

GNU/Linux, software libre para la comunidad universitaria

Sistemas embebidos

Copyright (C) 2010 José María Peribáñez Redondo chema@softlibre.net

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

COLABORADORES

	<i>TÍTULO :</i> GNU/Linux, software libre para la comunidad universitaria		
<i>ACCIÓN</i>	<i>NOMBRE</i>	<i>FECHA</i>	<i>FIRMA</i>
ESCRITO POR	José María Peribáñez Redondo	1 de noviembre de 2010	

HISTORIAL DE REVISIONES

NÚMERO	FECHA	MODIFICACIONES	NOMBRE
1.00	28-10-2010		José María Peribáñez Redondo

Índice general

1. Introducción	1
2. Por qué GNU/Linux triunfa en los sistemas embebidos	2
3. No todo es de color rosa	4
4. Atom sí, Atom no, Atom depende	5
5. Cambiar el software del dispositivo	7
5.1. El kit de la cuestión	7
5.2. Cómo de complicado es cambiar el firmware	8
6. Qué hardware podemos usar	10
6.1. NSLU2 de Linksys (XScale IXP420)	10
6.2. FreeAgent DockStar de Seagate (Marvell 6281)	10
6.3. El NAS de Dealextreme/WLX-652/ (STAR STR8131 ARMv4)	12
6.4. BifferBoard (RDC 321x)	12
6.5. Omnima Embedded Controller/Edimax router BR6104KP (Infineon ADM5120)	13
6.6. Otras opciones	13
7. Instalar OpenWrt en un ADM5120	15
8. Ecotorrent	18
9. Sugerencia de línea de proyectos innovadores	19
10. Para saber más	20
A. GNU Free Documentation License	21

Capítulo 1

Introducción

GNU/Linux tiene fama de ser un sistema operativo minoritario fuera del campo de los servidores y de nichos de mercado como la industria del cine: los más agoreros pronostican que nunca podrá ser un sistema operativo para el gran público como Windows. Se equivocan. En realidad, mucha gente está usando Linux sin saberlo y de hecho puede tener más de un GNU/Linux en su casa funcionando.

El motivo es que GNU/Linux es muy usado en sistemas empotrados (la traducción habitual, junto embebidos, del término *embedded* en inglés), es decir, en hardware que no son ordenadores personales sino dispositivos con un uso específico, por ejemplo un GPS o un reproductor de vídeo.

Así, si una persona tiene un navegador de TomTom, un televisor Sony Bravia, el lector de libros electrónicos de Amazon, cualquiera de los muchos teléfonos Android, un PalmPre o un Nokia N900 tiene un Linux en su vida. Varios modelos de routers ADSL (como el Comtrend CT536+) así mismo llevan GNU/Linux y lo mismo es aplicable a la mayoría de los reproductores multimedia de MKV, a no pocos modestos reproductores de Divx, los decos de Imagenio o a muchos NAS. Linux no sólo se usa en smartphones sino en teléfonos móviles convencionales: sólo con mobilinux se han vendido 35 millones de dispositivos móviles como teléfonos, equipos médicos, GPS y POS móviles (http://mvista.com/product_detail_mob.php)

El mercado de empotrados tiene un gran potencial. Desde que en 1990 se fundara ARM se han comercializado más de 15.000 millones de chips bajo licencia ARM y no hay que olvidar que hay otros vendedores como MIPS. Como referencia la población mundial se cifra en unos 6.800 millones de personas.

GNU/Linux se incluye tanto en dispositivos de uso muy específico que valen muchos miles de euros (como simuladores de combate) como en soluciones para construir Set-top-boxes. Llama la atención que ya hace 5 años (desde luego Linux no es un recién llegado en el sector) se podía encontrar un SoC (System on chip) que por 11\$ integraba el procesador, controladores USB, smartcard, salida televisión, decodificación MPEG2... Por supuesto no vamos a encontrar hardware por 11\$ dónde ejecutar GNU/Linux: al SoC hay que añadir más componentes electrónicos, conectores, memoria RAM y flash, alimentador, además de ser precios aplicados a grandes volúmenes de compras. Pero por 28 EUR sí que vamos a encontrar un dispositivo con 32MB de RAM, dos conectores USB y puerto de red.

Hay dos tipos de personas que tienen interés práctico en aprender sobre sistemas empotrados y GNU/Linux: por un lado están aquellos que se plantean crear un nuevo producto. Por otro están aquellas personas que les interesa la posibilidad de cacharrear con hardware popular y tratar de sacarlo el máximo partido. A estas segundas personas va orientado este texto, con el deseo de que la persona inquieta que hoy empieza cacharreando se anime quizás en el futuro a dar el salto a innovador y convertir un hobby en un trabajo.

En el terreno práctico, nos vamos a centrar en hardware concreto con la característica de ser barato y consumir muy poco, que sea una alternativa a tener el ordenador siempre encendido. La primera aplicación que viene a la cabeza es cliente de bittorrent, pero realmente hay muchas más posibilidades. Comentaremos también si esto sigue teniendo sentido hoy en día con los procesadores Atom, que son bastante baratos y de bajo consumo.

Capítulo 2

Por qué GNU/Linux triunfa en los sistemas embebidos

Antes de entrar en harina e ir a por los aspectos prácticos, un poco de culturilla general sobre el porqué se está apostando por usar Linux en tantos dispositivos.

De partida, cabe destacar que Linux triunfa pese a que hay otros sistemas que están diseñados específicamente para sistemas empotrados y que gracias a ello además de funcionar muy bien demandan menos recursos. Esto no es un detalle menor, porque si un sistema especializado necesita menos memoria que Linux y se trata de un dispositivo de bajo coste fabricado masivamente, el ahorro que supone incluir menos memoria no es en absoluto irrelevante.

¿Qué explica entonces el éxito de Linux? ¿serán los royalties? En sistemas en que se fabrican muchas unidades puede ser un factor importante, pero posiblemente en otros proyectos pese a su favor más lo que en principio parecía un inconveniente: no es un sistema específico para empotrados, sino un sistema diseñado para todo tipo de máquinas y muy especialmente los populares PC. Esto, a la postre tiene sus ventajas:

- Soporta todo tipo de hardware e incorpora los más modernos protocolos de red y tiene muchas bibliotecas de desarrollo. En contraste, puede haber sistemas más optimizados para dispositivos con pocos recursos, pero ser un problema para añadir soporte para nuevo hardware o añadir software. Pensemos por ejemplo en un NAS que admite discos internos y USB: muchos sistemas antiguos pueden limitar la controladora SATA y no digamos los dispositivos USB que reconocerán, así como la posibilidad de que funcione SAMBA y un cliente de bittorrent.
- Se programa igual que cualquier otro Linux; se puede reutilizar software y lo que es también muy importante, es más fácil encontrar programadores. Linux no es el sistema más popular, pero tiene un número de desarrolladores entusiastas que desearían los otros sistemas.
- Es un sistema muy robusto y eficiente, muy probado al ser el mismo código que el usado en escritorios y servidores: ningún otro kernel, incluyendo el de Windows, tiene tanta gente contratada trabajando en él.
- La desventaja de estar en origen menos optimizado para empotrados, se ha difuminado por dos motivos; por un lado mucha gente ha aportado modificaciones para adaptarlo y optimizarlo para sistemas empotrados, curiosamente a la par que otros han progresado en la misma medida en el sector más diferente a los embebidos, el de las supercomputadoras. También ha influido la propia evolución tecnológica: ahora hay chips lo suficientemente potentes para ejecutar Linux que son tan baratos, que no merece la pena tratar de buscar otros más simples y si los hay, pues para tener opciones como USB o red no tienen sentido los diseños más simples antiguos.

