

Butiá: Plataforma robótica genérica para la enseñanza de la informática

Gonzalo Tejera Andrés Aguirre

Federico Andrade Pablo Gindel Santiago Margni
Guillermo Reisch Jorge Visca

Instituto de Computación, Facultad de Ingeniería, Universidad de la República
J. Herrera y Reissig 565, Montevideo, Uruguay
<http://www.fing.edu.uy/inco/proyectos/butia>
butia@fing.edu.uy

4/03/2011

Introducción

Objetivos

Prototipo

Arquitectura Butiá

Arquitectura de Hardware

Implementación

Los niños toman el control

Implantación

Entrega de robots

Conclusiones y trabajo a futuro

...Butiá 2.0

finalmente...

Más que mil palabras

Gracias por su tiempo



CASE

Congreso Argentino de Sistemas Embebidos

Objetivos

Proyecto Butiá

- ▶ Crear una plataforma simple y económica que permita a alumnos de liceos públicos interiorizarse con la programación del comportamiento de robots
- ▶ A través de la robótica transmitir a profesores y estudiantes conocimientos básicos sobre las nuevas tecnologías y sus aplicaciones
- ▶ Disminuir las asimetrías existentes entre liceos públicos y privados
- ▶ El proyecto fue financiado por la ANII y apoyado por la unidad de extensión de la Facultad de Ingeniería



Objetivos

Motivación

- ▶ Programar los comportamientos de un robot móvil genera mucho interés en los adolescentes y estimula la creatividad.
- ▶ Permite alcanzar resultados visuales inmediatos de sus programas
- ▶ Uruguay ha implantado el proyecto OLPC en todas las escuelas y gran parte de los liceos públicos del país.
- ▶ Software: GNU/Linux y Sugar. Hardware: Netbook de bajo consumo (computadora XO)

Objetivos

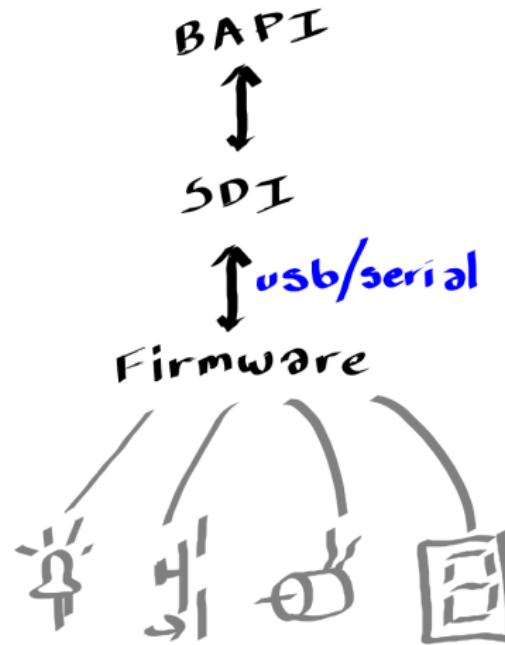
Transformando la XO en un robot móvil

- ▶ XO es el *cerebro* del robot
- ▶ Interacción con Hardware de la XO
 - ▶ Webcam
 - ▶ Micrófono
- ▶ Interacción con Software de la XO
 - ▶ Tortugarte
 - ▶ Python
- ▶ Interacción con sensores
 - ▶ Luz, escala de grises, distancia, vibración, campo magnético, inclinación, contacto, temperatura
- ▶ Interacción con actuadores
 - ▶ Led, motores, pinza, display

Características

- ▶ Portable a plataformas con bajas prestaciones de hardware (SBCs, smartphones)
- ▶ Enfoque modular
- ▶ Fácilmente extensible
- ▶ Arquitectura orientada a capas

