Linux como plataforma de desarrollo profesional con Microprocesadores de 8 bits

Presentación para el 5º Encuentro Linux 2004

Ricardo Albarracín B.
Ingeniero de Diseño de Sistemas Digitales
email: rab@electrolinux.cl, rab@cdsl.cl

Valparaíso 21 de Octubre del 2004

Resumen de temas abordados

- Situación presente
- Etapas generales del desarrollo digital
- Herramientas de desarrollo bajo open-source y freeware.
- Requerimientos de PC para el desarrollo bajo Linux
- Microprocesadores posibles de uso para de-sarrollo.
- Como partimos con los desarrollos.
- Ejemplos de programación simples.
- Futuros desarrollos.
- Aplicaciones en robótica.

Situación presente

- Que herramientas de desarrollo conocemos?
- Actualmente la gran mayoría de las aplicaciones de desarrollo de sistemas digitales son bajo otra plataforma de uso masivo, las cuales tienen un alto costo (la mayoría), un soporte bajo (algunas) y además una alta latencia en sus actualizaciones y mejoras. Algunas de ellas son Orcad, Tango, Smartworks, Protel, Workbench, Keil, MicroC y varias otras, al menos hay una gran fauna de ellas.
- Sus upgrade son casi siempre con costos que son un porcentaje del costo de licenciamiento.
- Que pasa con estas políticas?...tienden al uso sin licenciamiento.

Etapas generales del desarrollo digital

1. Supongamos que ya existe la necesidad, el estudio de factibilidad, el pre-diseño del sistema, y su financiamiento.
2. O sólo queremos hacerlo por hobby o investigación personal.... si, es factible y posible, además sin mucha inversión.

- Como ya sabemos que hacer, debemos poner manos a la obra, con al menos, las etapas siguientes:
- Etapa del “Diseño Esquemático”...(diagrama eléctrico).
- Etapa del “Diseño del Circuito Impreso o PCB”.
- Armado y montaje de partes.
- Programación del Sistema.

Herramientas de desarrollo bajo Open-Source

- Listado muy completo de las aplicaciones bajo Linux se encuentra en http://www.metamachine.com/~owen/eda.html
- o buscar simplemente en www.google.cl usando “EDA for linux”
- Podemos mencionar gEDA, Oregano, PCB, Eagle y muchas otras.
- Compiladores

GCC http://gcc.gnu.org

SDCC http://sdcc.sourceforge.net/snap.php

Herramienta integrada de desarrollo

- Como hemos visto, herramientas open-source, hay variadas y en desarrollo, no están totalmente integradas en una misma aplicación, pero se pueden usar sin problema.
- Para desarrollo profesional la que uso es Eagle que es una aplicación completa para desarrollo no open-source, la que tiene; un modulo para hacer el circuito eléctrico completo, otro para realizar el diseño del PCB y algo que ahorra bastante trabajo el autorouter, sin embargo tiene una versión freeware que permite hacer placas de hasta 10x8cm, lo que es mas que suficiente para pequeños proyectos. La versión profesional no es freeware y su costo esta del orden de los ~US$1300 FOB para Linux.

Su URL es http://www.cadsoftusa.com

Requerimientos de una maquina para desarrollo en Linux

- Nada especial.....
- Procesador desde 500MHz o superior, Celeron o Pentium en adelante, arquitectura i686 recomendable, los i586 funcionan pero muy lentos.
- RAM lo más posible... siempre es recomendable, pero a partir de los 128MiB funciona (y menos... pero lento).
- Capacidad de HDD.... 3GiB hacia arriba.
- La tarjeta gráfica PCI o en lo posible AGP con 4MiB de RAM hacia arriba.
- Tarjeta de red y conexión internet en lo posible..
- Es decir, cualquier PC actual, con hardware apropiado.

Microprocesadores posibles para desarrollar de 8 y 16 bits

Hay varios y tal vez más de lo que uno podría usar algunos son:

En 8 bits:

CISC familia MCS51/52, Z80 y Z180, DS80C390, DS80C400 basados en el MCS51 pero de alto rendimiento, Serie PIC que son RISC P16F84 de 14 bits, para el Motorola HC08, Toshiba TLCS-900H y Phillips XA51 (variante del MCS51), serie MCS51/52/AVR de Atmel.