Hay varios ejemplos de todo esto, como el avance de Linux frente al retroceso de Windows Mobile (también conocido como Windows CE) que aunque sea de Microsoft no tiene nada que ver con Windows y es un sistema diseñado específicamente para empotrados. De hecho el otro sistema que ha triunfado en los móviles, el iPhone, usa un sistema operativo adaptado de MacOSX, no un sistema diseñado desde el principio para móviles. También se puede comparar con la evolución de eCoS, un sistema operativo diseñado para empotrados desde el principio, que es software libre y tiene la misma licencia que Linux por lo que tampoco hay pago de royalties. ECoS fue adquirido en su momento por la empresa líder en GNU/Linux, Red Hat, pero lo ha

ido abandonando en favor de Linux y ha terminado cediendo el copyright a la Free Software Foundation. Con todo, eCOS sigue desarrollándose y usándose con numerosos casos de éxito: <http://www.ecoscentric.com/ecos/examples.shtml>) y debe destacarse que al utilizar el API POSIX es utilizable por programadores de Linux. Como curiosidad, muchos sistemas empuotrados con independencia del sistema que lleven, incluyen como cargador RebBoot, que está basado en eCoS.

Por supuesto sigue habiendo sitio para los sistemas optimizados para empuotrados especialmente cuando se trata de ahorrar en hardware: es el caso de los routers de Lynksys que migraron de Linux a Vxworks excepto en versiones especiales avanzadas, porque aunque Vxworks sea de pago, ahorraron al sacar los nuevos modelos con la mitad de memoria.

Capítulo 3

No todo es de color rosa

A la hora de lanzarse ilusionado a personalizar un dispositivo basado en Linux, la primera decepción es que aunque siempre habrá más oportunidades que con un sistema cerrado, los dispositivos con Linux pueden ser también terriblemente cerrados y ser complicado o imposible modificarlos. ¿Cómo es esto posible? ¿no es GNU/Linux software libre?

Lamentablemente el tema de los firmwares depende mucho del vendedor. El kernel y el sistema base puede ser libre, pero no el software que realiza la tarea propia del dispositivo. Por ejemplo en un reproductor de DVD, podemos no sólo no tener el código del reproductor, sino del sistema gráfico que nos permite mostrar algo en pantalla, ni tener ninguna forma de recoger los eventos del mando a distancia. Es más, hay sistemas en los que el vendedor no permite actualizar el firmware, porque sólo se puede por web y una utilidad comprueba que el firmware esté firmado. Lamentablemente la GPLv2 no puede hacer contra esto, mientras que la GPLv3 sí, pero no es la licencia utilizada en el kernel ni en muchas otras partes que se usan en los sistemas embebidos.

La práctica de no permitir firmwares mejorados tristemente tiene su razón de ser para muchos fabricantes, que temen que las nuevas características que hacen más atractivo un modelo nuevo o de gama más alta puedan simplemente incorporarse a los modelos más viejos o baratos mediante una actualización de firmware. Por ejemplo los televisores con decodificador TDT-HD fácilmente podrían reproducir películas MKV con audio MP3 o AC3, porque los codecs ya los necesitan para la TDT y además la mayoría incluyen un puerto USB de servicio. Sin embargo esta posibilidad sólo la ofrecen algunos modelos (y como muestra de que simplemente tienen que aprovechar los codecs necesarios para la TDT es que no suelen reproducir el audio DTS precisamente porque el codec tendrían que licenciarlo y sin embargo no lo necesitan para la TDT-HD). En algunos televisores LG esta característica podía activarse entrando en el modo de servicio y LG desactivó esta posibilidad al actualizar el firmware.

La GPLv3 es una buena forma de promover los derechos de los usuarios frente a los fabricantes: que puedan usar como quieran el hardware que han pagado, utilizando el software libre creado por otras personas para que se pueda usar sin restricciones por cualquiera. Esto es una garantía para el usuario tanto en dispositivos embebidos como en los smartphones en los que el fabricante es el único que puede aprobar qué aplicaciones se instalan en el teléfono. Es también una oportunidad de negocio: la GPL no hace fácil pretender que la versión de Linux sea libre y en cambio la de Windows de pago y que no se pueda copiar. Pero he aquí que fabricantes como Apple o Microsoft propician que sí sea posible cobrar por cada copia que se instale en un iPhone sin dejar de ser libre y cumpliendo escrupulosamente la GPLv3. El truco está en que mientras que la GPLv3 da los mismos derechos de uso y copia a los usuarios de Windows que a los de Linux, no permite distribuir el software a través de una plataforma que restringe que los usuarios no puedan instalarse lo que quieran. De este modo el autor del código al tener el copyright puede publicar bajo una licencia dual, una gratis para los usuarios del mundo libre y otra de pago para los usuarios que libremente eligen ser cautivos y pagar por lo que otros tienen gratis.

Otra fuente de frustraciones es la tendencia de los fabricantes de cambiar el hardware sin cambiar el nombre del modelo. Ni que decir tiene la mayoría de las veces este cambio es para ahorrar con un alternativa hardware más barata o incluso con prácticas como reducir la memoria RAM o flash.

Capítulo 4

Atom sí, Atom no, Atom depende

Tal y como dijimos, nos centraremos en hardware barato y que consuma poca energía que sea la alternativa a dejar el PC encendido. Pero la pregunta es ¿no es Atom también barato y consume poca energía, por lo que me sirve para dejar un cliente torrent las 24 horas del día, pero además me ofrece mucha más cosas como PC que es?

La respuesta es que Atom es efectivamente más barato y consume menos energía que un procesador de portátil. Pero aún así la diferencia con muchos procesadores de ARM es abismal, sobre todo si las aplicaciones a ejecutar no requieren apenas potencia y pueden usarse micros muy simples y a pocos megahercios. En el caso de la mayoría de los diseños basados en ARM y MIPS, la energía necesaria para todo el sistema en un Atom apenas serviría para alimentar tan solo el procesador, si llega.

Un menor consumo no es sólo ahorro de electricidad. Está la menor emisión de calor y al no necesitar ventiladores la ausencia total de ruido y tamaños mínimos. Como muestra, algunos de los sistemas que veremos llevan alimentadores de 6W de potencia máxima, mientras los más potentes alcanzan teóricamente algo más de 20W pero para poder alimentar uno o más discos duros por el puerto USB. Por contra la mayoría de las fuentes para placas mini-ITX como las que llevan los Atom no bajan de 150W y sólo en cajas específicas como las de Zotac o ASRock encontraremos los típicos alimentadores como los de netbook, pero aún así de 65W. En precio, una solución Atom empieza por los 66 EUR de la placa (incluye procesador) a los que hay que sumar caja con fuente de alimentación y la memoria. Por supuesto no tiene ni comparación en cualquiera de sus características con una solución barata (empezando por la memoria, que como mínimo posiblemente compremos 1GB, frente a los rídiculos 16MB de algunas soluciones empotradas) pero la cuestión a pensar es siempre si ese hardware es suficiente para el propósito que quiere darse....

En realidad hay casos en los que un sistema Atom nos parece una alternativa interesante a sistemas embebidos con procesadores basados en ARM o MIPS. Es el caso de los reproductores multimedia de gama alta, con precios por encima de los 200 EUR. Ahí la diferencia de precio no es argumento de peso y el consumo tampoco es fundamental pues no se trata de tenerlo encendido todo el rato y en cualquier caso la diferencia de consumo entre Atom y el reproductor es anecdótica al lado del consumo del televisor de trepocientos pulgadas. El único punto delicado es el ruido, para lo que hay que escoger con cuidado el equipo. Las ventajas que ofrece el sistema Atom son muchas:

- Por el precio ya se tiene un ordenador completo de salón, apto por ejemplo para navegar o conectarse a las redes sociales. También se podrá tener funcionalidades avanzadas como recibir notificaciones mientras se ve la película
- La máquina Atom puede utilizarse también como escritorio remoto para tener acceso desde el salón a otro ordenador, puede funcionar de NAS avanzado...
- Como reproductor multimedia tiene más posibilidades, no habrá fichero que se resista y no hay problema de codecs, se usa el mismo software que para PC, no se depende del firmware del fabricante, se tienen media centers muy trabajados como XBMC. Eso sí, se necesita un sistema Atom con chip ION o ION2, lamentablemente propietario. Es una pena que Intel no incluya en los Atom los chips gráficos que llevan sus micros normales (i3, i5 e i7) porque potencialmente estos sí que permiten reproducir ficheros de alta definición y no son privativos. También hay una solución hardware con controladores libres para hacer sólo la decodificación (http://www.broadcom.com/support/crystal_hd/), pero el problema de los Atom sin ION es que ninguna placa viene con salida HDMI. Hay que decir que hay vientos de cambio y se vislumbra una mayor presencia de micros ULV en lugar de Atom, que aunque cuesten y consuman más, siguen siendo una ganancia respecto a otros procesadores.