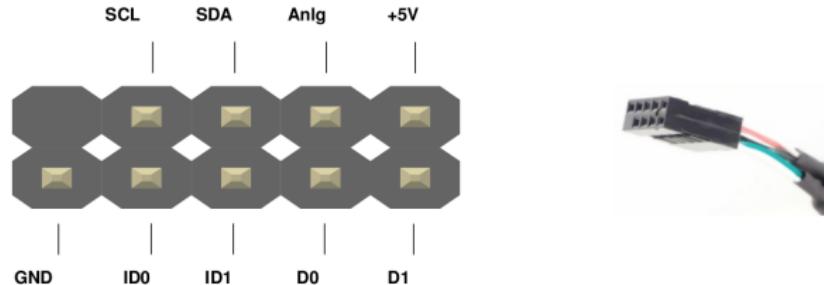
Diseño



Capa Firmware

- ▶ Encargada de la interacción con el Hardware
 - ▶ Sensores y Actuadores
- ▶ El hardware es abstraído como *módulos* en la programación del firmware
 - ▶ Motores, Luz, Grises, Distancia, entre otros
 - ▶ Existe una Interfaz que cada módulo debe cumplir
 - ▶ Permite una fácil extensión
- ▶ Funcionalidades exportadas en servicios de software
 - ▶ Motores:mover(sentido1, velocidad1, sentido2, velocidad2)
- ▶ Implementada en el microcontrolador utilizado (PIC/ATMEL AVR)

Capa Firmware - Plug & Play



- ▶ Cada conector posee pines de identificación
- ▶ El firmware identifica al sensor y le asigna un módulo
- ▶ La lista de módulos presentes en el hardware es notificada a la capa superior

Capa 2 - Service Discovery and Invocation

- ▶ Descubrimiento dinámico de módulos y servicios
- ▶ Invocación de servicios
- ▶ Brinda independencia de la placa de entrada/salida
 - ▶ Arduino
 - ▶ USB4all
 - ▶ GoGo board
- ▶ Brinda independencia de la tecnología de comunicación con la placa de entrada/salida
 - ▶ USB
 - ▶ Serial
 - ▶ Bluetooth



▶ Puede ser ejecutada en una XO/PC/SBC



CASE

Congreso Argentino de Sistemas Embebidos

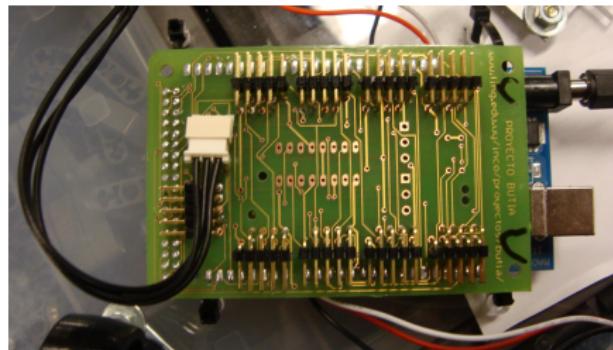
Capa3 - Butiá Application Programming Interface

- ▶ Brinda una interfaz de alto nivel para poder interactuar con los módulos
 - ▶ LIST
 - ▶ DESCRIBE moduleName
 - ▶ CALL moduleName operation param1, param2, ... , paramN
- ▶ Los clientes se implementan sobre esta capa utilizando dichas primitivas
 - ▶ Existe un cliente (bobot-server) que expone este protocolo en la red

Ventajas

- ▶ Fácil extensión
 - ▶ No es necesario preocuparse por aspectos relacionados con la comunicación o detalles de bajo nivel del firmware
 - ▶ Solo hay que centrarse en la lógica de control del sensor/actuador a controlar
- ▶ Mantenible
 - ▶ Comportamiento del robot implementado en lenguaje de alto nivel, en firmware, solo la lógica de control del sensor/actuador
- ▶ Autoconfigurable
 - ▶ Plug & Play y descubrimiento de módulos y servicios elimina etapa de configuración

Componentes de Hardware



- ▶ XO/PC/SBC
- ▶ Placa E/S basada en un microcontrolador
- ▶ Shield diseñado por el grupo
- ▶ Sensores/Actuadores



Implementación

Hardware Butiá v1.0

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 y sensores Sharp

Implementación

Chasis constructivo

- ▶ Cortes en láser sobre acrílico
- ▶ Superficie suficiente para llevar un netbook o XO
- ▶ Orificios para colocar los sensores o actuadores
- ▶ Barandas para proteger la XO
- ▶ Dos ruedas de acrílico las cuales están sujetas a los motores
- ▶ Dos ruedas locas, una de ellas amortiguada para poder sortear desniveles

Componentes de Software

- ▶ La Capa Firmware está implementada C/C++ dependiendo de los lenguajes de programación soportados por el microcontrolador
- ▶ La Capa Service Discovery and Invocation y Butiá Application Programming Interface están programados en Lua
 - ▶ Lua es un lenguaje liviano y portable que lo hace muy adecuado para sistemas embebidos