En 16 bits:

Motorola Familia HC12 (variante del HC11 mejorado), serie nueva de µC y µP, con memoria Flash.

PIC P18F452 y sus variantes.

Como partimos con los desarrollos

Veremos algunos aspectos generales en el uso de Eagle, con un ejemplo practico.

- Como lo indicamos, hacemos primero el circuito esquemático.
- Realizamos el diseño del PCB o Circuito Impreso.
- Vemos al Autorouter en forma general y algunos consejos en su uso.

Compiladores disponibles SDCC y GCC

Para compiladores C, tenemos dos posibilidades disponibles y lo mejor es que son completamente open-source.

- El compilador Small Device C Compiler, es libre, reorientable, ANSI-C optimizado de Sandeep Dutta, diseñado para Microprocesadores de 8 bits.
- El compilador por excelencia en Linux GCC, el cual tiene muchas posibilidades en su uso.

Recomiendo usar los manuales de ambos compiladores.

Ambos compiladores generan código de muy buena calidad comparable a otros compiladores de alto costo para otras plataformas.

Datos Generales de algunos test de los Compiladores

Para poder dimensionar y tener alguna idea de lo que estamos hablando podemos indicar algunas pruebas:

- Debo indicar que no he sido muy riguroso en los test, pero lo que he realizado me da una idea más que satisfactoria (por supuesto con el mismo hardware).
- He probado un mismo código para el MicroC de Dave Dunfield y SDCC con ventajas para este, dependiendo del tipo de rutina, (1) difieren en eficiencia en el assembler generado en forma importante, (2) en tamaño de 1 a 2/2,3 veces y (3) en rendimiento por consecuencia.
- El compilador que genera mejor código que el SDCC es el Keil, pero su costo es importante.

Como aporta el open-source a la Empresa y a los desarrolladores

- Disponer de aplicaciones de una alta calidad técnica, lo que no es barato en otras plataformas.
- Uso de aplicaciones con una constante evolución y mejoras en su rendimiento, sobre todo en áreas especificas
- Posibilidad de participar en proyectos de desarrollos internacionales, eso permite el constante mejoramiento de sus profesionales y como consecuencia su ganancia en productividad interna.
- Cuando los aportes son significativos, hay publicación en muchos foros y paginas de los aportes de la empresa al proyecto en particular.
- Rebaja importante en los costos de implementación.

Fondo Económico Mundial FEM

- Por todos es sabido que en FEM ha calificado a Chile en el lugar 22, muy por sobre los países latinoamericanos, el más cercano es Mejico en el lugar 48, sin embargo esto contrasta violentamente con las calificaciones en aspectos tan relevantes como la educación y el desarrollo tecnológico de nuestro país.
- Lo anterior no es mas que una confirmación a que seguimos siendo un país colonial tecnológicamente hablando, ya que estamos acostumbrados a importar toda la tecnología y dejamos pocos espacios para nuestros desarrollos.
- Aspectos tan importantes, sólo vemos esfuerzos personales o en algunos casos empresariales pero muy modestos.
- En estos aspecto el open-source y particularmente linux/bsd pueden hacer bastante a desarrollar estas falencias de país.

Instalando el Compilador SDCC

Bajamos de la URL http://sdcc.sourceforge.net la versión actual del compilador sdcc-2.40, su instalación es a la forma tradicional a los archivos tar.gz

- Al bajarlos realizamos:

tar xvfz sdcc-2.4.0.tar.gz --> cd sdcc --> en el directorio que se ha creado con el tar.gz

```
[user@maquina]$ ./configure

[user@maquina]$ make

[root@maquina]# make install
```

Probando el compilador SDCC

Una vez instalado procedemos a probar que este bien instalado, para ello ejecutamos.

```
[user@maquina]$ sdcc -v
```

Esto nos debe entregar el siguiente mensaje (en una linea):

```
SDCC :
mcs51/gbz80/z80/avr/ds390/pic16/pic14/TININative/xa5
1/ds400/hc08 2.4.0 (Aug 13 2004) (UNIX)
```

Felicitaciones!!....(con cierto escandalo por cierto), lo hemos instalado bien.

Ejemplos muy simples de Programación en lenguaje C

1. Vemos el programa led.c, que es un pequeño ejemplo de encendido de un led con retardo de tiempos controlados por una interrupción.
2. Compilaremos el programa y veremos algunos detalles del Makefile, el código fuente, los archivos que se generan y sus definiciones generales.