- Meter nuevo software es trivial, porque se ejecuta la misma distribución que en un PC y aunque el hardware Atom esté limitado en relación a un PC contemporáneo y se pueda considerar equivalente a un desfase de varios años, la mayoría de aplicaciones funcionarían también en hardware aún más antiguo.

Hay que lamentar que si un PC de salón con Atom puede ser mejor alternativa que uno basado en chips especializados como los de Sigma o Realtek, es precisamente por el carácter cerrado de estos dispositivos a la hora de tratar de hacer algo con el firmware. No hay ninguna razón técnica por la que no se saquen modelos más flexibles que sean reproductores y permitan navegar libremente (y no sólo acceder a unos pocos servicios previstos por el fabricante como YouTube o Flickr), más allá del carácter cerrado de los desarrollos, quizás motivado por el pánico a los clones chinos de cualquier producto tratándose de diferenciar en el firmware.

Otra alternativa es como NAS, sobre todo considerando que muchos NAS las versiones más económicas llevan chips ARM y los caros Atom. Aunque un NAS con Atom va a consumir sensiblemente más, una ventaja es que posiblemente la placa soporte wake-on-lan y se pueda apagar cuando no se usa.

Puede considerarse que la ventaja del reproductor multimedia o del NAS avanzado con ARM frente a Atom es que también puede usarse para estar todo el rato encendido con tareas como descargar torrents. Ahí cabe considerar otra opción: no empeñarse en tener un sólo aparato para todo, sino conforme a la filosofía Unix en las utilidades, apostar por la especialización: un procesador Atom o uno más avanzado para multimedia o NAS de alto rendimiento y un dispositivo que sea más barato para tener siempre encendido, encargarse de cosas como las descargas e incluso de encender el ordenador.

Capítulo 5

Cambiar el software del dispositivo

5.1. El kit de la cuestión

En los sistemas empotrados el firmware (software) normalmente está grabado en una memoria flash. A la hora de tratar de aprovechar las características de un dispositivo más allá de lo que permitía su firmware original, caben dos posibilidades: seguir usando el sistema original, pero ampliado con nuevas aplicaciones o reemplazar totalmente el firmware que venía en la memoria flash por otro. Lo primero, que tiene como ejemplo el popular optware, puede ser menos peligroso y permite conservar la funcionalidad del firmware original. Es muy útil cuando el firmware original es bueno y cuenta con una interfaz web cómoda u ofrece funcionalidades de las que no se dispone el código fuente por lo que se perderían al reemplazar el firmware, por ejemplo la parte de ADSL en muchos routers. Sin embargo esta opción también suele ser más limitada al obligar a usar el mismo software que usaron los creadores originales del firmware: esto se traduce en la práctica en que hay menos programas disponibles que si se reemplaza el firmware por otro software más flexible. Es más, en algunos sistemas esto obliga a usar un kernel 2.4 y versiones muy viejas de las librerías.

A la hora de sobrescribir el firmware con otro sistema Linux, en ocasiones se puede elegir entre varias opciones: desde instalar una versión de Debian para ese procesador, que normalmente es la solución que más programas (paquetes) ofrece pero también la menos optimizada, a usar un sistema como OpenWRT <http://www.openwrt.org> con versiones más reducidas del software y de la librería libc, pero que tiene la ventaja de necesitar menos memoria RAM y menos espacio en la memoria Flash. Una alternativa a OpenWRT también orientada a consumir el mínimo de recursos es buildroot, algo más orientado a desarrolladores que quieren construir un sistema totalmente personalizado más que una distribución: openWRT nació como un firmware para un router y cuenta con interfaz web, un sistema de configuración y paquetes instalables, mientras que en Buildroot no hay nada de eso y aunque utilice el término paquetes son más bien módulos pues no es algo que instale el usuario sino partes que se añaden o no al sistema de ficheros.

Los sistemas más interesantes son los que permiten conectar un disco USB, ya sea para poder montar un sistema de ficheros /opt donde estarán las nuevas aplicaciones o para poder montar el sistema de ficheros directamente de un disco USB en lugar de usar la memoria flash. La memoria flash suele estar muy limitada en capacidad y además permite muchas menos escrituras antes de empezar a fallar, lo que hace que sea inviable usarlas para hacer swapping y tener memoria virtual (muy útil porque estos dispositivos suelen venir también muy cortos de RAM) e incluso pueden llegar a ser un problema con algunos programas como clientes de bittorrent que escriben mucho en el disco para mantener estadísticas actualizadas. Además cualquier operación inadecuada en el disco USB se sabe que se no va a impedir que el dispositivo arranque o al menos va a poder arreglarse desde otro ordenador, mientras que una mala modificación en el firmware instalado en la memoria flash es más delicado. Además en una memoria flash no se puede escribir directamente ficheros como en un disco, salvo que haya una capa de software (en los pendrives de hardware) que se encarga de traducir de un dispositivo de bloques a las peculiaridades de las memorias flash (por la limitación en el número de escrituras, en una memoria flash no se puede indicar que se quiere escribir en el bloque x, sino que hay una lista de bloques libres; para modificar un fragmento de un fichero, el bloque que lo contenía se marca como borrado y se encola para su uso futuro, tomándose el primer bloque libre de la lista para escribir el nuevo contenido; de este modo el número de escrituras se reparte homogéneamente entre todos los bloques de la memoria flash y no hay bloques que se queman en seguida por alojar contenidos que cambian con frecuencia). Hay sistemas que en la memoria flash aloja simplemente un sistema de ficheros de sólo lectura.

La inmensa mayoría de los dispositivos, aunque permitan usar un disco USB lamentablemente lo que no permiten es arrancar de él al estilo de los PCs. Esto es una pena, porque eso permitiría arrancar un firmware más potente sin necesidad de tocar el

existente, con lo que todo sería más fácil, sin posibilidad de romper nada y elegir entre usar el firmware original y el modificado sería tan simple como conectar o no el disco USB. En realidad cualquier dispositivo podría arrancar (de forma limitada) de USB sin necesidad de hardware especial, simplemente utilizando como cargador uboot con una extensión especial para reconocer discos USB y dentro de ellos las particiones Linux y poder ejecutar así un kernel presente en el disco. No es una solución igual de buena que un arranque USB de verdad (por ejemplo no funcionaría si se ha corrompido el contenido de la memoria flash y afecta al cargador) pero es mucho mejor que no tener nada.

Si no se puede arrancar directamente de USB, hay un par de alternativas que en la práctica funcionan muy bien, pero que implican modificar el firmware. La idea es o bien cambiar los parámetros del kernel para que trate de montar como sistema raíz uno situado en un dispositivo USB en lugar de en la memoria flash, que es la opción más rápida, o bien que el software que se ejecuta al arrancar el Linux presente en la memoria flash lo primero que haga sea montar el disco USB externo y pasar a utilizarlo como sistema de ficheros raíz (hacer pivotroot), que aunque sea menos directo permitiría decidir si montar el disco USB o ejecutar el software de la memoria flash u otras opciones como montar un sistema de ficheros en red. Por ejemplo puede decidirse entre un arranque u otro en función de que el usuario pulse un botón, el resultado de una consulta de red o simplemente que esté o no conectado el disco USB.

Ambas alternativas implican modificar el firmware para poder montar el sistema por USB, pero a partir de ese momento ya no hará falta volver a tocar el contenido de la memoria flash. La única excepción es para modificar el kernel, dado que se lee de la memoria flash y es él quien monta el disco USB. En algunos sistemas es posible utilizar una característica interesante del kernel de Linux llamada kexec. Consiste en que una vez que ha arrancado el sistema, es posible cargar un kernel en memoria y ejecutarlo reemplazando al actual (equivalente a volver a arrancar, pero ahora con el nuevo kernel y de forma muy rápida). Esta opción es muy valiosa, porque permitiría modificar tan sólo una vez el disco flash para grabar un kernel con kexec y un pequeño sistema que se encargue de buscar el kernel que quiere ejecutar el usuario de un disco USB o una localización de red y a partir de ahí ya no haría falta tocar más la memoria flash, ni para actualizar el software del sistema de ficheros ni para cambiar el kernel. Esta opción es atractiva sobre todo porque modificar la memoria flash es delicado (es cuando hay riesgo de que luego el sistema no arranque) y en algunos dispositivos implica hacer algo especial, como abrirlos y usar un cable serie. Un vendedor que quiera proporcionar un sistema tan fácilmente modificable como uno que permita arrancar de USB, simplemente tendría que comercializarlo con el firmware así modificado.

