocimiento se construye a partir de la interacción de quien aprende con el medio, más precisamente establece que una persona entiende el mundo que lo rodea a través de modelos mentales que elabora. Se desprende de lo anterior que el aprendizaje es un proceso dinámico en el cual el sujeto desempeña un papel activo, no sólo incorporando información, sino también, produciéndola [1].

Seymour Papert, discípulo de Piaget, catedrático del MIT (Massachusetts Institute of Technology) y fundador del Laboratorio de Medios de la misma universidad, desarrolló una corriente derivada del constructivismo llamada construccionismo. Esta se basa en la idea de que los modelos mentales se desarrollan de manera mucho más

efectiva cuando el estudiante construye objetos del mundo real [1].

En particular, Papert, establece la conexión entre la computación y la enseñanza, haciendo foco en al campo de la ciencia y la tecnología.

En el año 1967, junto con un grupo de investigadores, Papert desarrolla un lenguaje de programación educativo llamado Logo. Una de las principales funciones de este lenguaje es la de producir gráficos mediante comandos a un cursor de la misma manera que se haría con un robot, indicando la dirección del movimiento y la distancia a recorrer. De la misma manera, con iguales instrucciones, es posible controlar un robot llamado Turtle, utilizado en enseñanza [2]

A comienzos de los 80s la empresa Lego Group, comenzó a financiar las investigaciones desarrolladas por el Laboratorio de medios del MIT. Inicialmente se desarrolló un robot programable con Logo, posteriormente, en 1998, Lego lanzó el Lego Mindstorm.

La versión actual de este robot es utilizada en niveles de educación secundarios y universitarios.

Se han desarrollado numerosos estudios en todo el mundo acerca de las mejoras en el aprendizaje producidas por la utilización de robots de enseñanza (mayormente Lego Mindstorm). En todos se observa un aumento significativo en el nivel de desarrollo de las capacidades intelectuales y de aprendizaje [3].

A nivel universitario, en particular en las carreras de ingeniería, la robótica es una rama de estudio que requiere integración de conocimientos de diferentes áreas. En particular en el diseño y construcción de un robot están involucrados la electrónica, la programación, las comunicaciones, los sistemas de control, la mecánica, etc. haciendo de un proyecto de este tipo el ambiente ideal para la integración de estos conocimientos.

Actualmente se implementa en la Facultad de Ciencias Exactas, Físicas y Naturales un proyecto que tiene por objetivo elevar el rendimiento de los alumnos en el aprendizaje de la informática. Para lograr este fin se propuso, entre otras cosas, “desarrollar, implementar y evaluar actividades prácticas basadas en la construcción y programación de robots en los cursos de Ingeniería” [4].

Esta estrategia se basa en la premisa de que “la utilización de la robótica como herramienta de enseñanza constituye una actividad altamente motivante [5], donde los estudiantes pueden autoevaluarse observando si el robot hace o no lo pedido en las consignas de trabajo” [4].

A partir de lo anteriormente mencionado, surge naturalmente la necesidad de construir robots que puedan ser utilizados en la enseñanza. Es fundamental tener en cuenta las posibilidades reales de una institución educativa en Argentina.

En cuanto a los requerimientos técnicos es necesario un robot con capacidades mecánicas de desplazamiento, y capacidades de software para ser programado con algoritmos de identificación y trayectorias.

En el presente trabajo se realiza el diseño y construcción de un robot educativo, llamado “Hermes” (mensajero de los dioses del olimpo) que cumple con las premisas aquí mencionadas, haciendo viable su fabricación y uso dentro de la universidad y de los colegios secundarios.

II. MÉTODOS

Se optó por emplear un proceso de desarrollo iterativo con entregas incrementales ya que este permite satisfacer los requerimientos más críticos al inicio del desarrollo disminuyendo así el riesgo total.

Partiendo de los objetivos planteados se extrajo una primera aproximación del sistema a construir: “Debe ser un robot con capacidad de movimiento con tres grados de libertad (dos de desplazamiento y uno de rotación sobre un eje perpendicular al plano de desplazamiento), estos movimientos se realizan mediante algoritmos pre programados en el mismo. Requiere de sensores para reconocer el ambiente y tener parámetros para decidir el movimiento a realizar. Finalmente ha de ser capaz de mantener comunicaciones bidireccionales con otros dispositivos.”

