

Plataforma Robótica de Bajo Costo GNU con Visión Local

Andrés Aguirre Jorge Visca
{aaguirre, jvisca}@fing.edu.uy

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

VI Workshop de Inteligencia Artificial y Robótica

Agenda

- 1 Introducción
 - Motivación
 - Requerimientos de la plataforma

- 2 Arquitectura Propuesta
 - Arquitectura del Hardware



Objetivos

- Construcción de un robot prototipo con visión local.
- Utilización en trabajos de investigación y de grado.
- Arquitectura extensible y genérica.
- Autónomo y con capacidad de procesamiento distribuido.
- Alternativa económica para plataformas comerciales.



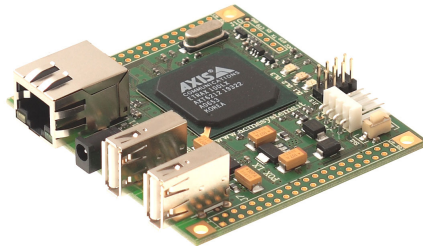
Características

- Abierta y liberada con licencia GNU.
- Económica y posible de construir con componentes existentes en el mercado.
- Independiente del hardware utilizado.
- De fácil configuración/plug and play.
- Comunicación inalámbrica.



Nuevas Funcionalidades en Dispositivos Embebidos

- Aumento en funcionalidades de dispositivos embebidos.
- Utilización de sistemas operativos estándar para afrontar nuevas funcionalidades.
- GNU/Linux una opción frecuente.
 - Extensiones de tiempo real incluidas en el Kernel.
 - Permite reutilizar conocimiento ya existente y librerías desarrolladas.
 - Sistema operativo preferido a la hora de trabajar con SBC.



Ventajas para el desarrollo de robots

- Facilita la construcción de robot autónomo.
- Permite utilizar herramientas de desarrollo de mayor nivel de abstracción.
- Sin costo de licencias de software.
- posibilidad de migrar a muchas plataformas de hardware (x86, MIPS, ARM, uc sin MMU).
- Una comunidad abierta.
- Reutilización de bibliotecas de software.
 - A nivel del Sistema Operativo (stack tcp/ip, drivers para dispositivos USB, etc).
 - A nivel de aplicaciones de usuario Permite reutilizar conocimiento ya existente y librerías desarrolladas (Vision por computadora, reconocimiento del habla, etc).
- Reutilización de dispositivos Hardware.
 - Utilizar un sistema operativo permite tener un gran soporte de periféricos.
 - Alternativa económica.



Interfaz Homogenea con Dispositivos

Utilización del BUS USB

- La arquitectura de hardware se basa en la utilización del BUS USB para la comunicación entre la placa embebida y los dispositivos sensores/actuadores.
- Existe una gran cantidad de dispositivos (COTS) con esta interfaz que pueden utilizarse para la robótica.
 - GPS
 - Dongle Bluetooth.
 - Tarjeta de audio.
 - Modem 3G.
 - Almacenamiento externo.
 - etc.
- Bajo costo.
- Fomenta un enfoque constructivo.



Independencia de la Placa Controladora

- La placa controladora puede ser una mother PC, celular/palm, SBC, etc (cualquier plataforma compatible con el sistema operativo sistema GNU/Linux)
- Se desacopla la placa controladora de los puertos de E/S particulares de cada plataforma.
- Utilización de controladora de E/S con interfaz USB para interactuar con sensores.
 - Permite agregar procesamiento de bajo nivel en la controladora permitiendo ser más que simplemente una pasarela
- Al no estar atados a mecanismos de E/S particulares de la placa controladora, nos va a permitir poder cambiar de plataforma sin necesidad de modificar el software.



Lenguaje de Programación Seleccionado

LUA

- Portabilidad.
- Funciona con hardware de bajas prestaciones.
- Lenguaje de alto nivel de abstracción
- Interpretado (evita cross compilar).
- Permite interactuar fácilmente con código C.



Descubriendo las Capacidades Expuestas por la Placa de E/S

- El robot debe aprender que tipo de sensores y actuadores dispone y cual es su interfaz.
- Esta información cargada dinámicamente en el robot a partir de metadatos almacenados en el firmware de la placa de E/S.
- Interacción entre el firmware y el software del robot para cumplir el requisito.



Compartiendo Servicios Entre Agentes

- No todos los robots tienen porque tener los mismos sensores/actuadores.
 - Esto va a condicionar sus capacidades.
- Mecanismo que permita publicar las capacidades de cada robot como servicios que puedan consumir otros robots.

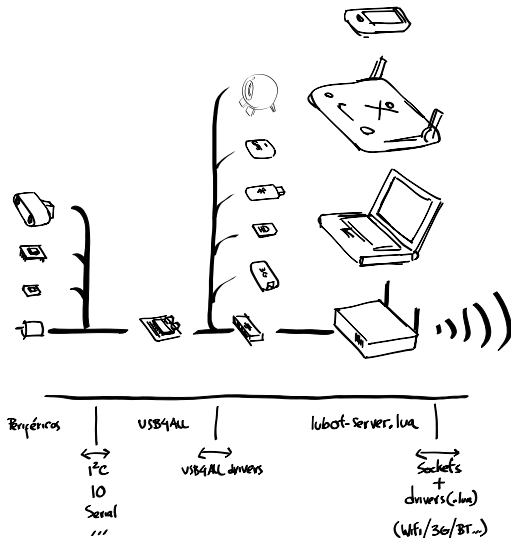


Funcionamiento General 1/2

- Cada agente pose un cliente y un servidor.
- El servidor se encarga de exportar los servicios.
- El cliente consume sus servicios y los de otros agentes.
- Interacción entre el cliente y el servidor mediante Sockets.



Funcionamiento General 2/2



Construcción de la Estructura



Dispositivos Presentes

- Placa controladora USB4all.
 - Control de aceleración de motores steppers.
 - Acelerómetro
- USBAudio.
- Almacenamiento externo mediante pendrive.
- Modem 3G.
- WebCam USB 2 MP.
- Placa controladora para permitir la interacción entre todos los dispositivos conectados.



Placa Controladora Utilizada

- Dada su gran disponibilidad en el mercado se opto por utilizar la placa de un router doméstico.
- Procesador MIPS.
- 32 MB RAM.
- Puerto USB como Host.
- Placa de red inalambrica Atheros compatible.



Sistema Operativo

- Se utilizó el sistema operativo para dispositivos embebidos OpenWrt.
- A diferencia de otros sistemas operativos basados en Linux, se posee de todos los fuentes, lo que nos permite tener independencia tecnológica.
- Repositorio y manejador de paquetes disponible.
- Herramienta de cross compilación.



Situación Actual



Conclusiones

- Superada la prueba de factibilidad.
- Integración con COTS (siempre que sea posible) en lugar de desarrollar dispositivos desde cero.
- Arquitectura que le permite al robot aprender dinámicamente los sensores/actuadores que dispone y publicarlos para que sean utilizados por otros robots.



Trabajo a Futuro

- Anexar más sensores a la placa de E/S.
- Integrar con Lenguajes gráficos, Xlogo, Scratch.
- Desarrollo de una solución a un problema distribuido conocido utilizando la arquitectura propuesta.
- Utilizar el protocolo XMMP como transporte para la notificación de los servicios expuestos y utilización.
- Integración con software de reconocimiento de imágenes.



¿Preguntas?