Hablando de dispositivos USB, una tentación es usar un pendrive en lugar de un disco USB: consumen menos, son más silenciosos y si nos vale uno de poca capacidad son baratos. Para muchos casos está bien, pero para escrituras intensivas o si hay necesidad de usarlos para swap, ya se han comentado que limitaciones tienen en número de reescrituras. Curiosamente hay una alternativa a los pendrives, basada en usar unos productos que se han quedado anticuados con el abaratamiento de las memorias flash pero que se pueden obtener baratos en EBay. Se trata de los microdrives, que con el tamaño y conectores de una tarjeta compact-flash (como las que suelen llevar las reflex) en lugar de tener en su interior memoria flash tienen un diminuto disco duro. Suena increíble, pero en las wikipedia se pueden ver fotos de cómo son por dentro y efectivamente son discos duros. Los microdrives se usaban en su día como alternativa a las caras memorias flash y en su momento lo han usado dispositivos como los primeros iPod. La ventaja para nosotros es que los microdrives al ser discos duros no tienen problema con el número de reescrituras y se pueden usar también para swap. Consumen más que un pendrive normal (por este mayor consumo muchos lectores que admiten tarjetas compact-flash no son capaces de leer microdrives, por lo que lo mejor es comprar directamente un microdrive con USB, que parecen pendrives un poco bastos) pero obviamente menos que un disco duro convencional.

5.2. Cómo de complicado es cambiar el firmware

La mayoría de los dispositivos dividen el espacio de la memoria flash en tres o cuatro áreas: el cargador que arranca el kernel, el kernel, el sistema de ficheros con el software y en algunos casos un área de datos para guardar las preferencias del usuario, como configuración de red y contraseña, para que no se pierdan al actualizar el firmware. Muchos firmwares ofrecen una interfaz web que permite reemplazar el propio firmware por otro: en la mayoría de los casos sólo tocan el sistema de ficheros y el kernel; ojo con algunos sistemas que no nos sirven para reemplazar el firmware, porque sólo sirven para modificar una parte del software, no todo el sistema de ficheros (por ejemplo en una TDTHD con posibilidad de ver MKVs, es típico sólo poder modificar el propio programa de la TDT y películas, no el resto del sistema). Así pues, dependiendo del sistema, sin necesidad de abrir el dispositivo podremos cambiar tanto kernel como sistema de ficheros, sólo sistema de ficheros o directamente no podremos. En algunos casos habrá una segunda opción, que es conectarnos por SSH y modificar directamente el sistema de ficheros.

Cuando no es posible modificar el firmware vía web o simplemente si por cualquier razón se han hecho modificaciones inadecuadas en el sistema de ficheros o en el kernel y el sistema no arranca y por lo tanto no es posible recurrir a la opción web, en muchos sistemas es posible la opción de usar un cable serie. Aquí lo que se aprovecha es que la mayoría de los dispositivos

incluyen un cargador avanzado como redboot o uboot, que no sólo son para arrancar el kernel sino que además permiten escribir en la memoria flash y así modificar el firmware. El cable serie es para poder interactuar con el cargador e indicarle que queremos escribir en la memoria flash, en algunos además también habrá que transferir por el cable serie el contenido a grabar, mientras que en otros se podrá indicar que se descargue de un servidor TFTP, con lo que la transferencia al ser vía red será mucho más rápida. Hay que reseñar que en los sistemas en los que hace falta un cable serie para indicar al cargador que queremos sobrescribir el firmware, es porque el fabricante no ha querido proporcionar un sistema alternativo. En sistemas como el NSLU2 de Linksys o los tablets y smartphones de Nokia N800, N810 y N900, existe la posibilidad de utilizar determinados botones (puede ser tan simple como pulsar el botón de reset mientras se enciende) para indicar al cargador que se debe poner en modo de reflaseo. En estos dispositivos no hace falta acceso al puerto serie y la imagen a grabar se pasa vía USB. Lamentablemente la mayoría de los fabricantes son más hostiles a que los usuarios puedan modificar el firmware y además suelen negar la garantía a quienes lo hagan: posiblemente la garantía legal siga amparando al usuario mientras quede claro que no es culpa de la modificación del firmware, pero como se sabe la garantía legal sólo es eficaz para el usuario los primeros 6 meses, el año y medio restante la presunción de inocencia cae del lado del fabricante y es el consumidor quien tiene que demostrar.

En muchos sistemas existen los tres pines necesarios para tener un conector serie: TX, RX y GND, es decir, el pin para transmitir, para recibir y el común para tierra, pero no están accesibles desde el exterior de la carcasa. En el caso de los fabricantes hostiles a que los usuarios modifiquen el hardware que venden, abrir la caja podría invalidar la garantía comercial; con la legal es menos claro, pero aparte de lo recordado de los 6 meses está el problema de que ante un fallo de un componente eléctrico interno es menos claro demostrar que no se ha tocado nada. Aunque esos tres pines componen un puerto serie, hay que tener en cuenta que no usa los voltajes que establece la norma RS232, sino que opera con voltajes muy inferiores (3.3v): se suele hablar de puerto serie TTL, en mención a que se usa voltaje de comunicaciones entre transistores (estrictamente TTL sólo sería adecuado para voltajes de 5V, siendo el término correcto para 3.3V LVTTTL, pero se ha popularizado utilizar el mismo término para los dos). Afortunadamente hay muchos cables en el mercado con un chip adaptador (por ejemplo el cp210x), para conectar a un puerto USB y es más, es posible hacerse con uno muy barato por coincidir con el cable de datos de modelos antiguos de teléfonos de marcas como Nokia o Siemens: así pues basta con buscar en un sitio como Dealextreme por Nokia CA-42. Además, en el caso de tiendas online que venden directamente equipos dedicados a personas que quieren meter un Linux en ellos, lo normal es que vendan también el cable ya adaptado para que se pueda conectar al dispositivo directamente. Con otros dispositivos es posible que nos toque cortar el cable y hacer alguna pequeña soldadura. Todos los dispositivos que vamos a analizar tienen en común que el puerto serie es de este tipo, TTL de 3.3v y por lo tanto serviría para todos el mismo cable conversor de TTL serie a USB o RS232, con la peculiaridad que la conexión en el dispositivo variará.

Para aficionados, las opciones para reflasear un dispositivo se agotan aquí: si se sobrescribe el cargador de arranque probablemente hayamos convertido nuestro dispositivo en un pisapapeles (el temido término utilizado en los foros en inglés es bricked). Hay algunas excepciones, en kits de desarrollo que vienen con dos memorias flash y un conmutador para arrancar de una u otra: con este sistema siempre es posible arrancar de la que funciona para arreglar la otra.

¿Hay alguna solución ante un dispositivo bricked? Muchos dispositivos tienen también pines de la interfaz JTAG, que se usa para depurar y para acceder a los chips del dispositivo directamente. Mediante un adaptador JTAG en la mayoría de los casos es posible escribir en la memoria flash y esto se hace directamente, por lo que funcionaría aunque estuviera totalmente corrupta la memoria flash y no se ejecutara el cargador. En algunos casos es posible incluso reflasear el firmware con el dispositivo apagado (la picaresca para si el sistema está averiado y se quiere retornar con el firmware original). Lo malo es que la opción JTAG no está al alcance de la mayoría de los usuarios, tanto por complicada como porque necesita hardware especial (el adaptador JTAG). Desafortunadamente ni siquiera está estandarizado el conector JTAG, aunque al menos la mayoría de dispositivos ARM tienen el mismo y la mayoría de los MIPs también tienen uno común pero distinto al que suelen llevar los ARM. Así pues puede ser necesario distintos adaptadores JTAG y distinto software. La mayoría de los adaptadores JTAG pueden costar más que el dispositivo a reflasear, por encima de los 100EUR, aunque los haya sensiblemente más baratos (sobre todo por puerto paralelo) que pueden servir según el sistema y están disponibles por unos 15 EUR.