Otras herramientas open-source de ayuda al desarrollo y trabajo en equipo

- Una de las principales herramientas de una gran utilidad es el uso de CVS (Concurrent Version System).

Ver articulo de Franco Catrin en:

http://www.tuxpan.com/fcatrin/files/cvs.html

Además siempre buscaremos en Google, usaremos una buena dosis de RTFM y finalmente aprenderemos a preguntar en forma inteligente en las listas de discusión, sobre Linux.

Cual es el problema que nos encontraremos ahora?

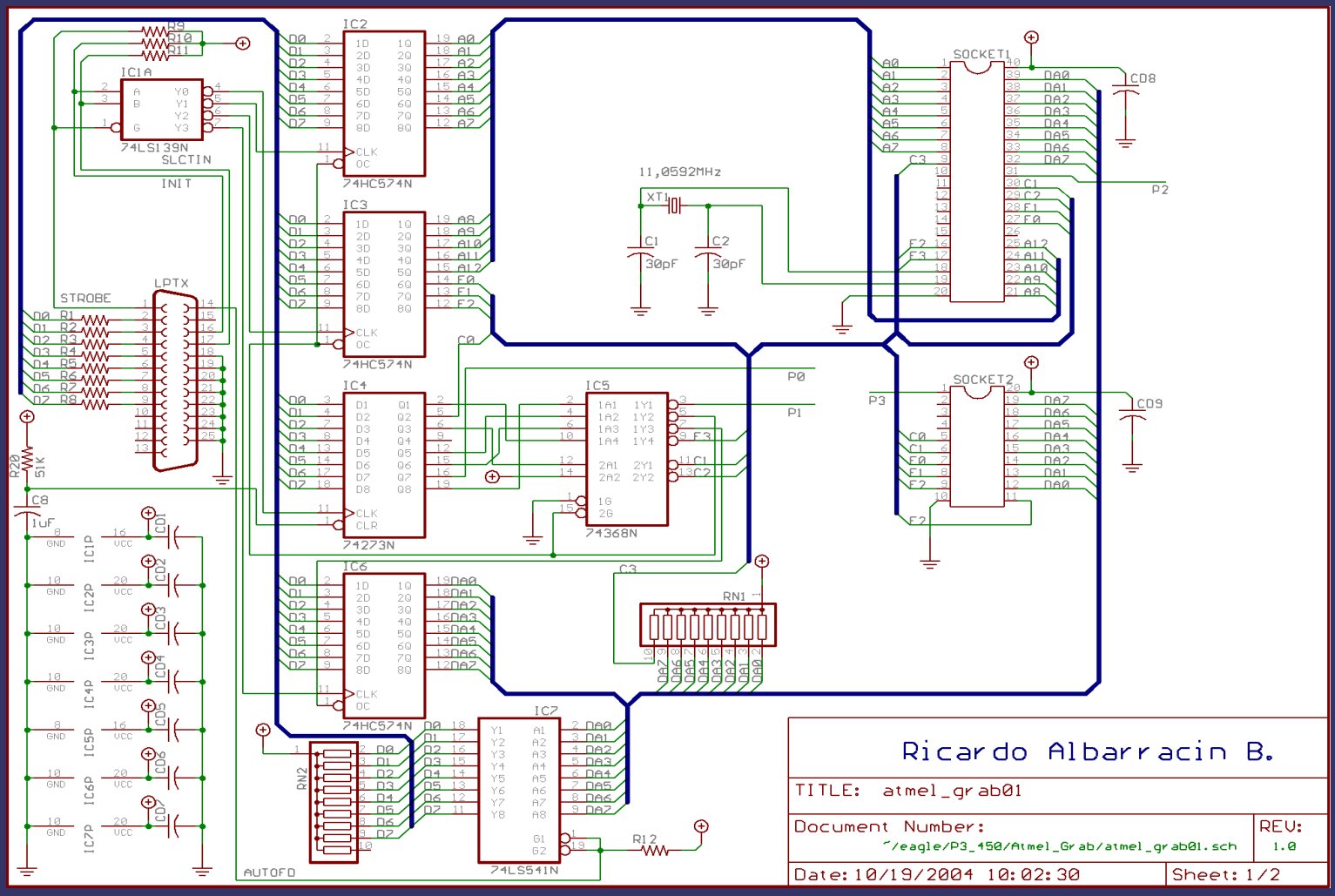
- Hasta ahora, hemos hecho el circuito esquemático con nuestro Eagle, hemos diseñado el PCB, lo hemos mandado a fabricar... cierto?... y nos ha llegado nuestra flamante placa impresa para montar nuestros componentes... que después de mucho meditarlo, lo hemos hecho.
- Además hemos bajado los compiladores y lo hemos instalado con cierto escandalo... :-), hemos realizado nuestra primera aplicación y tenemos el código listo.
- Ahora como diantres programo el Microcontrolador!

