

***Linux como plataforma de desarrollo
profesional con
Microprocesadores de 8/16/32 bits***

Presentación para Encuentro Linux 2005
Universidad Autónoma del Sur
Talca

Ricardo Albarracín B.
Ingeniero de Diseño de Sistemas Digitales
email: rab@eletrolinux.cl
<http://www.electrolinux.cl>

Talca 11 de Noviembre del 2005



Resumen de temas abordados

- En presente
- Aspectos generales del desarrollo digital
- Desarrollo bajo open-source
- Herramientas de PC para el desarrollo bajo
- Escenarios posibles de uso para desarrollo.
- Como partimos con los desarrollos.
- Ejemplos de programación simples.
- Futuros desarrollos.
- Aplicaciones en robótica.

tuación presente

- 
- Que sistemas de desarrollo conocemos?
 - Actualmente la mayoría de las aplicaciones de desarrollo de sistemas digitales son bajo otras licencias masivo, las cuales tienen un alto costo, un soporte bajo (algunas) y poca atención en sus actualizaciones y mantenimiento. Algunas de ellas son Orcad, Tango, Protel Workbench, Keil, MicroC y varias otras. 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.



generales del desarrollo digital

1 que ya existe la necesidad, el
factibilidad, el prediseño del
u financiamiento.

eremos hacerlo por hobby o
personal.... si, es factible y
mucho inversión.

que hacer, debemos poner manos a
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.



Instrumentas de desarrollo bajo Open-Source

muy completo de las aplicaciones
linux se encuentra en
metamachine.com/~owen/eda.html

plemente en www.google.cl EDA

onar gEDA, Oregano, PCB, Eagle

as.

➤ **Compiladores**

GCC <http://gcc.gnu.org>

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

Plataforma integrada de desarrollo



- Con herramientas open-source, hay variadas y no están totalmente integradas en una pero se pueden usar sin problema.
- Profesional la que uso es Eagle que es completa para desarrollo no open-source, para hacer el circuito eléctrico realizar el diseño del PCB y algo de 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>

Requisitos de una maquina para desarrollo en Linux



- ➔ Necesitas un procesador de 500MHz o superior, Celeron o Pentium
- ➔ Procesador de arquitectura i686 recomendable, los i586 son lentos.
- ➔ Memoria RAM... siempre es recomendable, pero a menos funciona (y menos... pero lento).
- ➔ Cantidad de memoria... 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.

procesadores posibles para desarrollar de 8 y 16 bits



Hay varios de lo que uno podría usar.... algunos

En 8 bits: Intel 8051, Z80 y Z180, DS80C390, DS80C400, 8051 pero de alto rendimiento, Serie 80C184 de 14 bits, para el Motorola 6809, 6800H y Phillips XA51 (variante del 6801), serie 8051/52/AVR de Atmel.

En 16 bits:

Motorola Familia HC12 (variante del HC11 mejorado), serie nueva de μ C y μ P, con memoria Flash.

PIC P18F452 y sus variantes.



o partimos con los desarrollos

Vemos los aspectos generales en el
con un ejemplo practico.

lucamos, hacemos primero el
emático.

el diseño del PCB o Circuito
Impreso.

- Vemos al Autorouter en forma general y algunos consejos en su uso.



Compiladores disponibles *SDCC* y *GCC*

Compiladores C, tenemos dos posibilidades
pero lo mejor es que son completamente

Small Device C Compiler, es libre,
SI-C optimizado de Sandeep Dutta,
para procesadores de 8 bits.

Recomiendo 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.

Generales de algunos test de los Compiladores



Dimensionar y tener alguna idea de lo que podemos indicar algunas pruebas:

no he sido muy riguroso en los test, realizado me da una idea mas que supuesto con el mismo hardware).

no código para el MicroC de Dave 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.

Trasporta el open-source a la Empresa y a los desarrolladores

- 
- A pixelated illustration of Tux, the Linux mascot, a black and white penguin with a yellow beak and feet, standing on the left side of the slide.
- ➔ Disponibilidad de aplicaciones de una alta calidad técnica, no solo en plataformas.
 - ➔ Aplicaciones con una constante evolución y rendimiento, sobre todo en área de desarrollo.
 - ➔ Participar en proyectos de desarrollos open-source permite el constante mejoramiento de las aplicaciones 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.