Con todo, hay dispositivos como el TP-LINK 1043WR-N que tienen interfaz JTAG pero no se conoce cómo usarla para escribir en la memoria flash. En esos dispositivos o en los que no tienen JTAG, la única opción (muy complicada para la inmensa mayoría de los mortales) sería desoldar los finísimos pines de la memoria flash, modificarla con un programador externo y volverla a soldar.

Capítulo 6

Qué hardware podemos usar

Una realidad frustrante con el hardware para sistemas empotrados, es que algunos componentes son teóricamente muy baratos y sin embargo no es nada fácil hacerse con una unidad a buen precio. Los kits de desarrollo con excepciones suelen ser caros sobre todo cuando hay que pedirlos a EE.UU. y hay aduana por medio (aunque son bastante útiles cuando se está desarrollando profesionalmente) y normalmente la única alternativa barata es tener la suerte de encontrarse en el mercado con algún dispositivo basado en Linux en el que sea fácil de modificar el firmware.

A continuación vamos a comentar hardware que podemos encontrar a precios no muy caros, en la mayoría de los casos son productos que la gente ha descubierto que llevan Linux y son fáciles de modificar, salvo en dos productos, que se vende específicamente para que los compradores puedan meter Linux y el software que quieran.

6.1. NSLU2 de Linksys (XScale IXP420)

Este dispositivo originalmente lo comercializaba Linksys como un NAS económico para compartir discos USB por red. Desafortunadamente está descatalogado, aunque es probable que lo encontremos todavía en alguna tienda o en eBay. Es más caro que el resto de los productos analizados, aunque también el más potente excluyendo el FreeAgent DockStar. Así mismo es el más versátil, con mayor comunidad de desarrolladores y más información sobre él en Internet, incluso con páginas en castellano.

Tiene 32 Mbs de RAM, dos puertos USB2 y tarjeta de red. El procesador es un XScale IXP420 a 266Mhz: como curiosidad la familia de procesadores XScale aunque sea ARM es de Intel y ha seguido evolucionando, por ejemplo desde 2007 existe el Intel CE 2110 para dispositivos empotrados.

De lo mejor del dispositivo es que permite elegir entre quedarse con su firmware original e instalar paquetes aparte (optware), instalar la versión para ARM de Debian, instalar un firmware específico optimizado para el dispositivo, muy potente que permite incluso compilar los propios paquetes en el dispositivo, sin necesidad de usar un crosscompiler, o instalar OpenWrt, entre otras opciones.

Es muy fácil de reflasear sin necesidad de abrirlos ni utilizar cables especiales: mediante los botones se puede poner en modo de reflaseo y simplemente se ejecuta un programa en el PC que transfiere la imagen vía USB. Como esta operación la hace el cargador, está disponible aunque la imagen del Linux instalado no funcione.

No hace falta detenerse mucho en el NLUS2, porque en la web <http://www.nslu2-linux.org/> ya hay todo tipo de información.

6.2. FreeAgent DockStar de Seagate (Marvell 6281)

Como el NSLU2, se comercializa como un NAS para discos USB, pero es hardware más potente con la interfaz de red Gigabit, procesador a 1.2 GHz, 128 MB de RAM y nada menos que 4 puertos USB2. Uno de ellos es un conector macho microsub para insertarlo directamente en los discos FreeAgent de Seagate, pero debería funcionar con cualquier otra carcasa 2.5 con este tipo de conector en el centro.

El DockStar lo está liquidando Seagate en su web de EE.UU. y en Ebay está realmente barato, por debajo de los 28 EUR, pero lamentablemente entre gastos de envío y posibilidad de aduanas, ya no sale tan económico. Si se compra directamente en España

sale con gastos de envío por unos 60 EUR en Pixmania y se tiene la ventaja de la garantía y no hacer falta adaptador para la clavija al venir ya la europea.

El DockStar lleva el mismo sistema (SoC Marvell Kirkwood 88F6281) que el famoso plug computer Sheeva Plug. Los plug computer son pequeños dispositivos que se conectan directamente a un enchufe de pared con lo que casi no ocupan espacio. El Sheeva Plug es considerado como el sucesor natural del NSLU2 y ha sido llamado el NSLU2 killer, si bien no se comercializa en España y sale algo caro con los gastos de envío. El DockStar aunque lleve el mismo chip que el Sheeva Plug por lo demás incluye hardware más modesto: 128 MB de RAM en lugar de 512, la revisión más reciente también tiene menos flash, 256MB en lugar de 512 MB (aunque que nos toque la de 256 o 512 es cuestión de suerte) y no incluye lector de tarjetas SDHC. El único punto en contra del Sheeva Plug es que sólo tiene un puerto USB.

Una diferencia en contra del DockStar que le hace menos adecuado para el cacharreo es que el Sheeva Plug permite arrancar no sólo de la memoria flash, sino también de una tarjeta SD y de cualquier dispositivo USB. En realidad no es que arranque de USB como un PC porque lo permite una BIOS, sino que siempre arranca de la memoria Flash pero el cargador UBoot tiene la extensión ext2load (es decir, que si se corrompiera el cargador de la memoria flash, no sirve arrancar de USB para arreglarlo). Así mismo el Sheeva Plug proporciona interfaz serie y JTAG sin necesidad de abrirlo, a través del puerto USB. Sheeva Plug incluye software y documentación para usarlo con Linux y viene precargado con una Ubuntu (lamentablemente esta distribución se ha quedado obsoleta, al no haber sacado Ubuntu nuevas versiones para ARM compatibles). El DockStar carece de todo esto, tiene la posibilidad de usar un puerto serie pero abriéndolo y más difícil que en otros dispositivos. En definitiva, es más difícil de modificar y entraña más riesgo.

Otra diferencia entre el DockStar y el Sheeva Plug, es que el DockStar no es un plug-computer: no tiene el alimentador dentro de la caja, sino externo, como el resto de dispositivos comentados. Esto a la larga es una ventaja, pues el procesador tiene fama de emitir calor y es más conveniente que el alimentador esté fuera para que ventile mejor, además de ser así mucho más fácil y barato reemplazarlo si se estropea, algo que le ha pasado a más de un poseedor de un plug-computer.

Aparte del DockStar y Sheeva Plug hay otros modelos, algunos con características avanzadas como conexión eSata para aprovechar realmente las posibilidades como NAS (la propia Seagate posiblemente esté liquidando el DockStar en favor del otro modelo, el Go-Flex Net, que permite conectar dos discos SATA). No recomendamos el GuruPlug, por tener menos facilidades de arranque y puerto serie a través del puerto USB y porque en algunas versiones hay problemas graves de calentamiento. El DockStar realmente deriva del PogoPlug, que se comercializa como un servicio cloud para poder acceder a los discos desde Internet sin dejar el ordenador encendido, si bien mientras que en el PogoPlug este servicio es gratis de por vida, en el DockStar es sólo gratis el primer año.

Para poder instalar un nuevo firmware en un disco duro externo, lo que se hace en el DockStar es aprovechar que inicialmente se tiene acceso como root por ssh. Entonces se reemplaza el cargador por otro que permite arrancar de USB como en el Sheeva (rezando para que no se vaya la luz mientras escribe y acabemos con un pisapapeles). Este cargador además redirige lo que sería el flujo del puerto serie a un puerto de red, por lo que ya no haría falta abrir el dispositivo y poner un cable en el puerto serie TTL. Esta solución basada en simplemente arrancar de disco duro permite no tocar el firmware original y poder usar el servicio de acceso al disco duro por Internet si se hace un arranque normal. Lo que no existe en la actualidad es la opción optware, instalar nuevos paquetes pero usando el firmware original: antes existía, con el proyecto OpenPlug pero este dejó de mantenerse y se reemplazó por PlugBox, una versión de ArchLinux para estos dispositivos. La web <http://www.plugapps.com/> es el sitio de referencia para quien desea modificar su DockStar o hardware similar.

Aunque la frecuencia de reloj puede impresionar, hay que tener en cuenta que es un procesador tecnológicamente poco avanzado, con arquitectura bastante inferior a los smartphones del año 2007 o al que lleva hoy en día algunos lectores de libros electrónicos. Para hacerse una idea de las distintas generaciones de los procesadores ARM, con ejemplos de dispositivos de consumo como teléfonos que los usan, visitar http://en.wikipedia.org/wiki/ARM_architecture.

