

# ***Desarrollo de Electrónica Profesional bajo Linux***

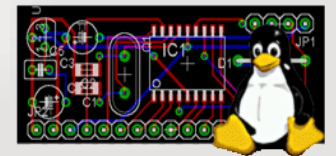
Primeras Jornadas de Linux en Rancagua

Organizada por la Escuela de Tecnologías Informáticas AIEP de la U. Andrés Bello

Autor: Ricardo Albarracín B.  
Diseñador de Sistemas Electrónicos Digitales con uC y uP  
Sistemas embebidos

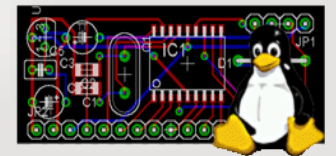
email: [rab@electrolinux.cl](mailto:rab@electrolinux.cl)  
<http://www.electrolinux.cl>

Rancagua 9 de Julio del 2005



# ***Temas que seran abordados***

- Situación presente.
- Consideraciones generales del diseño y desarrollo general.
- Herramientas de desarrollo open-source y freeware.
- Que PC debo usar para este trabajo?
- Microprocesadores (uP) y Microcontroladores (uC) posibles de utilizar en esta plataforma.
- Que herramientas de diseño electrónico existen bajo Linux.
- Que compiladores existen bajo Linux para estos uP y uC.
- Uso de Makefile y compiladores
- Ejemplos simples y hardware demostrativo.



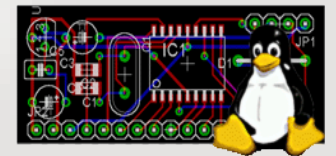
# ***Recomendaciones Importantes***

Que personas pueden desarrollar sistemas de este tipo?

Cualquier persona que tenga inquietud intelectual, sea dedicado, que le motive la tecnología, tenga facilidades de auto capacitarse, que le guste estudiar y aprender.

Importante:

**Desconecten los Teléfonos Celulares por favor.**



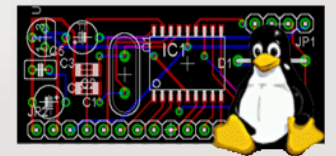
# ***Situacion Presente - I***

Por años la mayoría de las herramientas de desarrollo de electrónica están disponibles en “ese otro sistema”, son por lo general de un alto costo, un soporte regular (sobre todo para países como el nuestro, que se encuentra lejos de las grandes capitales del mundo), alta latencia de actualizaciones y para las nuevas versiones, se deben cancelar costos que bordean el costo de una nueva licencia.

Los costos siguen aumentando ya que el sistema “casi operativo” debe tener licenciamiento adecuado, lo que hace que el costo de cualquier desarrollo parta de un valor alto, sin siquiera hacer nada de desarrollo de electrónica aún.

Las herramientas típicas son; Tango, Orcad, Protel, Smartworks, Workbench entre varias otras, si bien esto pudiese parecer atractivo por su amplia “fauna”, en muchos casos se torna en contra debido a las incompatibilidades existentes entre las distintas herramientas, por lo que a los desarrolladores les obliga a aprender varias herramientas sobre todo a la hora de transferir información entre colegas o amigos.

Esto como resultado promueve el uso de estas herramientas sin un licenciamiento adecuado.



## ***Situacion Presente - II***

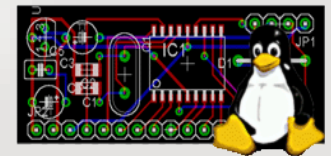
Por otra parte hay cada vez menos ingenieros en desarrollo de productos embebidos, son casi “bichos raros”, ya sea por motivación o simplemente por Mercado.

Actualmente existen otros fenómenos que atentan contra esto, es que los ingenieros que trabajan en las empresas consumidoras de tecnología nacional o desarrollos nacionales, no se atreven a “arriesgarse” pensando que actualmente “hay riesgos”, o simplemente desconocen de su existencia.

Ventajas de desarrollos locales: Tener al desarrollador localmente y adaptar el producto final a sus requerimientos específicos, esto trae beneficios a los participantes, pero además al país, ya que todas las criticas internacionales o los informes de expertos, critican la falta de desarrollo tecnológico local.

El peligro es que existen “desarrolladores locales” que están muy reñidos con el conocimiento y muy amigos de la “cuchufleta” y eso atenta en contra de los que trabajamos en desarrollo seriamente. También podemos decir que existen problemas de negociaciones, ya que el comprador en varias ocasiones pretende sacar el máximo de provecho al menor costo posible, ese “criterio” es malo para el proveedor y para el comprador.

Por último se debe decir que también los países industrializados pasaron por estas mismas etapas, también se equivocaron en sus diseños, pero afrontaron los problemas y salieron adelante con los productos que hoy en día son un standart, esto también lo podemos lograr pero debe haber un compromiso de todos, clientes y proveedores.