Instalando el Compilador SDCC



Buscamos la URL <http://sdcc.sourceforge.net>
actual del compilador sdcc-2.5, su
a la forma tradicional a los

Realizamos:

```
sdcc-2.5.0.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
```


Ejemplos muy simples de Programación en C



1. Vemos un programa `led.c`, que es un pequeño programa que enciende un led con retardo de tiempo controlado por una interrupción.

2. Ejecutamos el programa y veremos algunos resultados en `makefile`, el código fuente, los archivos ejecutables y sus definiciones.

Herramientas open-source de ayuda al desarrollo y trabajo en equipo

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

Ver el trabajo de Franco Catrin en:

www.fccatrin.com/fcatrin/files/cvs.html

Como 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.



¿Cuál es el problema que nos encontraremos ahora?

- ➔ Hemos hecho el circuito esquemático en Eagle, hemos diseñado el PCB, lo vamos a fabricar... ¿cierto?... y nos han llegado una flamante placa impresa para probar los componentes... que después de probarla lo hemos hecho.
- ➔ Hemos probado los compiladores y lo hemos hecho funcionar... ¡cierto escándalo... :-), hemos realizado nuestra primera aplicación y tenemos el código listo.
- ➔ Ahora vamos a programar el Microcontrolador!

Programación del μ Controlador



...mente debemos tener un programador de
...adores y microprocesadores, que es un
... se conecta a un PC a la puerta
...itualmente y es manejado por una
...adivinen donde corre.... si..... en
...lo repitan por favor!)

...todo... excepto el como programar
... Pero estimado asistentes no
...desesperéis TUX tiene solución para casi todo... y
...lo que no... lo inventa.

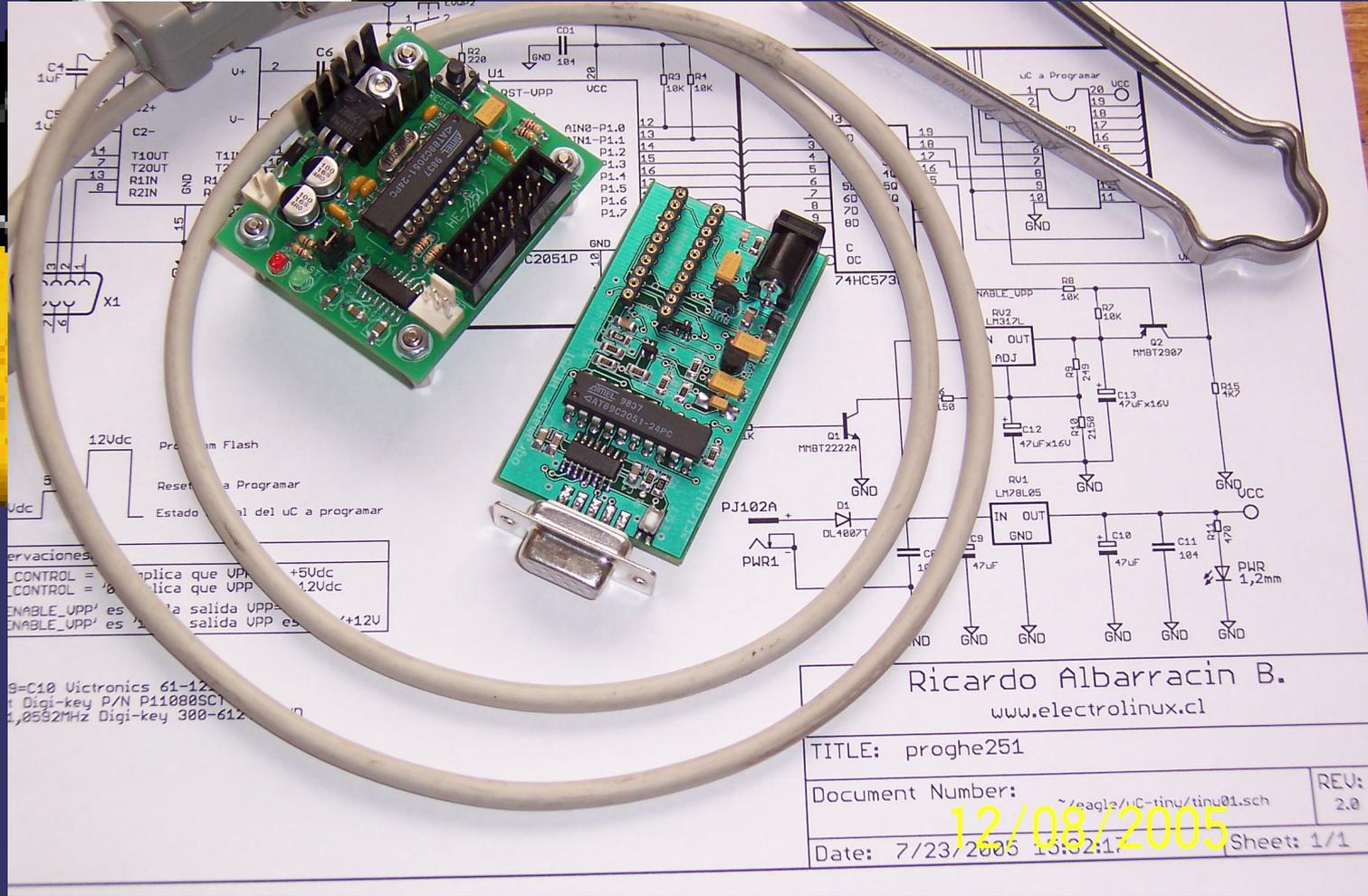
- ➔ Hay una aplicación open-source que permite programarlos, pero no todos los μ C... pero sirve, esta es la "uisp".

