Plataforma robótica genérica y constructiva para la enseñansa de la informática

Gonzalo Tejera, Andrés Aguirre, Federico Andrade, Pablo Guindel, Santiago Margini, and Jorge Visca

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

{gtejera|aaguirre|fandrade|pablod|smargni|jvisca}@fing.edu.uy

http://www.fing.edu.uy/inco/grupos/mina

Abstract. El aprendisaje de la robótica en los niveles iniciales de la educación es una herramienta poderosa para trasmitir a los profesores, estudiantes y sus familias conocimientos básicos sobre las nuevas tecnologías y sus aplicaciones. Existen muchos mitos sobre las computadoras y los robots, desconocimientos básicos tanto sobre lo que pueden como lo que no pueden hacer, en ambos sentidos, y que genera por un lado miedos infundados y por otro expectativas desmedidas; que la incorporación de los robots y de la inteligencia computacional se est´ dando de manera progresiva en nuestra sociedad, y es importante entonces contribuir a mejorar el conocimiento de estas tecnologías. Para esto es necesario contruir una plataforma simple y económica que permita a los alumnos de liceos públicos en coordinación con los docentes, interiorizarse con la programación del comportamiento de robots.

Keywords*: educación, robotica, computadora XO*

1 Introducción

Este trabajo analiza el área de aprendisaje informático a travez de la robótica. Este trabajo esta organizado de la siguiente manera. Luego en la Sección 4 se describen detalles del prototipo implementado. Resultados experimentales también seran presentados. Diferentes lineas de trabajo y perspectivas al área son sugeridas en la Sección 5 y el trabajo es concluodo en la Sección Section 6.

2 La robótica y la educación

Chachara inicial, haciendo referencia a las ideas de stallman de las bondades de poder compartir, modificar libremente el cdigo y las ideas de walter bender de llevar a un nivel de entendimiento y abstraccin mayor para que el cdigo pueda ser modificado y entendido por nios y adolecentes que se inician en la informática

2.1 El software libre y la educación

2.2 Plataformas existentes

3 Arquitectura del la plataforma butiá

In Section 2 we studied ....

3.1 Plataforma mecánica constructiva

Santi completame :D

3.2 Conectores Plug and Play

la magia del plug and play a nivel abstracto y de los conectores, luego en implementación explicamos como se hizo para arduino, Palmera completame

3.3 Servidor Bobot

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

- OLPC XO. Este es la plataforma del Plan Ceibal. El hardware contiene un procesador AMD Geode, con arquitectura x86. Los primeros modelos posean 256Mb RAM. El disco duro es de estado sólido. El software consiste en un kernel Linux con Sugar como interfaz de usuario.
- Intel Classmate. Es la plataforma para enseanza de Intel, y es un nombre genérico para una lnea 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.
- Router Inalámbrico. Plataforma de costo y poder de cómputo m ínimo prevista. Un ejemplo tpico es un router Asus 520GU. Conssite en un sistema embebido con un procesador Broadcomm, con 32Mb de RAM y 8Mb de almacenamiento Flash. En nuestro uso, se le instala OpenWRT, una distribución de Linux para sistemas embebidos.
- 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 inters, y que implemente las siguientes funcionalidades:

- Acceda a la placa del Microprocesador implementando el protocolo *USB4all* sobre el tipo de conexión asociado (USB o Serial)

– Ofrezca una API que permita acceder las funcionalidades provistas por Buti desde las aplicaciones de usuario.

Este componente se implement en dos partes: una librera que implementa la comunicacin con la placa (*bobot*, y una aplicacin que usa esta librera y exporta su funcionalidad mediante un socket (*bobot-server*). Esta solucin es la solucin de referencia, de mxima portabilidad. Para plataformas especficas es perfectamente posible desarrollar soluciones particulares.

Este componente se implement en Lua. Este es un lenguaje de scripting dinmico basado en una mquina virtual. Est escrito en ANSI C99 por lo que es altamente portable, y es muy compacto: la mquina virtual ocupa unos 800kb de RAM. Es un lenguaje muy eficiente y provee varias funcionalidades potentes, tales como expresiones regulares, mapas asociativos basados en hash, funciones como miembros de primera clase, etc.

