

HermesAP Howto

Daniel Martínez Ponce
dmescal@madridwireless.net
Versión 1.0, 17/04/2003, DMP

Copyright (c) 2003 Daniel Martínez Ponce. Se otorga permiso para copiar, distribuir y/o modificar este documento según los términos de la Licencia GNU Para Documentación Libre (GNU Free Documentation License), versión 1.2 o cualquier versión posterior publicada por la Free Software Foundation. Esta licencia está disponible en <http://gnu.org/copyleft/fdl.html>.

Resumen

Este documento describe como configurar una tarjeta inalámbrica (Orinoco, Avaya, Wavelan, etc) en GNU/Linux, para ponerla en modo Master.

Descargo de responsabilidad

No me hago responsable en ningún caso de los posibles daños o pérdidas de garantía que pueda ocasionar el uso, debido o indebido, de la información existente en este documento. Si decides hacerme caso es tu problema. Acepto información diciendo: "me lo he cargado", pero no del tipo: "me lo he cargado por tu culpa".

Configuración

Para configurar HermesAP lo primero que tenemos que hacer, es bajarnos los fuentes de los siguientes drivers: [pcmcia-cs-3.2.4](#) y [HermesAP](#). Una vez que tenemos estos, tenemos que compilar el kernel con las siguientes opciones configuradas.

1. Desactivar el soporte PCMCIA del kernel, vamos a utilizar el driver PCMCIA compilando los fuentes, para luego a este driver aplicarle los módulos de HermesAP.

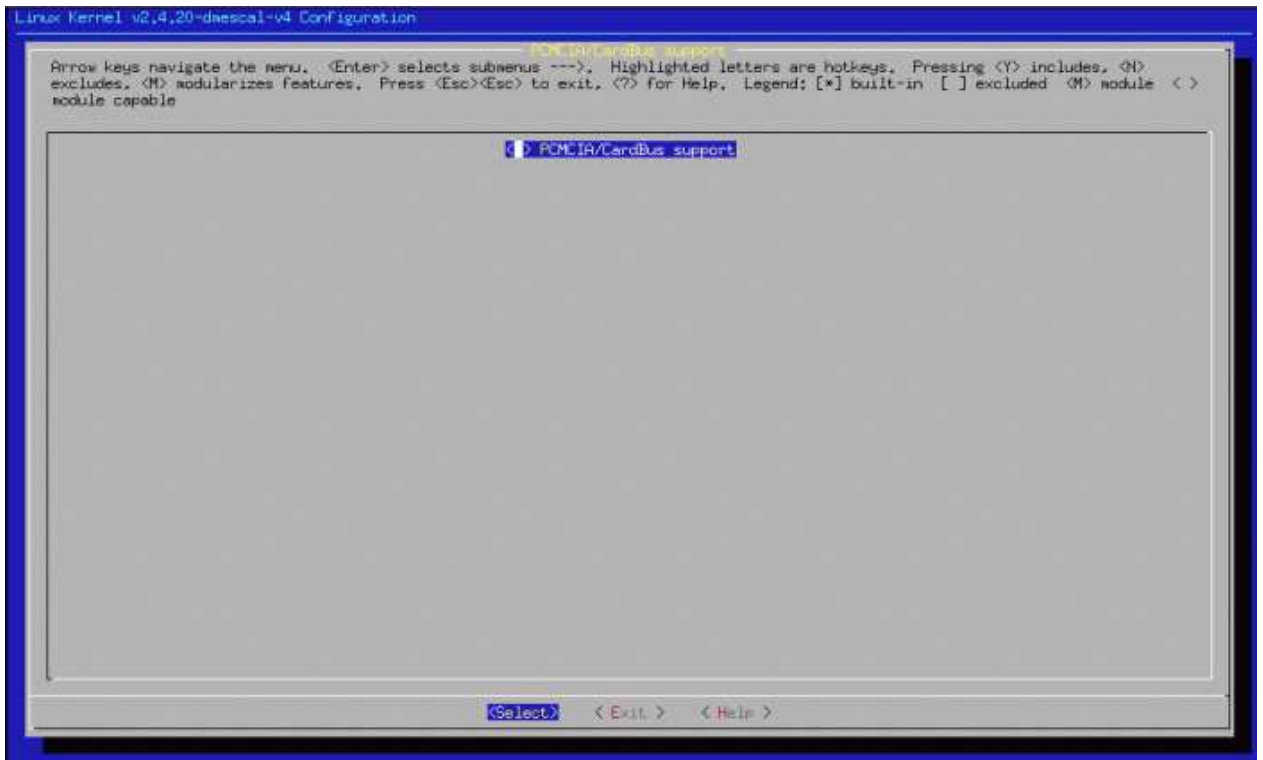


Fig. : General Setup/PCMCIA CardBus Support

2. Activar soporte wireless, pero sin activar ningún módulo de tarjetas wireless, para que los coja de los instalados por el driver PCMCIA.

