

Butiá: Plataforma robótica genérica para la enseñanza de la informática

Gonzalo Tejera, Andrés Aguirre, Federico Andrade, Pablo Guindel, Santiago Margini y Jorge Visca
{gtejeralaaguirrelfandradelpablodlsmarginiljvisca}@fing.edu.uy

Instituto de Computación, Facultad de Ingeniería, Universidad de la República J. Herrera y Reissig 565,
Montevideo, Uruguay

Resumen—El aprendizaje de la robótica en los niveles iniciales de la educación es una herramienta poderosa para transmitir a los profesores, estudiantes y sus familias conocimientos básicos sobre las nuevas tecnología y sus aplicaciones. Existen muchos mitos sobre las computadoras y los robots, desconocimientos básicos tanto sobre lo que pueden como lo que a no pueden hacer, en ambos sentidos, y que genera por un lado miedos infundados y por otro expectativas desmedidas. La incorporación de los robots y de la inteligencia computacional se está dando de manera progresiva en nuestra sociedad, y es importante entonces contribuir a mejorar el conocimiento de estas tecnologías. Este artículo presenta una plataforma simple y económica que permita a los alumnos de los liceos públicos del Uruguay en coordinación con los docentes, interiorizarse con la programación del comportamiento de robots.

Index Terms—Educación inicial, enseñanza de informática y robótica.

I. INTRODUCCIÓN

LOS robots son una herramienta pedagógica poderosa y flexible. Permite a los estudiantes realizar elaboraciones mentales de orden superior, reflexionar sobre el por qué de las cosas, experimentar e identificar las repercusiones de las decisiones que se toman y comprenderlas.

El proyecto Butiá[1] desarrollado con fondos de la Agencia Nacional de Investigación e Innovación crea una plataforma que permite a alumnos de liceos públicos, en coordinación con docentes e inspectores del Consejo de Enseñanza Secundaria (CES), interiorizarse con la programación del comportamiento de robots.

El proyecto provee a los liceos seleccionados por el CES una plataforma simple y económica que permite a los alumnos definir el comportamiento de un robot autónomo, que proporciona ambiente lúdico e idóneo para que los estudiantes se interioricen con la programación y la robótica.

Es importante señalar que, en cuanto a la robótica, existe actualmente una profunda asimetría entre liceos públicos y privados, extendiéndose de manera creciente la robótica en la currícula de muchas instituciones privadas. Esta propuesta pretende acercar sistemas robóticos a instituciones públicas de nivel secundario en las que hasta el momento, este tipo de sistemas no ha sido ampliamente incorporado.

II. LA ROBÓTICA Y LA EDUCACIÓN

Dada la creciente importancia que tiene la tecnología hoy en día y el amplio terreno que viene ocupando la robótica

y la mecatrónica en el mundo, resulta conveniente comenzar a incorporar estos conceptos en las primeras etapas de la formación educativa de nuestros jóvenes. En este sentido, es interesante ofrecer a alumnos de educación secundaria la posibilidad de acercarse a nuevas tecnologías a través del manejo de robots y lenguajes de programación simples que les permitan controlarlos. Se espera con esta experiencia que los alumnos dispongan de un elemento más para definir su vocación hacia orientaciones Científico ■ Tecnológicas.

Programar los comportamientos de un robot móvil genera mucho interés para los jóvenes. Además, les permite alcanzar resultados visuales inmediatos de sus programas, estimula su creatividad, así como al mismo tiempo se les enseñan conceptos básicos de programación, como por ejemplo: variable, operador, estructuras de control y función. Se entiende que el trabajo con robots, desde una perspectiva de la robótica pedagógica potencia el desarrollo del aprendizaje inductivo y por descubrimiento guiado; posibilita el diseño de situaciones didácticas que permiten a los estudiantes construir su propio conocimiento. [2]

Esta propuesta busca generar entornos de aprendizaje centrados en la actividad de los propios estudiantes. Uno de los factores a destacar es la posibilidad de integración de las diferentes áreas, como matemáticas, ciencias experimentales, comunicación, filosofía, entre otras, que amplían la propuesta de trabajo a nivel de los centros educativos, posibilitando que se desarrollen proyectos de fin de año integrando varias asignaturas al trabajo con los robots, como se propone en el plan de estudios vigente de bachillerato.

Los alumnos de estos liceos podrán presentar los trabajos realizados sobre el sistema robótico en el marco del evento de robótica organizado por la Facultad de Ingeniería de la Universidad de la República[3], simultáneamente con trabajos realizados por alumnos y docentes universitarios, generando un rico ambiente de intercambio de experiencias.

El proyecto Butiá se apoya, en todo sentido, sobre las computadoras OLPC (One Laptop per Child)[4] proporcionadas al sistema de educación público del Uruguay a través del Plan Ceibal[5], llevado adelante por el gobierno nacional del Uruguay.