bobot Como se mencion, esta librera accede la placa microcontroladora mediante USB o Serial. Para lo primer opción, se enlaza con libusb, una librera portable de espacio de usuario para manipular dispositivos USB. Este enlace se realiza mediante un binding desarrollado, llamado *lualibusb*. Para acceder mediante serial se utiliza una pequea librera en C que implementa una comunicacin orientada a mensajes sobre Serial llamada *serialcomm*. Esta librera es fcilmente extensible para agregar soporte a nuevos mdulos de la placa controladora (*USB4all/Arduinoi*). Esto se logra mediante drivers cargados dinmicamente de acuerdo a los mdulos declarados por la placa controladora. Para agregar soporte para un nuevo mdulo, basta con agregar el driver apropiado a la directorio de drivers.

bobot-server Esta aplicación exporta la funcionalidad subyacente de Butiá a las aplicaciones de usuario. Lo hace mediante un protocolo muy sencillo en un socket TCP (por defecto en el puerto 2009). Para utilizarlo las aplicaciones de usuario deben abrir una conexión TCP, e implementar dicho protocolo. Los mensajes soportados por bobot-server son:

`LIST` Devuelve la lista de módulos detectados, separados por comas.

`OPEN module [ep1 ep2]` Abre el módulo. Si es necesario o no, depende del modulo. Devuelve "ok" o "fail".

`DESCRIBE module [ep1 ep2]` Devuelve una descripción de la API del modulo.

`CALL module function [parameters ...]` Invoca la función indicada en el módulo dado. Los parámetros dependen de la función. Devuelve la respuesta de la llamada, o "fail".

`CLOSEALL` Cierra todos los módulos. En general no hace falta. Devuelve "ok"

3.4 Extendiendo los servicios del Robot

El mecanismo por el cual se pueden extender las funcionalidades del sistema es mediante la implementación de nuevos módulos en el firmware como se mencionó

en la sección 3.3, las funcionalidades de estos módulos son exportados como servicios por el servidor *bobot*, esto es gracias a la existencia de drivers en el sistema *butiá*, los cuales se encargan de implementar el protocolo de comunicación entre el módulo y el *bobot*, de esta forma se encapsula dentro del servidor, en cada uno de los drivers los detalles de bajo nivel exportando al usuario primitivas de mayor nivel de abstracción. Dentro de un driver se debe de especificar cada uno de los servicios exportados, los parámetros requeridos, su tipo y tipo y valor de retorno. Los drivers son cargados de forma dinámica por el *bobot-server* el cual obtiene que funciones implementa y cuales son sus parámetros de entrada y valores de retorno. Este tipo de enfoque permite que el programador se centre en los detalles de la lógica de control del robot sin preocuparse de los aspectos de bajo nivel que refieren a particularidades de los sensores o actuadores utilizados. Lo cual redunda en lógica de control independiente del hardware, tanto de los sensores/actuadores utilizados como de la placa controladora de entrada/salida utilizada. Estos aspectos afectan de forma directa y positivamente la mantenibilidad del código.

4 Implementación del prototipo para la computadora XO

In Section 3.1 bla bla bla.

4.1 Implementation details

En la sección 3.3pudimos En esta seccin hablamos de los detalles de implementacin, desde el AX12 hasta particularidades en arduino Ejemplo de cita bibliográfica[**?**]. Ejemplo de inserción de figura: Figure 1

4.2 Prototipo en tortugarte

en esta sección se escribe acerca de tortugarte

5 Observaciones finales y trabajo futuro

Como vamos a seguir currando con el butiá

6 Conclusiones

Como curramos con el butiá y la robótica....

References

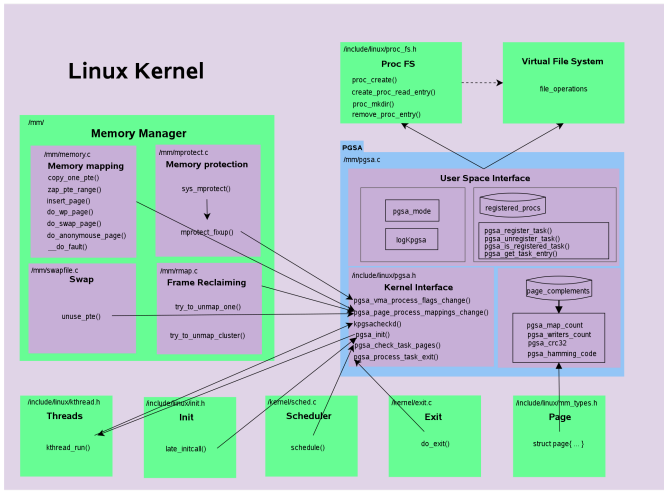

Fig. 1. Changes introduced to Linux kernel