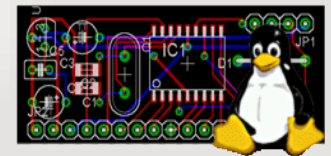
# ***Consideraciones Generales del Desarrollo Digital***

Que debemos hacer inicialmente:

- Identificar claramente los requerimientos del proyecto.
- Identificar las variables de éxito del proyecto.
- Separar lo que no cumple los requerimientos.

Que debemos tener claro:

- Un uC = (uP + RAM + FLASH + Periféricos), es decir todo en uno, determinar si esto es suficiente para los requerimientos del proyecto, es decir determinar la “capacidad de procesamiento”.
- Definir si se necesita un uC o un uP para el desarrollo del proyecto, si es una arquitectura RISC o CISC.
- Arquitecturas disponibles, Capacidad de Memoria requerida y periféricos necesarios.
- Herramientas de desarrollo disponibles.
- Aunque sea obvio, es muy importante el “Suministro de partes”.

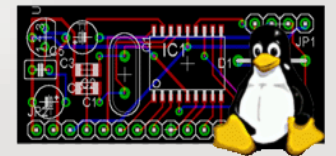


# Herramientas Open-Source y Freeware en Linux

Actualmente existe una serie de herramientas de desarrollo Open-Source, de muy buena calidad y en una constante evolución, ya que son proyectos que actualmente están en desarrollo, eso permite no solo que estas herramientas mejoren en el tiempo, sino que también permite que los usuarios corrientes participen en el proyecto... Esta es la magia del Open-Source.

## Herramientas disponibles:

- Para el diseño de esquemáticos y PCB, lo que uso es el Eagle (<http://www.cadsoftusa.com>). Existe una versión freeware que permite desarrollar placas impresas de hasta 10x8cm, lo que es bastante, incluso en proyectos profesionales.
- Para desarrollo de código, hay diferentes compiladores como el SDCC y el GCC, que son compiladores de una alta calidad técnica. En pruebas efectuadas con compiladores comerciales del “lado oscuro” las diferencias de tamaños y calidad de código son impresionantes, entre un 25% a un 60%.
- Hay variadas herramientas además como gEDA, Oregano, PCB, Kicad (3D) y otras disponibles bajo Linux.



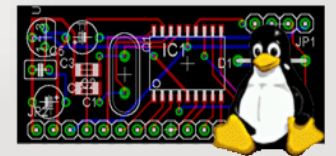
# Que PC debo tener para desarrollar?

Normalmente nos podemos imaginar que requerimos una maquina como las que son necesarias para correr "ese otro", que parece que nunca ese S.O. se encuentra satisfecho, a pesar de los tremendos recursos que usa.... "siempre requiere más" de lo que uno dispone.

En Linux afortunadamente no, debido a su alta calidad de desarrollo del código, un eficiente uso de los recursos disponibles y su mejor diseño, esto no es así

## Requerimientos:

- CPU de 500MHz o superior, deseable arquitectura i686 (P-III), los 586 funcionan, lento.
- Memoria RAM, lo que más pueda... esto es muy importante, mas que la velocidad de la CPU.
- Capacidad de disco 3GB hacia arriba, cualquier disco actual tiene 20GB o más.
- Tarjeta gráfica con 8MB de RAM, idealmente AGP y una Nvidia en lo posible.
- Tarjeta de RED 10/100BT, conectado a internet.





# ***Que uC y uP podemos usar para desarrollar en Linux***

Casi cualquiera.... si, hay arquitecturas de 8, 16 y 32 bits.

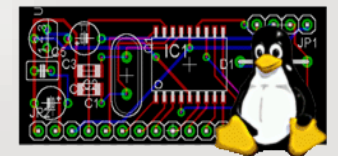
Motorola: Son procesadores CISC, Familia HC08, HC11 y HC12, muy usada en Europa.

Intel y clones: El archiconocido MCS51 y sus derivados, una amplia gamma, son procesadores CISC.

Atmel: Familias MCS51 y AVR (RISC), de muy buena calidad y de un excelente rendimiento. Familias ARM7 y otras más estos son procesadores de 16 y 32 bits, con compiladores bajo esta plataforma.

Microchip: Algunos de la serie de P16F y otros más (no los uso).

También se pueden usar el conocido Z80, Z180, DS80C390 y DS80C400 (variantes de la familia MCS51), Toshiba TLCS-900H, Phillip, Siemens, en fin muchos más de lo que se aprendería para desarrollar.