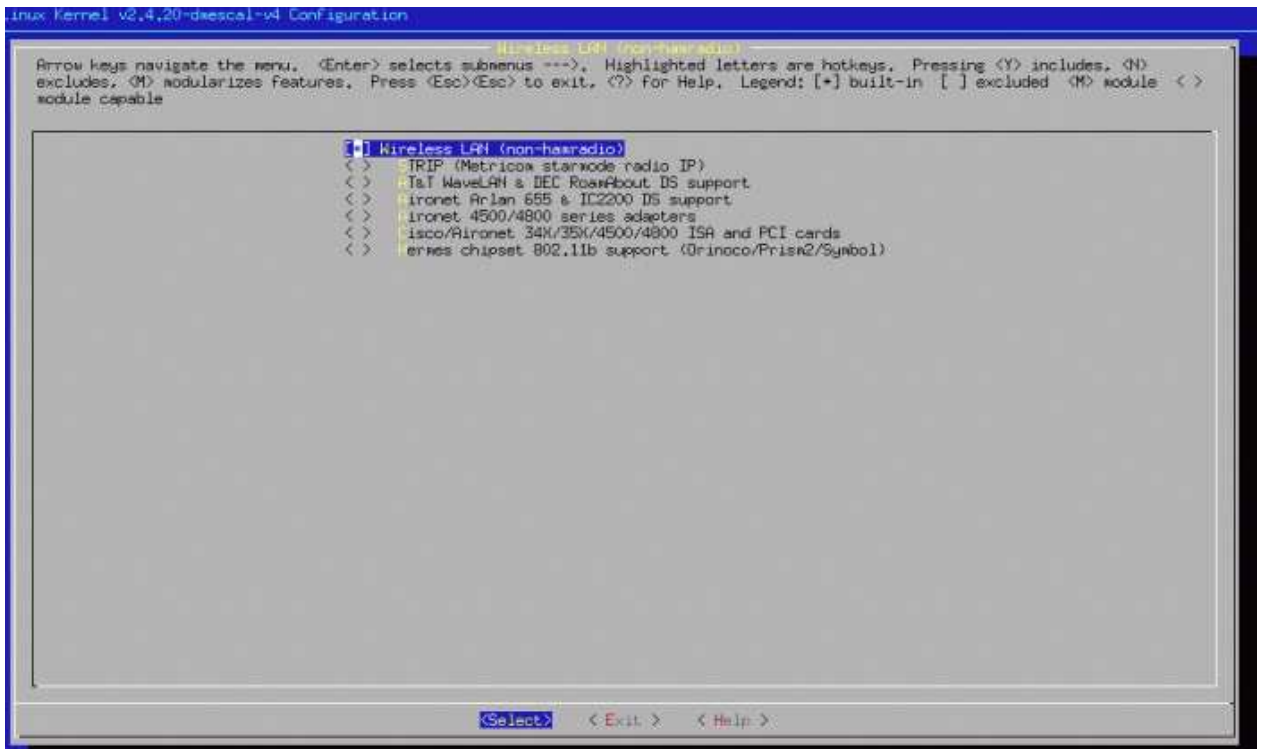


Fig. : Network device support/Wireless LAN (Non-hamradio)