III. ARQUITECTURA DEL LA PLATAFORMA BUTIÁ

Al momento de diseñar la arquitectura del sistema se tuvo en cuenta como principal objetivo la portabilidad del sistema,

así como su funcionamiento en plataformas de hardware con pocas prestaciones y la versatilidad al momento de integrarse con diferentes lenguajes de programación.

Para poder cumplir con los requerimientos de portabilidad se utilizó el lenguaje ANSI C para los componentes de software que tienen interacción directa con el hardware y donde es necesario minimizar las latencias vinculadas con los tiempos de comunicación. Para la lógica de mayor nivel de abstracción y más propensa a cambios, se decidió utilizar el lenguaje interpretado Lua dado que es un lenguaje de gran nivel de abstracción, su máquina virtual es muy pequeña, funciona en plataformas con bajas prestaciones y en cualquier arquitectura donde haya un compilador de ANSI C. Dado que es un lenguaje interpretado se evita generar binarios para las distintas arquitecturas, lo cual es un gran beneficio al momento del desarrollo, simplificando el testeado y puesta en producción sobre la plataforma objetivo.

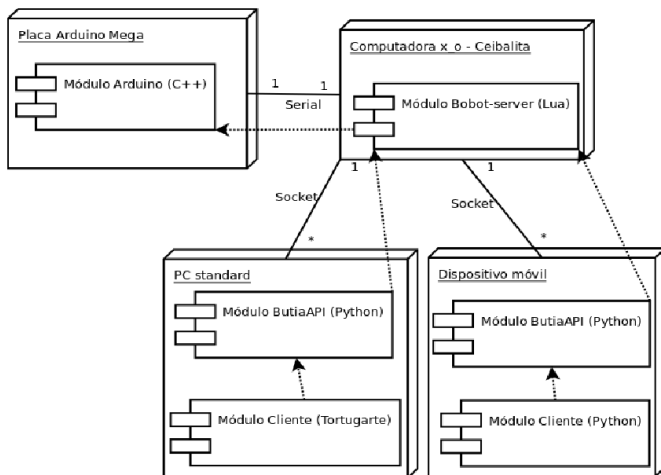


Figura 1. Diagrama de componentes.

En la figura Figure 1 on page 2 se incluye el diagrama de componentes y deploy, se puede apreciar como interactúan las distintas componentes del hardware, que módulos contiene cada componente y como se comunican.

De forma simplificada la arquitectura Butiá consta de 3 capas, tomando en cuenta un enfoque bottom-up, podemos identificar el nivel más cercano al hardware, la capa0, encargada de interactuar con los sensores y actuadores, en esta capa se especifican los servicios que el firmware brinda, estos servicios son agrupados de forma lógica en componentes de software llamados, módulos de usuario los cuales son una abstracción del sensor o actuador que se desea controlar en el robot. Estos componentes de software residen en el firmware de la placa de entrada/salida (E/S). La separación entre esta capa y la siguiente esta bien definida por un stack de protocolos, lo cual permite obtener un gran nivel de independencia del hardware subyacente, el prototipo construido es funcional con la placa de E/S Arduino y actualmente se presenta un gran nivel de avance con la placa USB4all [6], pero se espera portar a otro tipo de plataformas como Handy Cricket, PicoBoard o GoGoBoard. La siguiente capa, la capa2, se encarga de descubrir de forma dinámica los módulos presentes en la placa

de E/S junto con los servicios o funcionalidades que estos brindan, una vez descubiertos son expuestos para poder ser consumidos por una aplicación. Para permitir un uso más versátil al momento de integrarse con diferentes lenguajes de programación se realizó una tercer capa, que lleva estos servicios a un nivel mayor de abstracción permitiendo ser invocados en la red. Esta última capa permite interactuar con el robot desde cualquier lenguaje de programación que tenga soporte de sockets.

La arquitectura construida es, genérica, permitiendo una fácil extensión de sus funcionalidades e independiente del hardware de bajo nivel subyacente, así como portable a diferentes plataformas de bajos recursos de hardware.

A diferencia de otras plataformas donde la lógica de control se desarrolla completamente en una placa de E/S microcontrolada, la arquitectura Butiá fomenta un diseño híbrido, donde solo se implementa en la placa de E/S la interacción directa con el hardware encapsulándolo en los módulos de usuario, luego la lógica de control es desarrollada en un lenguaje de alto nivel en base a los servicios que estos módulos exponen. Esto permite utilizar herramientas de mayor nivel de abstracción, permitiendo generar código con un mayor nivel de mantenibilidad y abstracción, permitiendo ser comprendido por alumnos que se inician en las ciencias de la computación.