Implementación

Componentes de Software

- ▶ Los clientes pueden escribirse en Lua o en otro lenguaje mediante una conexión TCP/IP con bobot-server
- ▶ La comunicación con la placa de entrada/salida microcontrolada puede hacerse por USB o serial
 - ▶ Microcontrolador PIC18f4550 utiliza USB, comunicación utilizando binding de *libusb* (*lualibusb*) para Lua desarrollado por el grupo
 - ▶ Arduino utiliza conversor USB-serial, se utiliza biblioteca *serialcomm* implementada en C para comunicación serie

Los niños toman el control

Bloques Tortuga

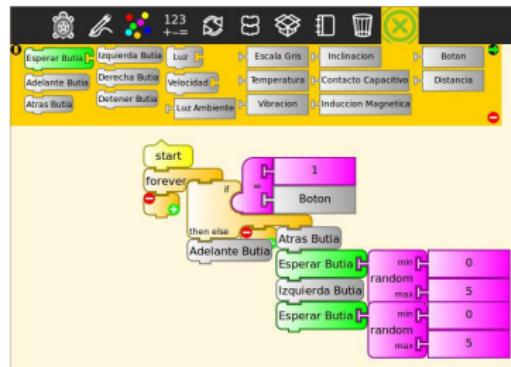
Paleta Tortuga Butiá



- ▶ Lenguaje de programación icónico inspirado en el lenguaje de programación Logo
- ▶ Cada sensor o actuador es mostrado como un bloque
- ▶ Plug & Play: Se colorean los bloques de la paleta Butiá si el sensor correspondiente esta conectado

Los niños toman el control

Programando Comportamientos



Programa para evitar obstáculos desarrollado en Tortuga



Pedro, un usuario de tan solo
9 años

Entrega de robots

Entrega, visitas y apoyo en los liceos

- ▶ En setiembre de 2010 se entregaron robots a 27 liceos seleccionados y se realizó capacitación y talleres para estudiantes y profesores (81 personas aproximadamente)
- ▶ Durante los meses de octubre y noviembre se realizaron visitas a los liceos.
- ▶ Se realizó capacitación y talleres para estudiantes y profesores (340 personas aproximadamente)
- ▶ Cada liceo tiene un estudiante universitario como referente el cual le brinda apoyo y soporte

...Butiá 2.0

Trabajo futuro

- ▶ Disminuir costos.
- ▶ Ser más eficientes con el uso de los pines para identificación
- ▶ Culminar la implementación de plug & play y shield para placa USB4all (microcontrolador PIC)
- ▶ Mejorar aspectos constructivos

finalmente...

Conclusiones

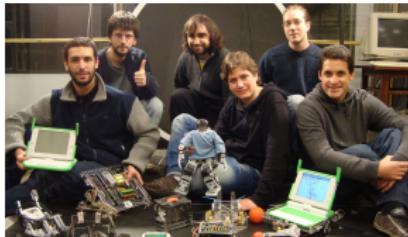
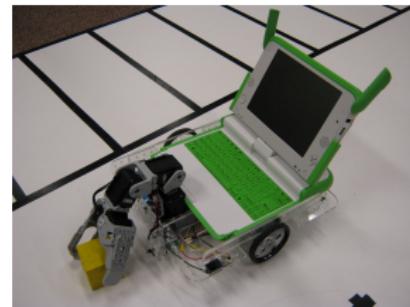
- ▶ Se realizó un prototipo de robot, constructivo, totalmente integrado a la computadora utilizada por OLPC y su sistema SUGAR
- ▶ Se implantó en 27 liceos del país
- ▶ Pudo validarse que niños sin conocimientos previos de programación pudieron implementar fácilmente el comportamiento de robots móviles

Más que mil palabras



Gracias por su tiempo

¿Preguntas?



CASE

Congreso Argentino de Sistemas Embebidos

Gracias por su tiempo

Butiá: Plataforma robótica genérica para la enseñanza de la informática

Gonzalo Tejera Andrés Aguirre

Federico Andrade Pablo Gindel Santiago Margni
Guillermo Reisch Jorge Visca

Instituto de Computación, Facultad de Ingeniería, Universidad de la República
J. Herrera y Reissig 565, Montevideo, Uruguay
<http://www.fing.edu.uy/inco/proyectos/butia>
butia@fing.edu.uy



4/03/2011



CASE

Congreso Argentino de Sistemas Embebidos