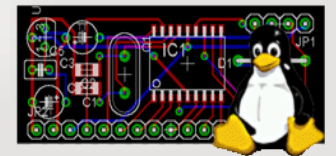
# ***Algunos productos desarrollados***

Nuestra empresa esta orientada al desarrollo de productos tecnológicos especiales, en que estos desarrollos son muy caros hacerlos fuera o son impensados por los volúmenes que demandan los fabricantes, en Chile tenemos una idiosincrasia distinta y comportamientos eléctricos muy peculiares.

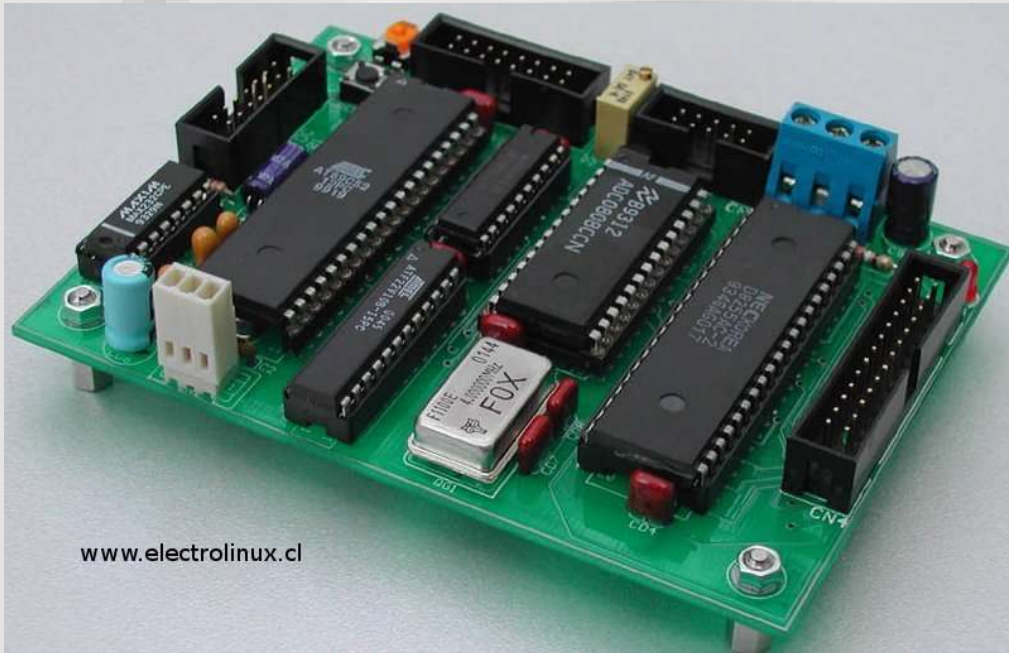
Hemos desarrollado para la educación, el desarrollo profesional y para los hobbistas de la robótica, una serie de productos orientados para tal efecto, están bajo el concepto de KIT's, los que puede ver en <http://www.electrolinux.cl>.

El kit tiene como objetivo entregar una rápida plataforma de desarrollo operativa, para que sobre ella se desarrolle el prototipo y el código, en este se prueba su funcionalidad y permite optimizar todo el desarrollo, esto evita el tener que partir sobre un desarrollo de hardware y software desde cero, permite ahorrar bastante tiempo.

Una vez que el prototipo esta terminado y funcionando, se le puede demostrar al cliente con una prueba de conceptos, sin haber invertido más que tiempo y algunos pesos. Posteriormente se le entrega el producto final, hecho a su medida.



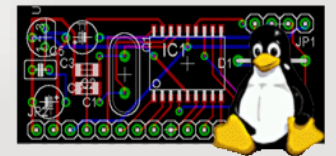
# Algunos Kits desarrollados Modelo TEC-01



## Sistema Completo de Control

- uP de 8bit con capacidad de hasta 22KB de Flash
- Puerta para un LCD de 2x16 o 4x16
- Puerta RS232C de hasta 57.600bps
- 8 Canales AD de 0-5Vdc.
- 40 Puertas de I/O programables
- Consume 145mA en 5Vdc.

Como complemento a este desarrollo hay conversores de 4-20mA a 0-5V, de tensión a corriente, los que son interfaces comunes en control industrial.