Las Fig. 1 y Fig. 2 muestran los casos de uso que se desprenden de lo anterior.

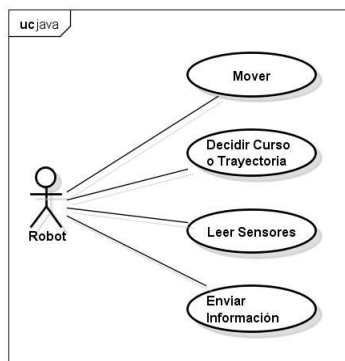


Fig. 1. Casos de uso del Sistema

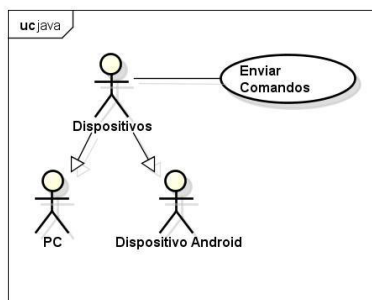


Fig. 2. Caso de uso Programación

A partir de los casos de uso planteados se procedió a dividir el sistema en subsistemas. La arquitectura

seleccionada fue del tipo centralizada [6], esto quiere decir que un módulo central comunica y sincroniza los distintos subsistemas que componen el robot (Fig. 3. Arquitectura del Sistema).

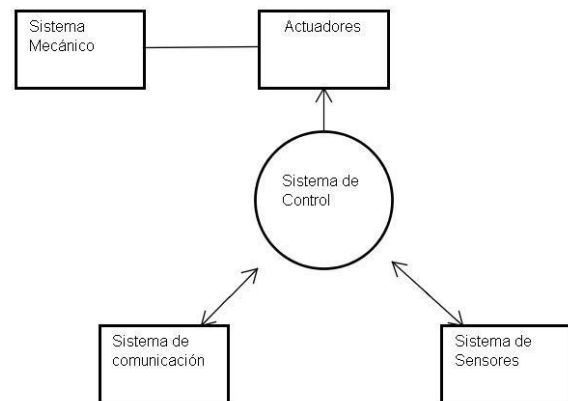


Fig. 3. Arquitectura del Sistema

En esta arquitectura el riesgo está dividido entre los módulos, de manera que si uno de ellos falla no es necesario rediseñar el sistema. También facilita el mantenimiento ya que cada parte puede modificarse individualmente sin impactar en el resto.

Como se ve en la Fig. 3. el sistema de control envía y recibe información del sistema de comunicación y del sistema de sensores. Con esta información comanda los actuadores que realizan el movimiento sobre el sistema mecánico.

Habiendo establecido los casos de uso y la arquitectura del sistema se procedió a plantear una lista de requerimientos funcionales del mismo:

1. Movimiento.
2. Programable.
3. Reconocimiento del ambiente.
4. Envío de datos.
5. Recepción de datos.
6. Ejecución de Hilos.

A su vez, se establecieron requerimientos no funcionales, dividiéndolos en requerimientos generales, de hardware y de software. Estas especificaciones marcaron los criterios a través de los cuales se eligieron los componentes del sistema.

Habiendo definido los requerimientos del proyecto se planteó una matriz de relaciones entre los subsistemas y los requerimientos funcionales (Tabla 1) y un diagrama de dependencias (Fig. 4).

TABLA 1
MATRIZ DE RELACIONES

Módulo Req.	Sistema Mecánico	Sistema de Control	Sistema de Comunicación	Sistema de Sensores
1	X	X		
2		X		
3		X		X
4		X	X	
5		X	X	
6		X		