Este chip de Marvell u otros de la misma familia se usan también en varios NAS, con la frecuencia del procesador bajada a 800Mhz:

- Synologic DS210J (175EUR, 25W con discos duros/10W sin discos), DS110J (119EUR, 19W/9W). El software es bastante completo. 128 MB RAM.
- Qnap TS-110 (220,15 EUR con disco duro de 1GB). 256 RAM, 16 MB flash. También buen software y comunidad.
- Netgear ReadyNas Duo. 256 RAM.

Estos NAS tienen sus propios proyectos para modificar el firmware sin reemplazarlo, ya que su punto fuerte es precisamente todo lo que permite hacer su firmware original con una buena interfaz: Time Machine, DLNA, descargas...

Otro NAS más antiguo con chip Marvell es el D-LINK DNS-323, con dos bahías SATA por unos 162 EUR. <http://wiki.dns323.info/howto>
Hablando de D-LINK, una página curiosa es <http://tsd.dlink.com.tw/downloadslist.asp?SourceType=downloadOS=GPL> dónde está disponible el código fuente de todo el software GPL que usan sus productos. Este tipo de páginas las hay también en otros fabricantes y son interesantes tanto para ver en cuales de sus productos usan Linux como qué otro software interesante han usado y podríamos usar nosotros también en nuestros proyectos. Por ejemplo es recomendable la de TomTom.

6.3. El NAS de Dealextreme/WLX-652/ (STAR STR8131 ARMv4)

Es la opción más económica: se comercializa como un NAS para discos USB pero además con la posibilidad de funcionar como cliente bittorrent. El firmware original es bastante flojo, pero es posible instalarle un nuevo firmware sin necesidad de abrirlo, desde la interfaz web. Sólo si sale algo mal (y no ha sido al sobreescribir el cargador, cosa que sí se hace la primera vez) existe la opción de reflasear desde el cargador de arranque, utilizando un cable serie para lo que habrá que abrir. Como otros modelos, no es un puerto RS-232 y hará falta un adaptador, además de un poco de maña para añadirle unos pines al cable, porque al abrir el dispositivo nos encontramos simplemente tres pequeños agujeros en lugar de un conector, pero unos pines de tamaño estándar ajustan perfectamente.

Tiene dos puertos USB2, interfaz de red, 32 MB de RAM y 4 MB de flash, el procesador es a 250Mhz un ARMv4, una de las arquitecturas más básicas de ARM. El alimentador es de 5V (como los puertos USB) y hay que decir que el cargador original es bastante malo, pero es fácil hacerse con otro en Internet, teniendo en cuenta eso sí que no sólo cuenta el voltaje sino el diámetro del conector que es fino, coincide con el conector de la mayoría de los hubs USB que admiten alimentación externa.

Quizás lo más negativo es que no es posible meterle OpenWrt ni ninguna otro software popular y por ahora la única opción es el firmware SnakeOS que está disponible en Internet, con lo que no hay mucha flexibilidad, o cuanto menos implica complicarse más la vida para compilar uno mismo los programas utilizando un SDK. Así mismo esta distribución funciona grabándose en la memoria flash, no puede estar enteramente en un disco USB como en los otros sistemas, si bien tampoco es mucho problema porque sí permite montar un disco en el directorio /opt e instalar ahí los binarios.

Los dos modelos con este chips en DealExtreme son:

- <http://www.dealextreme.com/details.dx/sku.26320>
- <http://www.dealextreme.com/details.dx/sku.20383>. Un usuario asegura que los nuevos envíos de este modelo vienen con otro chip, pero no ha sido confirmado.

La página con información para cambiar el firmware por SnakeOS es <http://groups.google.es/group/dealextreme-nas/>

6.4. BifferBoard (RDC 321x)

Es la solución de menor tamaño de las analizadas, tiene un conector USB 2.0, un conector de red, 2 leds (realmente controlable sólo uno, el otro es de encendido) y un botón de reset. La memoria RAM es de 32 MB. El procesador es un RDC R3210 a 150Mhz, curiosamente este procesador es compatible con el 486.

La memoria flash de 8 MB viene precargada con OpenWrt (obviamente con pocos paquetes), lo interesante de poder usar la BifferBoard sin disco USB es que entonces se puede usar incluso un hub USB para alimentar el dispositivo. En lo negativo, el puerto USB que incluye no es capaz de alimentar un disco duro de 2.5. El alimentador es de 5V y 6W con el mismo tipo de barrilete que el de DealExtreme.

Se puede conectar un cable serie (como en otros modelos, no es RS232 sino TTL de 3.3V) para depurar o cambiar el firmware. El software necesario para usar la interfaz JTAG es de pago.

Es posible que el mismo SoC se use por ejemplo en el Conceptronic Printer and Storage Network Adapter disponible en algunas tiendas por unos 30 EUR, dado que físicamente es igual. Lamentablemente, además de que habría que confirmarlo, el diseño de cajas de este dispositivo hace difícil abrirlo sin romper nada una vez que está cerrado. Otro problema es que mientras que la BifferBoard tiene un cargador propietario que permite reflasear el dispositivo, muchos de los dispositivos con el mismo hardware vienen con un cargador sin esa opción.

La web dónde se puede comprar y obtener más información es: <http://bifferos.bizhat.com/>

6.5. Omnima Embedded Controller/Edimax router BR6104KP (Infineon ADM5120)

Se trata de un router con 4 bocas de red y dos puertos USB, originalmente concebido para compartir una impresora de red, pero que puede usarse también para conectar un disco USB. Este router lo ha comercializado la marca Edimax, pero hoy en día la mejor forma de hacerse con uno, con todas las facilidades para instalar Linux es acudir a una web inglesa de electrónica: <http://www.omnima.co.uk/> (eso sí, los gastos de envío no son nada baratos).

El procesador es un MIPS32 a 175Mhz. Ver <http://www.linux-mips.org/wiki/Adm5120>.

Omnima lo vende con la opción de que venga ya modificado el firmware para montar un sistema de ficheros vía USB y el pendrive con el software ya instalado. También comercializa el cable serie necesario, el mismo que hace falta para los otros modelos pero ya con el conector adecuado para este dispositivo.

Los sistemas basados en el SoC ADM5120 tienen dos importantes limitaciones respecto a las otras alternativas: tiene tan sólo 16 MB de RAM y el USB es 1.0. Además el controlador USB es bastante atípico y problemático, da problemas por ejemplo si se conecta a la vez un dispositivo USB 1.1 y uno USB 2.0, no permite dispositivos de audio.. La limitación de velocidad de USB no será problemática con bittorrent, pues la línea será más lenta (sobre todo en subida y se supone que se debería subir tanto como descargar) es significativa cuando se quiere transferir un fichero a otro ordenador mientras prosigue una descarga o para reproducir vía red un fichero de alta definición o un DVD (en cambio no da problemas para reproducir un fichero DivX tradicional a la vez que se descargan otros).

Entre lo positivo, sus posibilidades como router, con sus 5 bocas cada una de las cuales funciona como una interfaz de red independiente con su propia MAC. Es posible hacer también agrupar como switch las bocas entre sí independientemente y el switch está implementado por hardware. Puede ser una opción interesante para personas con conexión a Internet monopuesto (que sólo pueden conectar un equipo) y no quieren añadir más cacharros, al unificar el dispositivo las tareas de router, descargas y posibilidad de encender los otros equipos mediante wake-on-lan.

Otra de las características por la que se compra este dispositivo pese a sus limitaciones es por los 13 leds, perfectamente controlables por software, al igual que el botón de reset. Los aficionados a la electrónica desoldan los leds y los usan como GPIO (generic I/O), dado que se puede escribir y leer.

El alimentador es de 12V (como el conector de mechero de los coches) y el transformador que vende Omnima es de 6W, suficiente para alimentar el dispositivo (que consume unos 4W) pero no tanto si se conecta un disco duro USB de 2.5: para ese caso se recomienda hacerse en EBay con un alimentador de 12W. Hay tiendas en Alemania que venden alimentadores solventes: cuidado con los de DealXtreeme, que ni siquiera tienen el sello CE. Una alternativa es utilizar un cable USB en Y y utilizar el extremo que es sólo para recibir corriente en un hub USB .