<http://www.amelek.gda.pl/avr/uisp>



Grabador para μP MCS51

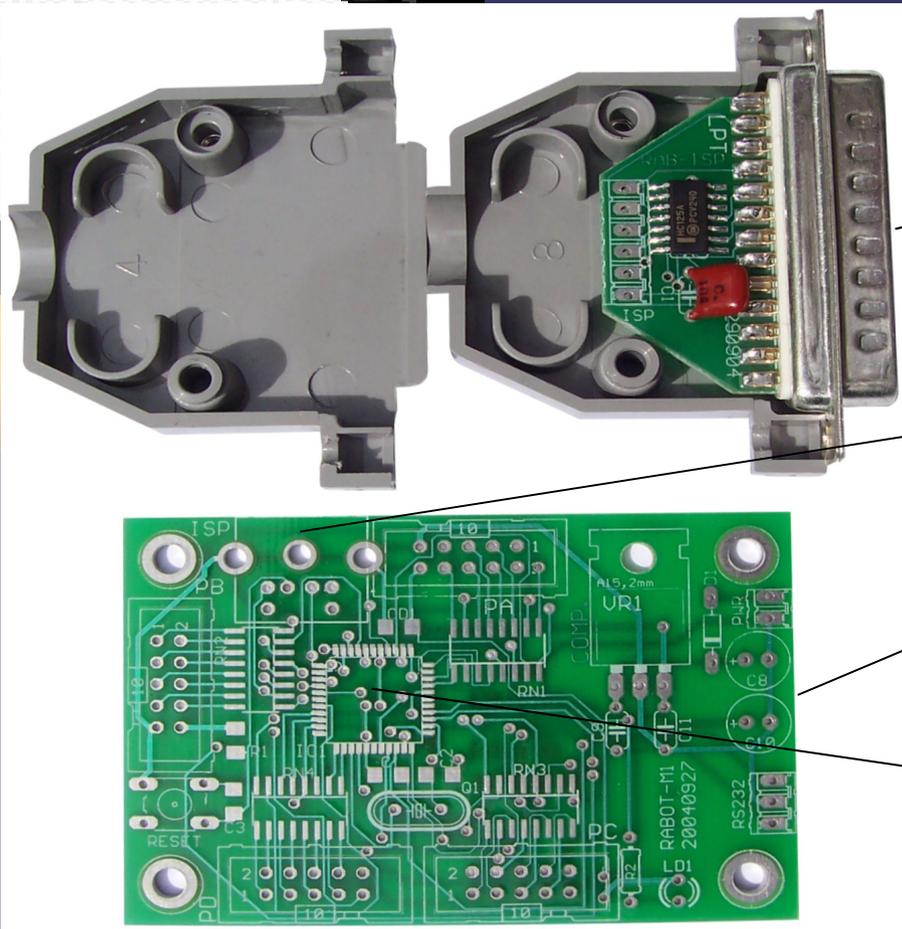
➔ **Ci Grabador MCS51, en desarrollo.**



Sistema de desarrollo completo para Microprocesadores y Microcontroladores



de desarrollo, desde el programador
microcontrolador, fabricado
si... en Chile



Programador ISP

Puerta InSystemProg

Sistema completo (PCB)

Microcontrolador RISC

Controlador completo

⇒ Controlador completo de Control



LCD 2x16

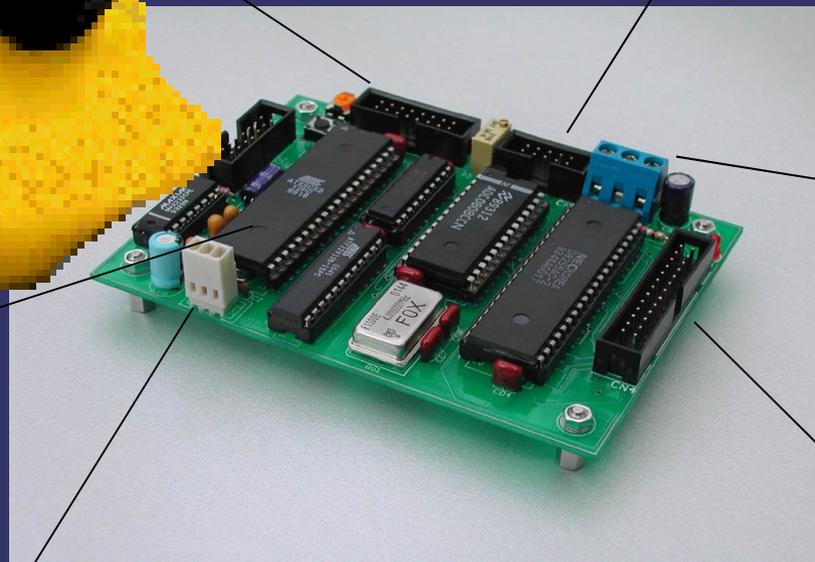
8 canales análogos

Alimentación
+12V/GND/+5V

μC MCS51

32 líneas I/O

Puerta RS232C



os desarrollos realizados
con estas Herramientas

ño, familia MCS51.