Programación del µControlador

- Tradicionalmente debemos tener un programador de microcontroladores y microprocesadores, que es un hardware que se conecta a un PC a la puerta paralela habitualmente y es manejado por una aplicación... adivinen donde corre.... si..... en esa cosa. (no lo repitan por favor!)
- Ahora tenemos todo... excepto el como programar nuestro µC. Pero estimado asistentes no deses-peréis TUX tiene solución para casi todo... y lo que no... lo inventa.
- Hay una aplicación open-source que permite pro-gramar a los µC, no todos eso si... pero sirve, esta es la “uisp” en http://www.amelek.gda.pl/avr/uisp
- Como resolvemos esto?

Grabador para µP MCS51

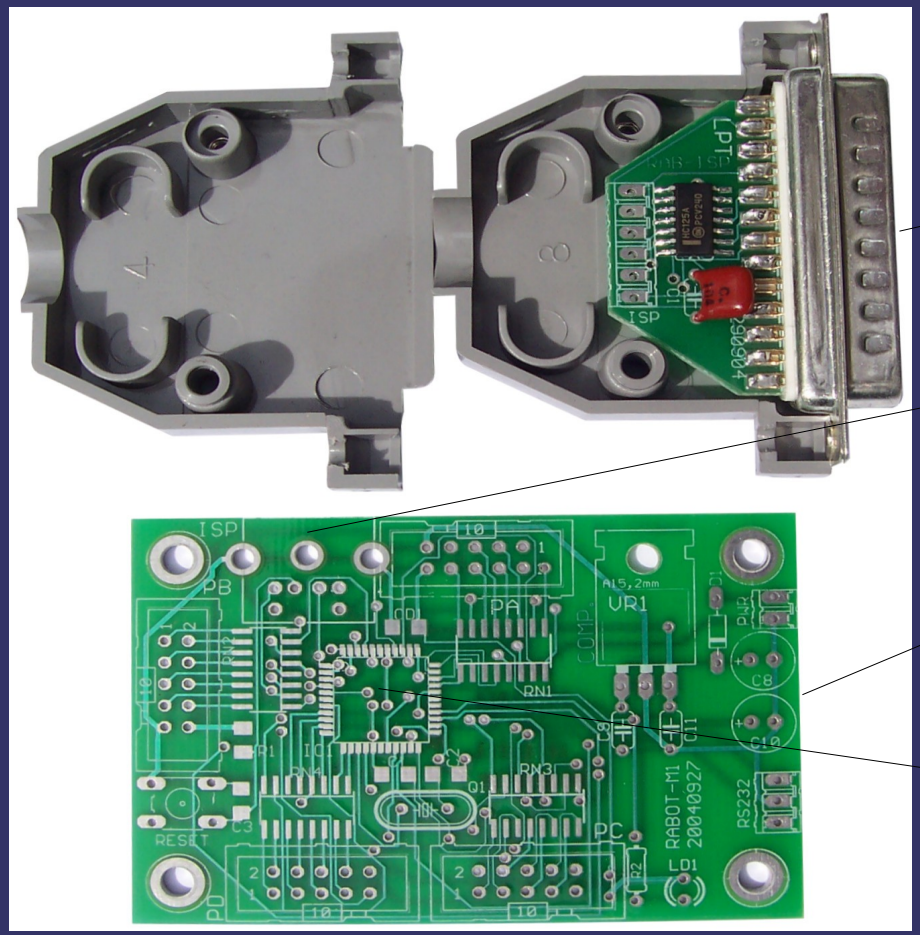
- Circuito del Grabador MCS51, en desarrollo.