El sistema operativo Sugar instalado en las computadoras OLPC incluye diversos entornos y lenguajes de programación. Entre ellos se enceta el entorno de peroración gráfico Tortugarte[7] basado en el lenguaje LOGO. En el marco del proyecto se extendió el lenguaje agregando una paleta para interactuar con el robot Butiá. En la parte superior de la figura2 se muestra la paleta Butia integrada al entorno Tortugarte que permite controlar los motores y acceder a la lectura de los sensores. En la parte inferior de la misma se presenta un comportamiento para evitar obstáculos utilizando el sensor botón.

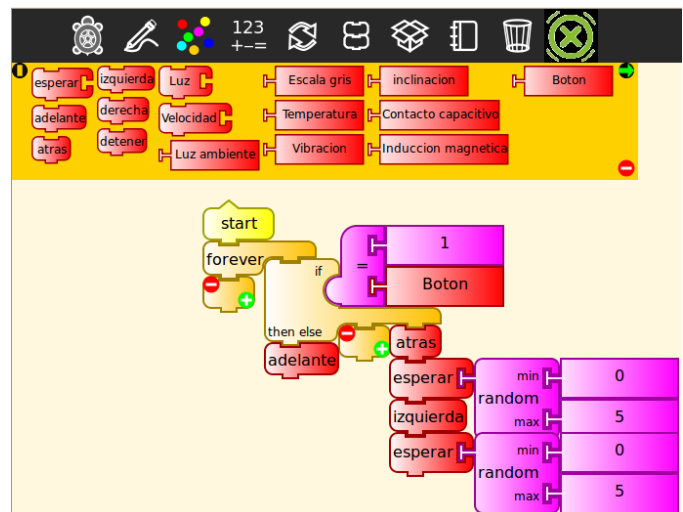


Figura 2. Evitando obstáculos con Tortugarte.

IV. IMPLEMENTACIÓN DEL PROTOTIPO

Uno de los requerimientos de la plataforma es ser altamente portable. Se espera que Butiá soporte al menos las siguientes

plataformas:

1. OLPC. Esta es la plataforma del Plan Ceibal. El hardware contiene un procesador AMD Geode, con arquitectura x86. Los primeros modelos poseían 256Mb RAM. El disco duro es de estado sólido. El software consiste en un kernel Linux con Sugar como interfaz de usuario.
2. Intel Classmate. Es la plataforma para enseñanza de Intel, y es un nombre genérico para una línea de productos de distintos fabricantes. El hardware es típico para un Netbook de bajo costo: procesador Intel Atom (x86), a partir de 512Mb de RAM, disco duro magnético.
3. Router Inalámbrico. Plataforma de costo y poder de cómputo mínimo prevista. Un ejemplo típico es un router Asus 520GU. Consiste en un sistema embebido con un procesador Broadcomm, con 32Mb de RAM y 8Mb de almacenamiento Flash. En el marco del proyecto se instala OpenWRT, una distribución de Linux para sistemas embebidos.
4. Smartphones. Hay multitud de fabricantes y de plataformas de software. Usualmente contienen un procesador ARM, ms de 64Mb de RAM y almacenamiento flash. El sistema operativo puede estar basado en Linux, Windows ME, u otros sistemas dedicados.

Se plantea la necesidad de disponer de un componente que pueda ser desplegado con mínimas modificaciones en todas las plataformas de interés, y que implemente las siguientes funcionalidades:

- Acceda a la placa de E/S implementando el protocolo USB4all sobre el tipo de conexión asociado (USB o Serial).
- Ofrezca una API que permita acceder las funcionalidades provistas por Butiá desde las aplicaciones de usuario.

Este componente se implementó en Lua como dos componentes: una biblioteca que implementa la comunicación con la placa (bobot), y una aplicación que usa esta biblioteca y exporta su funcionalidad mediante un socket (bobot-server). Esta arquitectura es la solución de referencia, de máxima portabilidad.

Este componente se implementa en Lua. El bobot accede a la placa microcontroladora mediante USB o Serial. Para la primera opción, se enlaza con libusb, una biblioteca portable de espacio de usuario para manipular dispositivos USB. Este enlace se realiza mediante un binding desarrollado, llamado lualibusb[8]. Para acceder mediante serial se utiliza una pequeña librería en C que implementa una comunicación orientada a mensajes sobre serial llamada serialcomm. Esta librería es fácilmente extensible para agregar soporte a nuevos módulos de la placa controladora (USB4all/Arduino). Esto se logra mediante drivers cargados dinámicamente de acuerdo a los módulos declarados por la placa controladora.