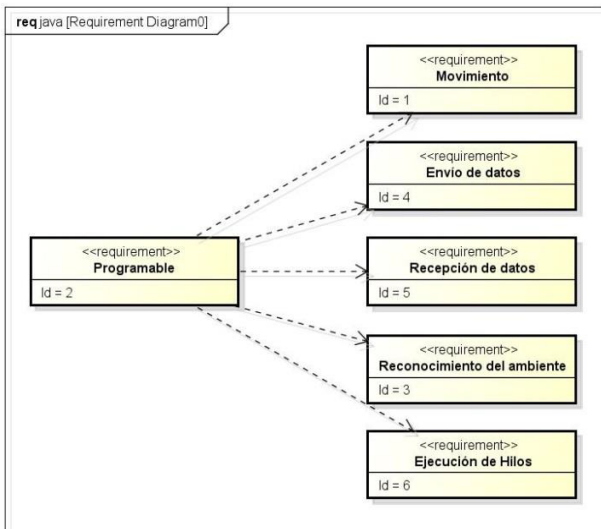


Fig. 4. Diagrama de Dependencias

III. MATERIALES

Los componentes fueron elegidos de acuerdo a los requerimientos no funcionales establecidos [7].

A. Sistema Mecánico

El sistema de cuatro ruedas omnidireccionales permite el traslado en dos dimensiones y la rotación en su propio eje, por otro lado esto brinda una mayor programabilidad del dispositivo.

Para implementar este sistema es necesario que todas las ruedas tengan tracción propia y que puedan variar su ángulo.

B. Actuadores

Se tuvieron en cuenta motores eléctricos: de corriente continua, brushless, paso a paso y servos.

Se decidió utilizar servos para el control de los ángulos y servos modificados para las ruedas. Esta elección se basó principalmente en el bajo costo de estos motores, en la simplicidad para su uso y en su tamaño compacto.

Se utilizaron motores Tunirgy TG9e, estos servos se conocen como micro servos. Poseen engranajes de plástico en la caja de reducción pero el engranaje de salida del motor de corriente continua es de metal lo que le da robustez.

C. Microcontrolador

Para la elección del microcontrolador se analizaron numerosas posibilidades. Habiendo descartado aquellas que no estaban disponibles en el mercado nacional se procedió a comparar en profundidad las restantes. Se dejaron de lado aquellas placas que no poseían las características electrónicas necesarias para el proyecto y aquellas de mayor costo [7].

La elección final fue entre las placas Kiwistick k40 de Freescale y el Arduino Mega2560.

La primera presenta un precio superior (sobre todo en los accesorios) y su programación es mucho más compleja, se requiere configuración de registros y escritura de código a bajo nivel. Sin embargo posee una enorme capacidad de procesamiento en comparación con el micro procesador de 8 bits del Arduino. Analizando lo anterior se opta por el

Arduino, ya que en los requerimientos se especifica la necesidad de un sistema económicamente viable y simple de programar, mientras que la capacidad de procesamiento es menos relevante.

D. Sensores

Se decidió integrar al sistema diferentes sensores [7].

- Dos sensores infrarrojos (TCRT5000).
- Un sensor de color (TCS230).
- Una placa IMU (del inglés *inertial measurement unit*), compuesta por un acelerómetro (ADXL345), un giróscopo (L3G4200D), un magnetómetro (HMC5883L) y un barómetro con sensor de temperatura (BMP085).
- Un sensor de distancia por sonar (HC-SR04).

E. Comunicación

Se analizaron tres formas de comunicación inalámbrica para el robot, Wi Fi, Bluetooth y Zigbee.

Luego de estudiar las tres posibilidades se eligió usar bluetooth, en particular el módulo HC-05, debido a su bajo costo y alta disponibilidad en dispositivos existentes [7].

Debido al requerimiento de bajo costo, no es posible incluir una placa con Wi-Fi, de cualquier manera las ventajas que este dispositivo brinda, alta velocidad y cifrado, no eran necesarias en este proyecto.

Zigbee no fue elegido por el poco uso en dispositivos domésticos, es decir es necesario un adaptador para permitir a una PC utilizar la comunicación, y es complejo para una Tablet o un Smartphone.

F. Arquitectura del Sistema

Se planeó la siguiente arquitectura incorporando todos los componentes seleccionados (Fig. 5).