Estoy reclutando programadores para hacer una aplicación Gnome/KDE

Sistema de desarrollo completo para Microprocesadores y Microcontroladores

- Sistema de desarrollo, desde el programador hasta el Microcontrolador, fabricado localmente... si... en Chile



Programador ISP

Puerta In_System_Programming

Sistema completo (PCB)

Microcontrolador RISC

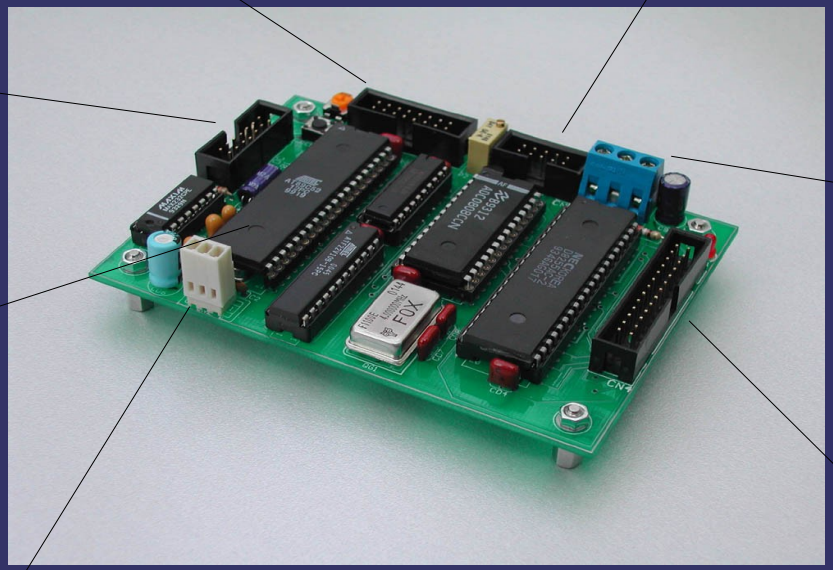
µControlador completo

- Sistema completo de Control

Puerta de LCD 2x16

8 canales análogos

8 lineas I/O



Alimentación
+12V/GND/+5V

µC MCS51

32 lineas I/O

Puerta RS232C

Otros desarrollos realizados con estas Herramientas

- µC pequeño, familia MCS51.