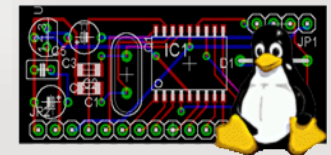


# Kit AVR arquitectura RISC



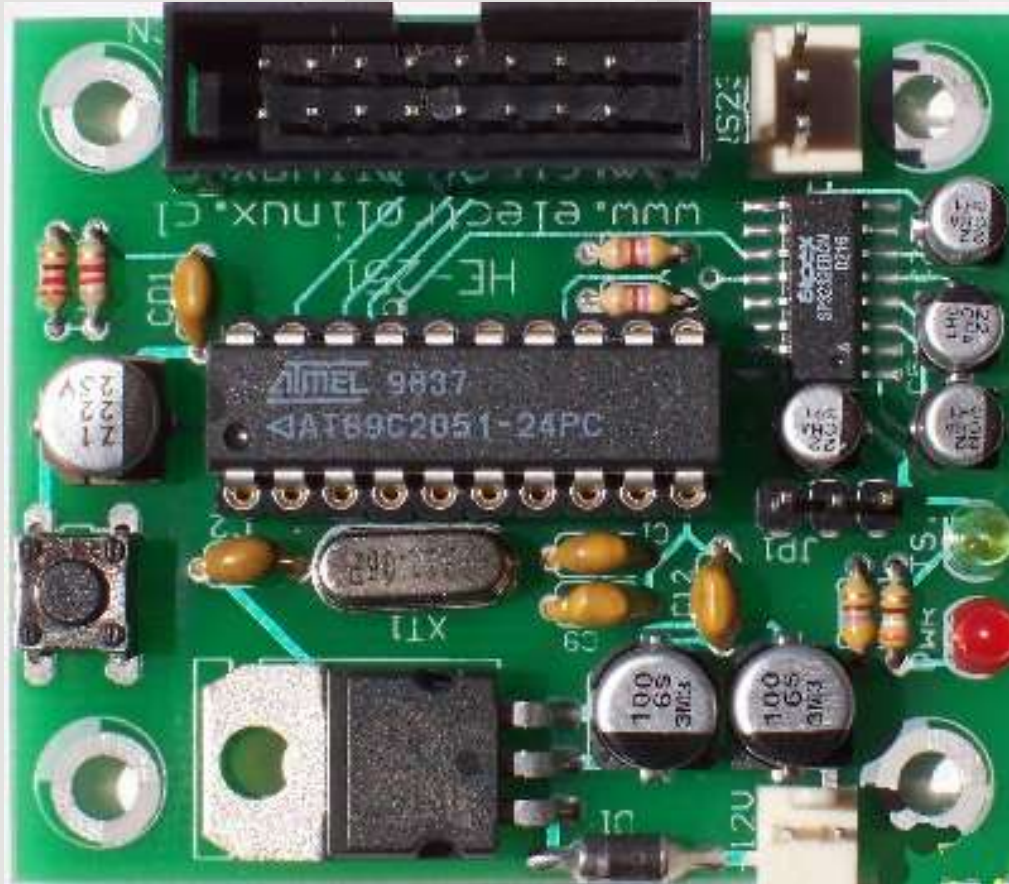
## Kit AVR (RISC)

- uC AVR Atmega8, excelente rendimiento.
- Hasta 8KB de Flash-ISP, EEPROM 512B, SRAM de 1KB.
- Puerta serial de RS232C de hasta 115.200bps.
- RTC incluido en el kits.
- 6 canales ADC de 10 bits (1024 pasos).
- Led de pruebas.
- 19 líneas de I/O programables
- Muy bajo consumo, típicamente 20mA activo.



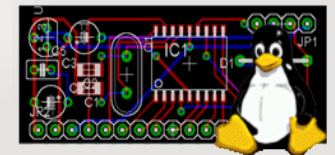


# Kits MCS51 modelo HE-251



Kit HE-251

- uC MCS51, de hasta 4KB de Flash.
- Conversor AD de 8 bits
- Hasta 12 líneas de I/O programables
- Puerta RS232C de hasta 57600bps
- Led de test
- Grabador de FLASH por puerta serial disponible próximamente
- Alimentación de entrada entre los 9Vdc-15Vdc.

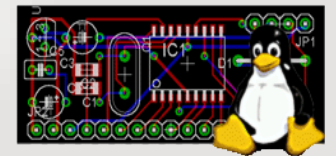


# ***Veremos 2 ejemplos prácticos de Código y Kits***

El primer ejemplo, es un uC que se encuentra conectado al PC y tiene una puerta RS232C, display LCD de 2x16 y una aplicación simple que permite desplegar los caracteres del terminal por la RS232 con un “echo” y con el LCD.

La segunda es el kit HE-251 conectado a un terminal como “echo”.

En nuestra pagina <http://www.electrolinux.cl>, tenemos pdf con la información y documentos que le serán de gran interés



# ***Eso fue todo amigos***

Espero ahora sus.....

- Quejas, reclamos, críticas, recomendaciones, comentarios, alcances... y manifestaciones.
- Estare gustoso de responderles sus consultas, aclarar sus dudas, responder sus preguntas.

Los espero en mi WEB ... <http://www.electrolinux.cl>

Tenemos foros, blog e información de interés para ustedes.

Gracias por su asistencia.