Para la parte práctica nos vamos a centrar en este dispositivo, aunque la mayoría de lo comentado será aplicable a cualquier otro dispositivo que soporte OpenWRT.

6.6. Otras opciones

A continuación se enumeran otras opciones que a diferencia de las anteriores no ha podido probar el autor, por lo que sólo dispone de la información presente en Internet:

- Las placas de Soekris son un clásico, aunque caras para sus posibilidades actuales: <http://www.soekris.com/>
- BeagleBoard <http://beagleboard.org/>. Muy completo, con procesador de la familia que llevan los smartphones de gama alta, salida DVI... La cuestión es que uso se le plantea dar al no ser barato (su punto fuerte es potencia en relación al consumo, muy útil con batería, pero no tan claro enchufado a red eléctrica). No tiene mucho sentido para uso doméstico, por ejemplo como ordenador de salón, por no ser válido para reproducir vídeo.
- PandaBoard <http://pandaboard.org/> Puede considerarse la sucesora de la BeagleBoard, con CortexA9 en lugar de A8 y características mucho más prometedoras. La parte negativa es que al precio hay que sumar gastos de envío y posiblemente IVA y aduana.

- EM2440-III: MINI2440 (Samsung ARM9 S3C2440A ARM920T a 400MHz) con 64MB RAM, 64 memoria Flash tipo NAND, 2MB tipo NOR, lector de tarjeta SD, dos puertos USB uno master y otro slave, red, 4 botones, 2 leds, sonido... Se vende con pantalla táctil, de 3,5 vale 106 EUR y de 7 cuesta 148 EUR. No se especifica resolución pantalla. Tiene más cosas que otros modelos analizados, pero la cuestión es que se espera de este dispositivo al tener pantalla y si merece dedicarle mucho tiempo, pues las tecnologías emergentes para embebidos como Android o Meego no van a poder correr en este dispositivo. No es fácil desarrollar aplicaciones gráficas para este modelo porque no tiene mucha RAM y el procesador tiene poco rendimiento. Como referencia el fallido teléfono FreeRunner de OpenMoko tiene el mismo procesador pero con el doble de RAM. La tendencia en sistemas con interfaz de usuario es hacia ARMv7 (Cortex A8 y A9), que es lo único por ejemplo que soporta la versión actual de Ubuntu para ARM.
- Routers Wifi que tengan puerto USB. Hay opciones como la Fonera 2.0n, fácil de modificar y con 64 MB. Si se valora sobre todo capacidad router y Wifi, TP-Link TP-LINK 1043WR-N es barato, interfaz Gigabit, tres antenas, aunque 32 MB de RAM. Con 1043WR problemático también que TP-LINK es hostil a cambios de firmware y modelo es especialmente fácil de convertir en pisapapeles. Tampoco muy probado el uso del puerto USB.

Capítulo 7

Instalar OpenWrt en un ADM5120

Aunque hay un firmware basado en OpenWrt antiguo listo para instalar en el ADM5120 y también podríamos haber incluido uno reciente hecho por nosotros en el DVD, vamos a hacer todo el proceso y crear nuestro propio firmware con propósitos didácticos. Podemos descargar una versión estable del código fuente o directamente del SVN el código más reciente. Optamos por lo primero:

```
wget http://downloads.openwrt.org/backfire/10.03/backfire_10.03_source.tar.bz2
```

```
tar xjvf backfire_10.03_source.tar.bz2
```

```
cd backfire_10.03
```

Instalamos una serie de paquetes para poder compilar y para finalmente grabar el nuevo firmware en la memoria flash:

```
sudo apt-get install ncurses-dev gawk flex build-essential quilt
```

```
wget https://squidge.svn.sourceforge.net/svnroot/squidge/adm_upload
```

Es el momento de configurar lo que queremos que se compile. En este primer momento no vamos a detenernos a pensar qué aplicaciones queremos en nuestro flamante sistema, sino tan sólo lo necesario para que el sistema arranque, el resto lo compilaremos luego como módulos, que se traduce en paquetes instalables. Es importante compilar todo lo que puede instalarse una vez que el sistema ha arrancado como módulo y no incluirlo directamente, porque todo lo que seleccionemos para incluir en el sistema y no vaya como módulo, no sólo irá luego en el disco USB sino también en la imagen de la memoria flash, que es importante no ocupe mucho porque hay poco espacio. En cambio opciones como el sistema de fichero ext3 o el soporte de USB no pueden ir como módulo, pues hasta que no arranque el kernel no podrá cargar los módulos y están en el disco USB, formateado con ext3.

Empezamos ejecutando *make menuconfig*: A continuación indicamos lo que hay que seleccionar o cambiar, todo lo demás se deja como está:

- en Target System marcamos Infineon/ADMtek ADM5120 y en Target Profile seleccionar Edimax BR-6140KP
- en Target Images dejamos sólo squashfs y tgz
- En base system cambiamos a módulos los componentes dnsmasq, dropbear, firewall, mtd. Esto quiere decir que en lugar de copiarse a la imagen, se crean paquetes que podrán instalarse luego. Esto es importante con la imagen que se graba en la memoria flash, porque tiene poca capacidad. Hacemos lo mismo con iptables y ppp en network y admswconfig en utilities.
- Activamos los siguientes componentes que estaban sin seleccionar, pero siempre como módulos: en utilities, kexec-tools y ldd
- hacemos exit hasta salir de la consola.

Todavía no ejecutamos make. Ahora hay que configurar el kernel, ejecutando *make kernel_menuconfig*

- empezamos por *Networking support/networking options/Network packet filtering framework*: aquí todas las selecciones serán como módulo. En Core Netfilter Configuration seleccionar netfilter connection tracking support. En IP: Netfilter Configuration seleccionar IPv4 connection tracking support, IP tables support, Full NAT, MASQUERADE target support, REDIRECT target support
- entramos en device drivers.
 - entramos en Memory Technology Device (MTD) support y desactivamos las tres opciones que comienzan por Automatically....
 - pasamos a SCSI device support, cambiamos el propio soporte de SCSI de módulo a que forme parte del kernel y activamos también para compilar en el kernel SCSI disk support (los discos USB se reconocen como discos SCSI).
 - seguimos por USB support, cambiamos support for Host-side USB para que se compile en el kernel en lugar de como módulo y lo mismo con adm5120 hcd support y USB mass storage support.
- ya podemos salir de devices drivers; es el turno de *file systems*. Tiene que ir integrado en el kernel *ext3* mientras que irán como módulos fuse, vfat y cifs (este último dentro de network filesystems).
- finalmente en kernel hacking, vamos hasta el final y sustituimos la línea que indica cómo montar el sistema de ficheros por esta otra:

```
console=ttyS0,115200 root=/dev/sda1 init=/etc/preinit ro rootwait
```

Tras esto, salimos, ignoramos los errores y ejecutamos make (con un procesador de doble núcleo, mejor con la opción -j3)

Al acabar todos estos pasos, tendremos todo en bin. El fichero con extensión .bin es el que tendremos que grabar en la memoria flash, el .tgz el contenido que habrá que volcar en la primera partición del disco USB, formateada con ext3. Finalmente en packages van los paquetes instalables una vez que el sistema arranque.

Para grabar en la memoria flash, tenemos que descargar el programa adm_upload, darle permisos de ejecución y hacer un pequeño cambio para que funcione en Ubuntu.

```
conectamos el cable serie de Omnima, sin encender todavía el ADM5120. Observar la pequeña ↵  
muesca circular en la cabecera del cable: deberá ir sobre el pin marcado con el número 1 ↵  
en la placa.
```

A continuación ejecutamos:

```
wget https://squidge.svn.sourceforge.net/svnroot/squidge/adm_upload
```

```
sed -e s/lsz -X/sx -X/ adm_upload adm_upload.new
```

```
mv adm_upload.new adm_upload
```

```
chmod 755 adm_upload
```

```
./adm_upload -x -b -d /dev/ttyUSB0 bin/adm5120/openwrt-adm5120-br-6104kp-squashfs-xmodem. ↵  
bin
```

Es el momento de encender el ADM5120; el proceso de reflaseo se iniciará automáticamente. Tarda un rato porque necesita transmitirse la imagen entera vía puerto serie, pero desde el principio se verá como va el progreso.