as

RS422

RS232

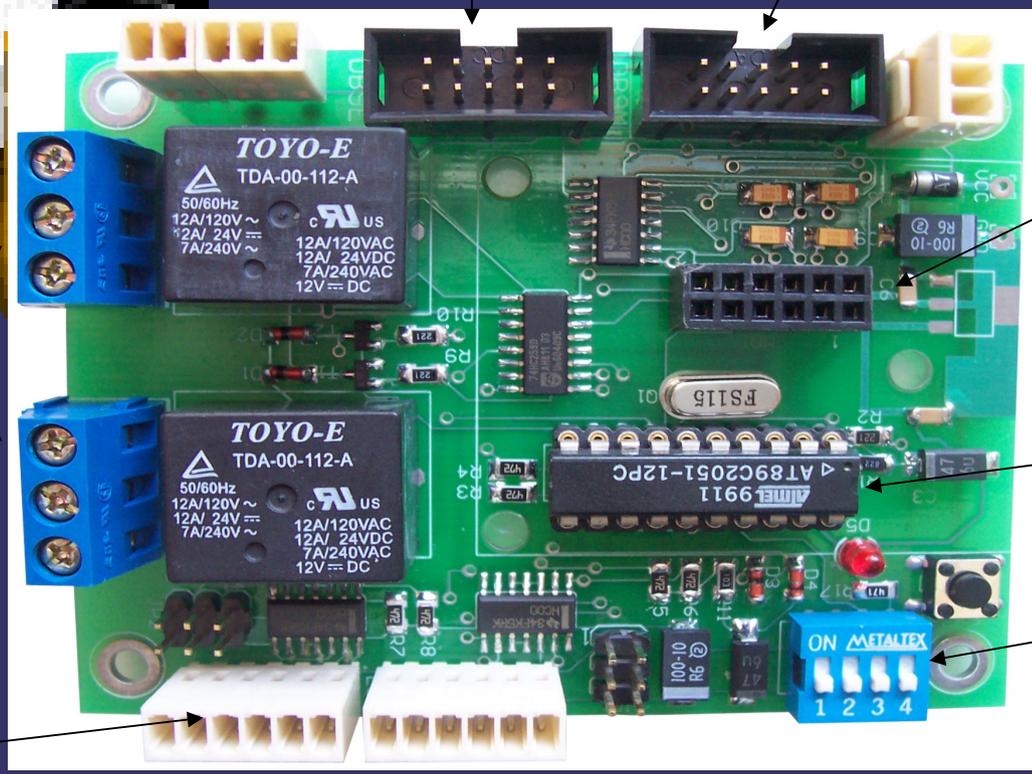
Puerta para
TCP/IP

μ C

Switch add

Entradas

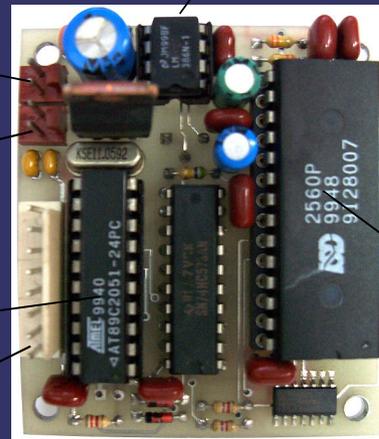
Actuadores
contacto seco



hardware que "habla"

prototipo por supuesto pero funciona.

Amplificador



Alimentación

CPU 8 bits

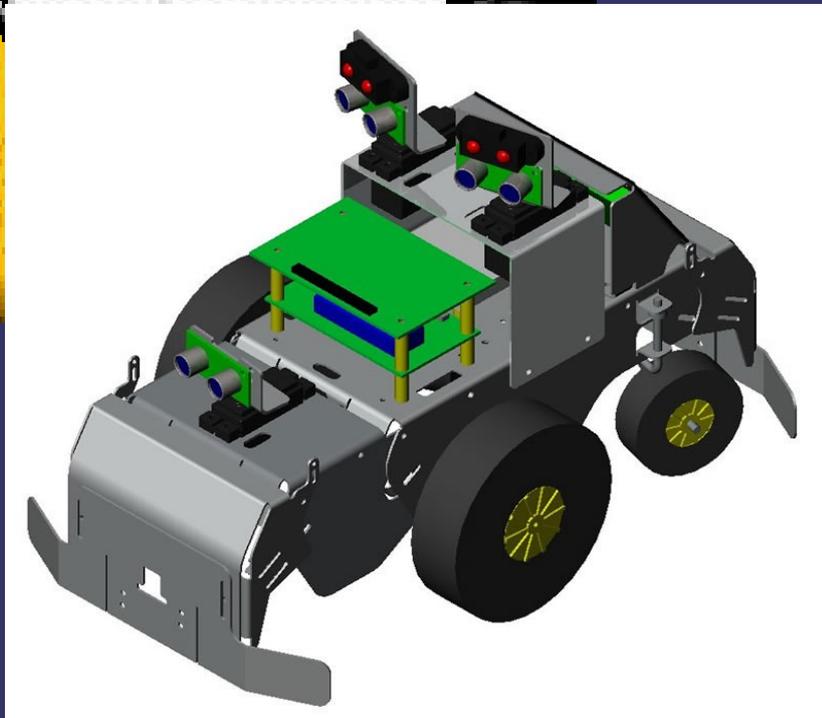
Entradas digitales

Grabador electrónico de audio

Tamaño de 5,1x5,6cm

Aplicaciones en Robótica y hobby's

esqueleto de un robot móvil, al que se pueden agregar diferentes sensores, cámaras, altavoces, etc...



Este es un prototipo de desarrollo y se está 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>



o es todo amigos

SUS...

icas, recomendaciones, alcances,
comentarios, reclamos,
nes, etc...

esta de sus dudas y preguntas,
atare de responder con el mayor
agrado.