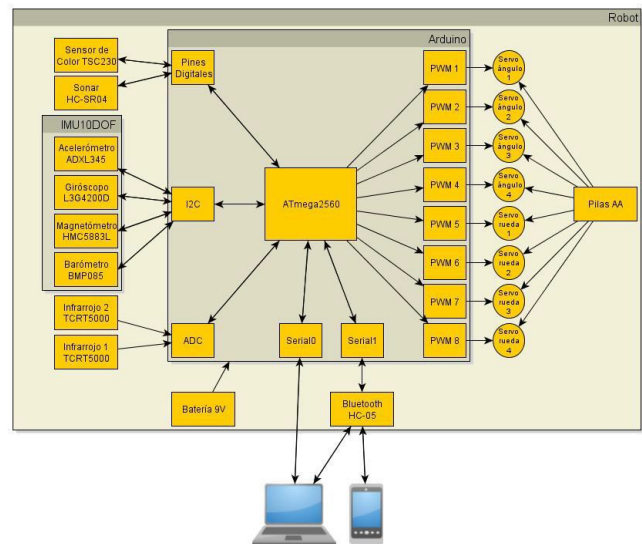


Fig. 5. Arquitectura del Sistema

del movimiento del robot. Se sincronizaron utilizando semáforos binarios.

Finalmente se implementó una aplicación en C++/QT que permite, a través del uso de expresiones regulares, la conversión entre la sintaxis de NXC, lenguaje usado para programar los robots Lego, y la sintaxis de Arduino, obteniendo así código compilable para el robot “Hermes”.

V. RESULTADOS

El sistema final obtenido es un robot móvil terrestre holonómico. Su sistema mecánico está compuesto por un chasis de aluminio y ocho servos, cuatro utilizados para las ruedas y cuatro que permiten que estas se orienten en cualquier ángulo (Fig. 9).

El sistema de control está centrado en la placa de desarrollo Arduino Mega2560. El sistema de comunicación está realizado mediante bluetooth, permitiendo el envío y recepción de datos de manera inalámbrica (módulo HC-05).

El sistema posee: dos sensores infrarrojos (TCRT5000), un sensor de color (TCS230), un sensor de distancia por sonar (HC-SR04) y una IMU de diez grados de libertad. Esta placa contiene un acelerómetro (ADXL345), un giróscopo (L3G4200D) y un magnetómetro (HMC5883L), todos de tres ejes. También incorpora un barómetro y un termómetro (BMP085). La comunicación entre la IMU y el Arduino se realiza mediante el bus I2C.

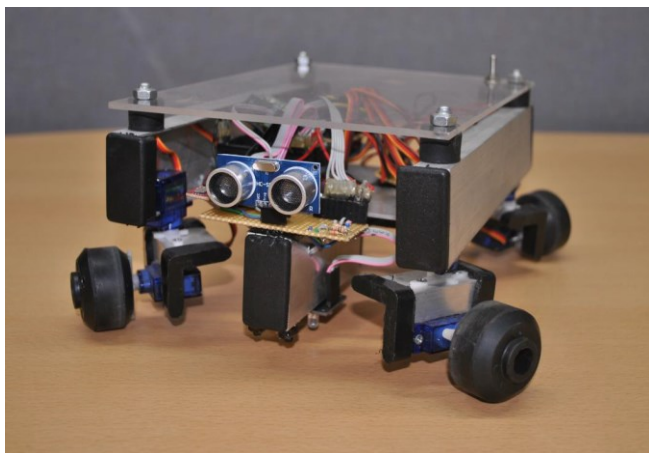


Fig. 9. Sistema Final

Este robot fue utilizado exitosamente por trescientas personas durante un mes (en la feria Cuatrociencias – UNC 2013). Se documentaron los veintitrés incidentes, los cuales fueron solucionados en el diseño e implementación.

Con respecto al software, la biblioteca permite modelar el funcionamiento de todos los componentes del sistema. Posee una clase central, la cual contiene los comandos de movimiento disponibles para el robot: avanzar, retroceder, giro horario y giro anti horario. Con lo cual se puede componer todos los movimientos posibles para un vehículo terrestre.

Se crearon clases para los sensores infrarrojos, de distancia, de color, y para el giróscopo. La instanciación de un objeto, de estas clases, permite controlar el sensor deseado. Mediante el uso de los métodos es posible configurar y realizar mediciones con cualquier sensor.

Se desarrolló a su vez la documentación y ejemplificación del uso de esta biblioteca. A su vez se

elaboró la documentación del sistema operativo NiRTOS y se acompañó con un ejemplo de su uso.

Con respecto a los robots Lego se logró parcialmente la traducción de un sistema al otro (Fig. 10).