Si todo ha salido bien, ya no volveremos a necesitar el cable serie ni tocar la memoria flash salvo que queramos cambiar el kernel. Tan sólo habrá que encender con el disco duro conectado.

Todas estas instrucciones son usando la última versión de OpenWrt (backfire), hay un tutorial bastante interesante y detallado para la versión anterior (kamikaze) que en caso de duda nos puede servir de referencia:

```
http://137.132.247.137/se-esd/workshop/
```

Al ejecutar `make menuconfig`, nos habremos dado cuenta que apenas había programas para elegir. El motivo es que hay muchos paquetes en OpenWrt que no son parte de la distribución. En cualquier caso es muy sencillo añadirlos: hay que ir al subdirectorio `scripts` y ejecutar `./feeds`. Un detalle importante es que estos paquetes se toman del SVN, por lo que en algún caso nos podemos encontrar con que incluso falla la compilación por añadir uno de estos paquetes. En ese caso simplemente se deselecciona y vuelve a ejecutar `make` para que ahora compile sin el paquete.

Estas son las opciones más interesantes:

- `update`: actualiza la lista de paquetes disponibles. Es el primer comando que hay que ejecutar
- `list`: muestra todos los paquetes disponibles
- `search`: permite buscar un paquete por cualquier palabra del nombre o de la descripción
- `install`: al especificar un nombre de paquete, hace que ese paquete aparezca en la lista de disponibles al ejecutar `make menuconfig` (instala también sus dependencias). Con la opción `-a` hará que aparezcan todos los paquetes disponibles

También está la opción de instalar los paquetes ya precompilados de la web de openwrt, en lugar de generarlos nosotros: http://downloads.openwrt.org/backfire/10.03/adm5120_mipsel/packages/. Existe una utilidad de uso parecido a `apt-get` llamada `opkg` para instalar los paquetes.

Capítulo 8

Ecotorrent

Ecotorrent es un proyecto que tiene por finalidad tener un dispositivo para descargar torrents que sea económico y ecológico (esto último por el bajo consumo) a la par que abierto. Consta de varios programas en C y unos pocos scripts, uno de ellos orientado a crear una imagen de partición a grabar en un disco duro o microdrive que pueda ser usado en un ADM5120 y que ya esté configurado. En el sistema de ficheros se incluyen así mismos scripts que se ejecuten durante el primer arranque para que hagan funciones como instalar paquetes con los componentes que falten. El script es interesante también para aprender como configurar distintos aspectos de OpenWrt.

Con sus 16MB de RAM, un ADM5120 con ecotorrent ofrece:

- demonio de bittorrent que admite compartir y descargar varios torrents simultáneamente
- manejo del cliente de bittorrent de forma remota mediante comandos, jabber o web. La web es la antítesis de la 2.0 al funcionar con comandos, pero gracias a su simpleza funciona desde cualquier dispositivo, incluido cualquier smartphone. Operaciones como añadir, parar, reanudar, borrar, descarga de un fichero vía wget, posibilidad de pasar el .torrent como una URL o subiendo el fichero (esto ha sido importante para poder añadir descargas desde un Nokia 5800, al no admitir ninguno de los navegadores copiar enlaces). El uso de bittorrent para controlar la máquina permite hacerlo de forma remota sin tener que abrir puertos
- disponibilidad de ssh para tener un proxy SOCKS y por ejemplo acceder a webs desde un ordenador del trabajo enviándose las peticiones realmente desde el adm5120 de casa. Esto es útil para privacidad o para webs de descargas directas.
- posibilidad mediante ssh, clave pública y extensión de Firefox flashgot, que al ir a descargar un fichero desde el navegador, en lugar de hacerlo el ordenador se encargue el adm5120 de modo que se pueda apagar el ordenador sin cortar la descarga. Lamentablemente esto no siempre funciona (implica repetir la petición, lo que no funciona en varios sitios de descargas) pero en construcción solución más genérica para proxy que se encargue ante descarga de hacerla él y devolver a navegador dirección donde estará el fichero. Esto útil para comenzar descarga desde fuera de casa pero usando de forma transparente el router, o hacerlo desde casa pero desde un teléfono móvil con conexión wifi.
- navegador web de ficheros con operaciones básicas
- script de buen ciudadano que para torrent cuando alcanza índice de compartición 1.
- samba para poder acceder a los ficheros, por ejemplo desde un reproductor multimedia de red.

Capítulo 9

Sugerencia de línea de proyectos innovadores

Nos encantaría que con esta introducción al mundo de los embebidos en GNU/Linux, a alguna persona le entre el gusanillo y con el tiempo de el salto a innovador. Para ir animando, una posible línea de ideas en las que pensar.

Se trata de pensar en funcionalidades que se ejecuten normalmente en un ordenador, pero que vamos a ejecutarlo en un dispositivo, con las siguientes ventajas:

- precio
- facilidad de manejo: debería ser como un electrodoméstico
- no hace ruido, no ocupa casi espacio, apenas consume, en definitiva, puede estar muchas horas encendido

Es decir, una línea de ideas es pensar en cosas que a lo mejor no son nuevas y ya se podrían hacer con un PC, pero se podrían hacer más fáciles y asequibles, sin necesidad de dejar el ordenador encendido.

No todas las ideas tienen que consistir en dispositivos que se usan directamente por las personas: también pueden ser para interactuar con otros dispositivos o máquinas (por ejemplo para acceder desde un teléfono móvil o desde el reproductor de un televisor). Ahí la buena y mala noticia es que muchas de esas ideas pueden implantarse también sin necesidad de dejar el ordenador encendido mediante una virtual appliance que se ejecute en el cloud en lugar de en un dispositivo: la mala es que el cloud puede competir con nuestra solución hardware, la buena es que si la idea es nueva y buena eso es lo que importa y no cómo llevarla a cabo técnicamente, de modo cuantas más posibilidades mejor. Además también podemos concebir una solución mixta, en que parte de la solución se ejecute en la nube y parte en local por tener que interactuar con dispositivos locales en casa del usuario.

Algunas posibilidades técnicas interesantes:

- hay ya varios proyectos combinando sistemas embebidos y software PBX o en general de comunicaciones de voz, con los que podemos tener todo tipo de posibilidades, como contestadores más avanzados tipo secretaria, posibilidad de redigirnos por VoIP si la llamada es importante...
- mediante Jabber podemos ejecutar cliente siempre conectado a servidor, con conexión olvidándose de si hay NAT y habría que abrir puertos, si cambia la IP etc.
- mediante wake-on-lan se puede arrancar ordenador si es necesario y mediante ssh redirigir cualquier conexión
- mediante la combinación de módulos FUSE y SAMBA, sería posible ejecutar comandos y cierta navegación a través de menús, en dispositivos que permitan acceder a carpetas Samba (por ejemplo cualquier reproductor multimedia con interfaz de red). Así, cabe navegar por un directorio virtual hasta un fichero de tipo JPEG que sea en realidad para tomar instantánea con la webcam de habitación del niño y verlo como una foto más aunque en realidad se ha generado al pedir abrirla. Hay posibilidades similares con generación de ficheros de audio e incluso con vídeo. Por ejemplo puede traducirse una fuente de noticias RSS generando la página web como una captura de pantalla o hacer un podcast.

Capítulo 10

Para saber más

Para estar al día:<http://www.linuxfordevices.com>

También puede ser interesante la publicidad en revistas como *linuxjournal* o buscar en Google por el nombre de los System on Chips; en la página de OpenWrt se pueden ver las distintas plataformas que se soportan. Los foros de los proyectos existentes también suele ser sitio dónde se comentan novedades cuando alguien descubre un nuevo dispositivo interesante que puede hackearse.

Linaro <http://www.linaro.com> iniciativa liderada por Ubuntu y que pretende también lanzar también software cada 6 meses que pueda ser utilizado de base para Android, Mee-Go u otros proyectos. No será una distribución, sino una base común para distribuciones. Es interesante que se centra en los dos procesadores tope de gama de ARM; es de esperar que surjan noticias de hardware de referencia y para desarrolladores.

Apéndice A

GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent

copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- Ñ. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.