Salidas

RS422

RS232

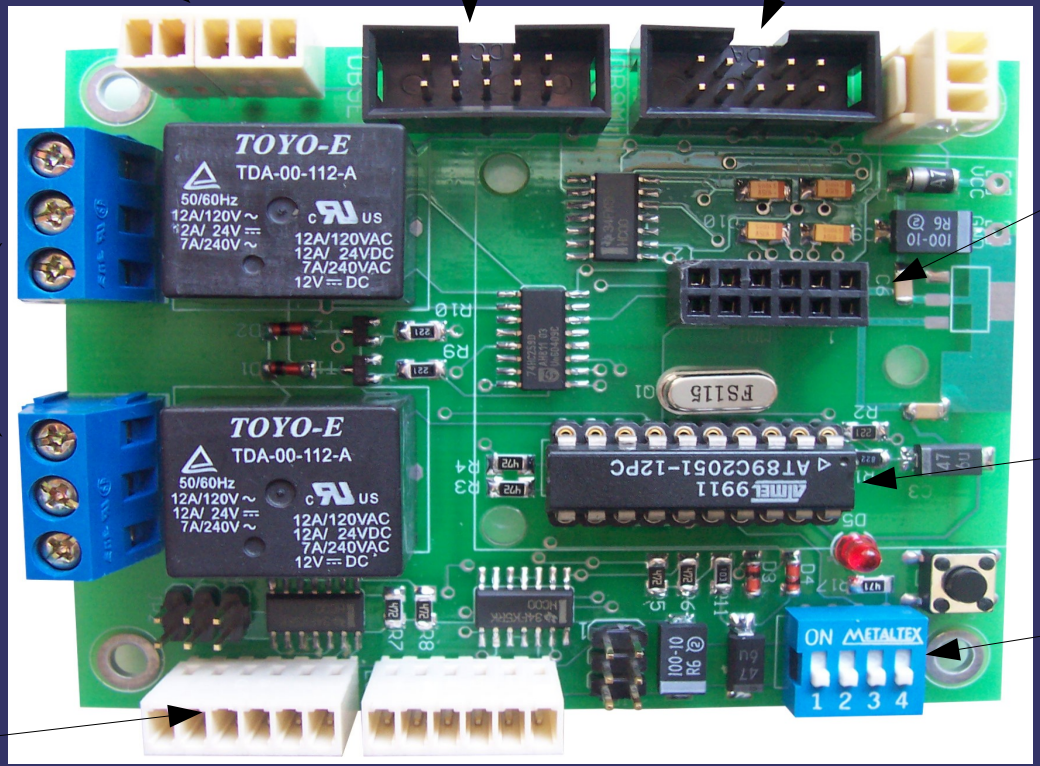
Puerta para TCP/IP

Actuadores contacto seco

µC

Switch add

Entradas



El hardware que “habla”

- Este es un prototipo por supuesto pero funciona.

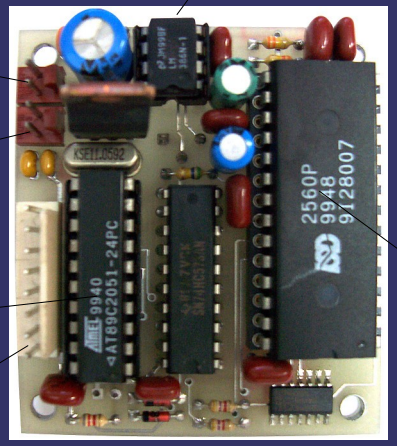
Amplificador

Salida Parlante

Alimentación 9V

CPU 8 bits

Entradas digitales

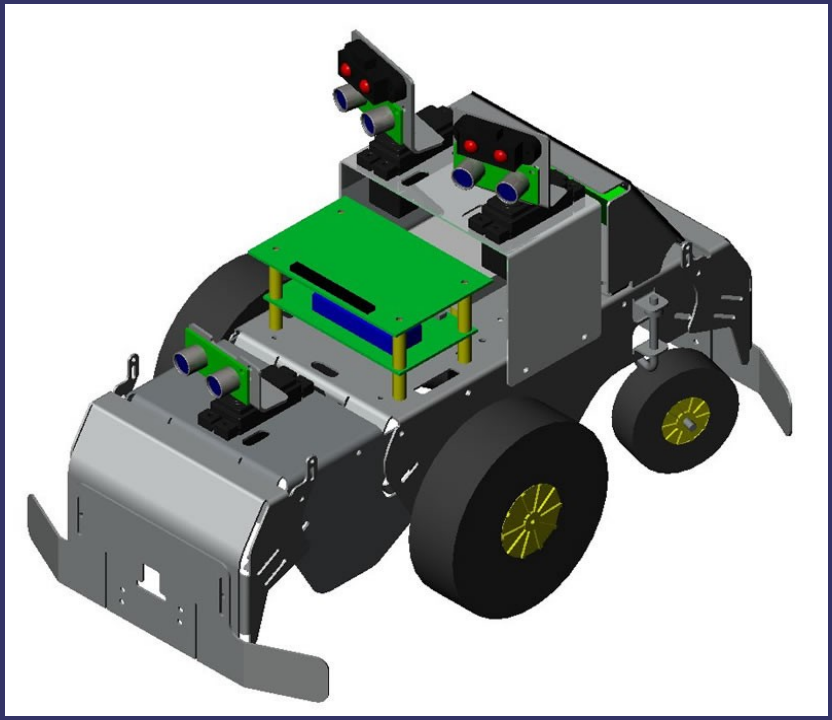


Grabador electrónico de audio

Tamaño de 5,1x5,6cm

Aplicaciones en Robótica y hobby's

- Este es un esqueleto de un robot móvil, al cual se le pueden agregar diferentes sensores, radar, sonido, etc...



Este es un prototipo de desarrollo y se esta realizando el plano mecánico, para ser hecho en corte láser.

Sus motores serán controlados por manejo de PWM independientes para cada costado de tracción.

Modelo sacado de http://www.superdroidrobots.com

Eso es todo amigos

Espero ahora sus...

- Quejas, criticas, recomendaciones, alcances, sugerencias, comentarios, reclamos, manifestaciones, etc...
- Además claro esta de sus dudas y preguntas, las que tratare de responder con el mayor agrado.

Los espero en mi pagina web

www.electrolinux.cl