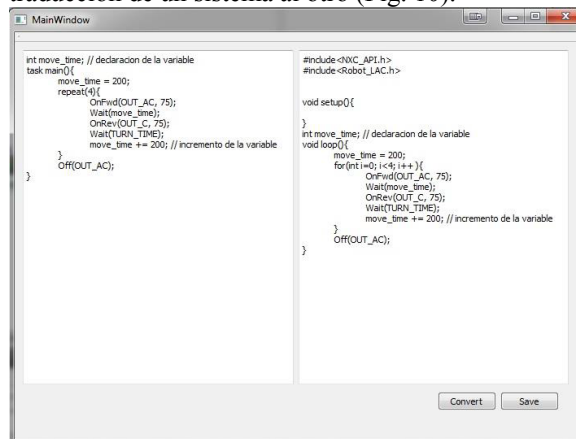


Fig. 10. Programa traductor

Se diseñó un experimento con ocho alumnos de la asignatura Programación Concurrente, los cuales, usando los robots Hermes, realizaron la programación de un sistema de producción compuesto por tres robots y cuatro máquinas. Se lograron tiempos y esfuerzos 53% menores a los resultados obtenidos con los robots Legos.

VI. CONCLUSIÓN

Se desarrolló con éxito un sistema compuesto por un robot con capacidad de realizar movimientos holonómicos y una biblioteca que permite la utilización de sensores con los cuales se pueden decidir y controlar trayectorias.

Mediante la prueba realizada en la muestra Cuatrociencia se mostró que el uso de este tipo de dispositivos es intuitivo y no requiere conocimientos elevados, haciéndolo ideal para la enseñanza en niveles primarios y secundarios. El uso intensivo del robot ha mostrado la robustez del diseño.

Con respecto al costo del robot, se determina que el mismo es económicamente viable y que es un producto competitivo en el mercado. Más importante que lo anterior es mencionar que es viable realizar proyectos aplicables a la realidad sin necesidad de grandes inversiones.

A través de la aplicación de traducción se determinó que el sistema desarrollado es perfectamente compatible con los sistemas que poseen mayor trayectoria en el campo de la educación a nivel mundial. Se puede establecer que este desarrollo puede contar con todas las características y capacidades del Lego Mindstorm. Se ha previsto un middleware que permite el uso de todas las herramientas de dicho sistema, las cuales están validadas y verificadas para la enseñanza.

Finalmente, el trabajo realizado representa una innovación tecnológica flexible para ser usada en la educación. Dentro de un ambiente educativo es provechoso el proceso de construcción del robot, su programación y uso.

Como desarrollos futuros se realizará una IDE que permita programar el Arduino con código de NXC.

Queda pendiente implementar todas o algunas de las funciones de la API en la biblioteca desarrollada. Para algunos casos esto representa sólo cambiar el nombre de un método, en otros casos representa modificar la implementación.

REFERENCIAS

- [1] E. Ackermann, «Piaget's Constructivism, Papert's Constructionism: What's the difference?», *MIT Learning Media Publications.*, 2010.
- [2] M. Resnick, «Behavior Construction Kits», *Communications of the ACM*, vol. 36, n° 7, 1993.
- [3] J. L. Bruni, «Robótica Educativa.» Mar del Plata, 2011.
- [4] C. A. Bartó y L. C. Díaz, «Sistemas inteligentes aplicados a la enseñanza de la programación en ingeniería.» de *XIV Workshop de Investigadores en Ciencias de la Computación*, 2012.
- [5] C. y. o. Fernández Panadero, «Impact of Learning Experiences Using LEGO Mindstorms in Engineering Courses», *IEEE EDUCON 2010 Conference*, pp. 503 -512, 2010.
- [6] Ollero Baturone, *Robótica: Manipuladores y robots móviles*, Barcelona: Marcombo S.A., 2001.
- [7] L. F. Lichtensztein, «Sistemas embebidos para la educación: "Hermes"», Laboratorio de Arquitectura de Computadoras, Facultad de Ciencias Exáctas, Físicas y Naturales, Universidad Nacional de Córdoba, Córdoba, Argentina, 2013. Disponible: https://www.academia.edu/7018308/Sistemas_embebidos_para_la_educacion_Hermes_