En las plataformas robóticas comercialmente más difundidas, el usuario debe llevar nota de qué dispositivo conectó en cada puerto de comunicación y declararlo en algún punto de la estructura del software. Esto difiere radicalmente respecto a la capacidad de auto-configuración o *plug&play* de la plataforma Butiá. Esta característica pretende un sistema lo más abierto y potente posible, pensado para dar soporte a la mayor cantidad

de dispositivos, tanto en número como en diversidad de los mismos. La idea es exponer al usuario las capacidades de E/S del microcontrolador de una forma ordenada que facilitara su aprovechamiento. Siguiendo estos principios de diseño un conector PnP flexible, en el cual se puedan conectar tanto sensores como actuadores, tanto analógicos como digitales, PWM e incluso I2C.

Ventajas del uso de los conectores PnP Butiá:

- Constituye una potente herramienta de diagnóstico.
- Simplifica el acceso al hardware.
- Permite que el firmware seleccione automáticamente el módulo adecuado para manejar cada tipo de dispositivo.

Se diseñó una placa adaptadora a la controladora de E/S que dispone de ocho conectores PnP Butiá, y en forma separada el conector para el bus Dynamixel (que también cuenta con auto-detección) y un conector de potencia para dos motores de continua o un motor de paso.

Cuadro I
ESPECIFICACIONES DEL ROBOT BUTIÁ

Elemento	Instancia
Motores	Dynamixel AX12
Control de bajo nivel	Arduino Mega
Control de alto nivel	Computadora OLPC
Chasis	Acrílico 5mm
Amortiguación	Placa acero alto carbono 0.5mm
Sensores	Kit DFRobot para Arduino e infrarrojo

V. CONCLUSIONES Y TRABAJO FUTURO

Es interesante ofrecer a los estudiantes la posibilidad de acercarse a nuevas tecnologías a través del manejo de robots y lenguajes de programación simples con los que pueden controlarlos. Los alumnos se interesan mucho por la temática de del proyecto dando lugar a un elevado nivel de atención e interacción durante los talleres. Para muchos fue un tema nuevo que resultó interesante, ayudando o contribuyendo, en algunos casos, en la definición o realización de sus proyectos curriculares.

En todos los casos hemos tratado de generar nexos entre los participantes, los docentes de la localidad y nuestro grupo de forma de generar proyectos tecnológicos en relación a la robótica.

Se espera que el robot Butiá continúe creciendo tanto en cantidad de centros en los que trabaja como en las prestaciones, características y soporte de hardware. Se espera llegar a las escuelas donde se encuentran masivamente las OLPC, otras universidades y centros de educación.

Se diseñó una arquitectura que brinda un enfoque genérico e independiente del hardware de control. Esta arquitectura permite desarrollar el comportamiento del robot Butiá mediante lenguajes de programación de fácil acceso y comprensión para estudiantes liceales como es Tortugarte. De todas formas al estar modularizado y bien definido el alcance de cada capa, así como el API para acceder a cada una de ellas, existen diferentes niveles en los cuales el alumno puede desarrollar el comportamiento, dependiendo de su nivel de conocimientos, llevando esto desde la programación de nuevos módulos de

usuario en el firmware, así como programación en otros lenguajes como Lua, Python, Java, VisualBasic.

A continuación se presenta las extensiones sobre las cuales se está trabajando actualmente:

- Jabber.
- Integrar a lenguaje hablado y Scratch
- Utilizar tecnología de desecho.
- Portabilidad a otras placas de E/S.
- Utilización de los sensores de la XO (video y audio).

REFERENCIAS

- [1] ButiÃa, "Proyecto butiÃa," Agosto 2010, <http://www.fing.edu.uy/inco/proyectos/butia/>.
- [2] A. SÃnchez Colorado, MÃnica MarÃa y Gauthier, "Ambientes de aprendizaje con robÃstica pedagÃgica," IngenierÃa ElÃctrica y ElectrÃnica - Universidad de los Andes, Memo de InvestigaciÃn IEL 2003-001, 2005.
- [3] sumo.uy, "Campeonato de sumo robÃstico," Agosto 2010, <http://www.fing.edu.uy/inco/eventos/sumo.uy/>.
- [4] OLPC, "One laptop per child," Agosto 2010, <http://laptop.org/>.
- [5] Ceibal, "Portal del plan ceibal," Agosto 2010, <http://www.ceibal.edu.uy/>.
- [6] C. G. AndrÃs Aguirre, Pablo FernÃndez, "Interfaz usb genÃrica para comunicaciÃn con dispositivos electrÃnicos," Diciembre 2007.
- [7] S. Labs, "Turtleart activity," Agosto 2010, <http://wiki.sugarlabs.org/go/Activities/TurtleArt>.
- [8] A. A. Jorge Visca, "Libusb binding for lua," Agosto 2010, <http://luaforge.net/projects/lualibusb>.