3. Activar soporte devfs en el kernel, junto con las opciones "Automatically mount at boot" y "Debug devfs".

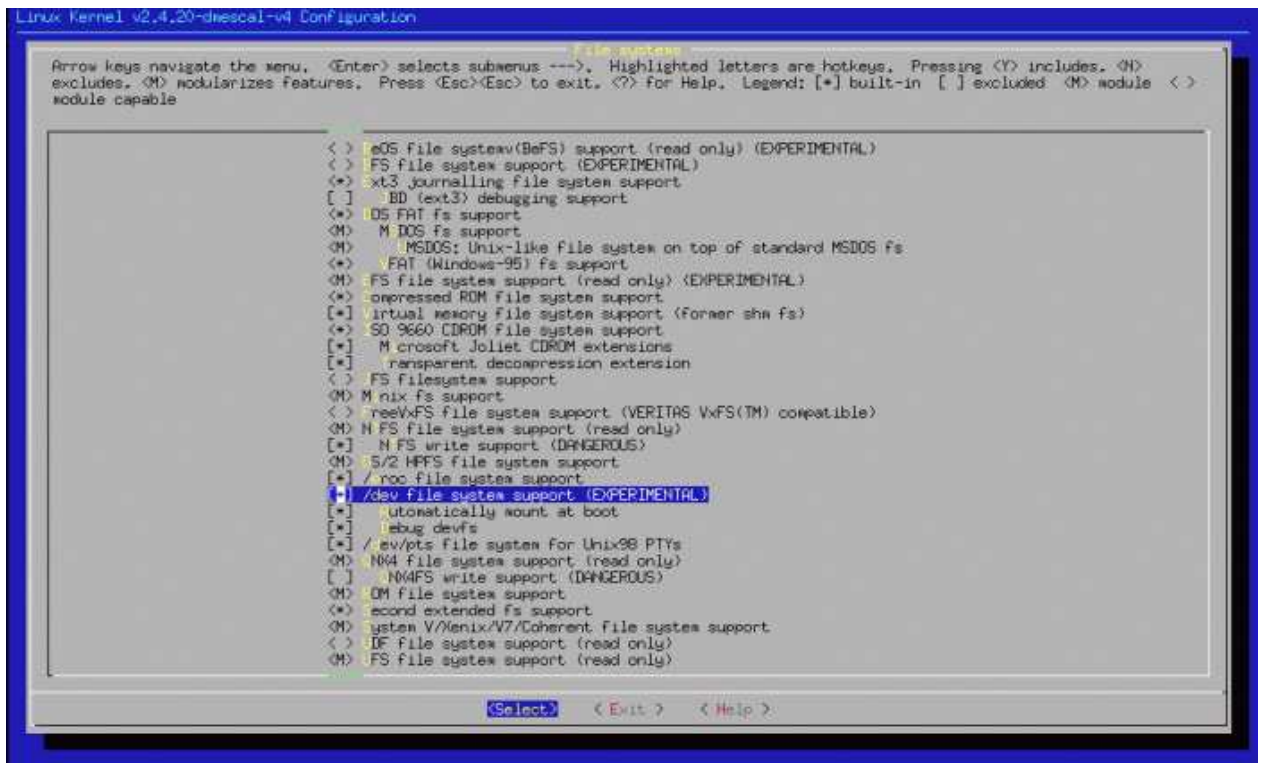


Fig. : File systems //dev file system support

Una vez compilado el kernel con estas opciones, nos instalamos "devfsd", si tienes una Debian, ya sabes! "apt-get install devfsd".

Ahora ya solo nos falta instalar el driver PCMCIA, con los módulos del HermesAP, para ello vamos a descomprimir los drivers PCMCIA y el HermesAP en /usr/src que previamente hemos descargado ahí. Para descomprimir el PCMCIA, simplemente "tar xvzf pcmcia-cs-3.2.4.tar.gz" y para el HermesAP "tar xvjf hermesap-0.1a.tar.bz2".

El driver HermesAP una vez descomprimido tiene varias carpetas, de momento sólo nos interesa la que contiene el driver, que en esta versión del driver la ruta es "/usr/src/hermesap-0.1/driver/orinoco-0.13b-hermesap-0.1a" y le borramos el Makefile, porque desde aquí no lo vamos a compilar, para borrarlo "rm Makefile" y ahora copiamos el contenido de "/usr/src/hermesap-0.1/driver/orinoco-0.13b-hermesap-0.1a" a "/usr/src/pcmcia-cs-3.2.4/wireless" es decir "cp * /usr/src/pcmcia-cs-3.2.4/wireless".

Con esto ya tenemos el driver PCMCIA con soporte HermesAP, ahora vamos a compilar e instalar el driver PCMCIA.

Instalación

```
almagr001:/usr/src/pcmcia-cs-3.2.4# make config
----- Linux PCMCIA Configuration Script -----

The default responses for each question are correct for most users.
Consult the PCMCIA-HOWTO for additional info about each option.

Linux kernel source directory [/usr/src/linux]:

The kernel source tree is version 2.4.20-dmescal-v4.
```

The current kernel build date is vie abr 12 06:15:07 2002.
WARNING: the source tree has a build date of jue abr 18 10:28:10 2002.
date: fecha inválida 'jue abr 18 10:28:10 2002'
date: fecha inválida 'vie abr 12 06:15:07 2002'
Did you forget to install your new kernel?

Build 'trusting' versions of card utilities (y/n) [n]:
Include 32-bit (CardBus) card support (y/n) [y]:
Include PnP BIOS resource checking (y/n) [n]:

The PCMCIA drivers need to be compiled to match the kernel they will be used with, or some or all of the modules may fail to load. If you are not sure what to do, please consult the PCMCIA-HOWTO.

How would you like to set kernel-specific options?
1 - Read from the currently running kernel
2 - Read from the Linux source tree
Enter option (1-2) [2]: 1

Module install directory [/lib/modules/2.4.20-dmescal-v4]:

Kernel configuration options:
Kernel-tree PCMCIA support is disabled.
Symmetric multiprocessing support is disabled.
Preemptive kernel support is disabled.
High memory support is disabled.
PCI BIOS support is enabled.
Power management (APM) support is enabled.
SCSI support is enabled.
IEEE 1394 (FireWire) support is disabled.
Networking support is enabled.
Radio network interface support is enabled.
Token Ring device support is enabled.
Fast switching is disabled.
Frame Diverter is disabled.
Module version checking is enabled.
Kernel debugging support is enabled.
Memory leak detection support is disabled.
Spinlock debugging is disabled.
Preemptive kernel patch is disabled.
/proc filesystem support is enabled.
PAE support is disabled.

The standalone Adaptec APA1480 CardBus driver is not supported with this kernel. If you need it, use the kernel PCMCIA subsystem.

The standalone IEEE 1394 CardBus drivers are not supported with this kernel. If you need them, use the kernel PCMCIA subsystem.

Your boot map file is older than /vmlinuz. If you installed /vmlinuz by hand, please run 'lilo' to update your boot data, and then reboot.
The Forms library is not available.
The X11/Xaw libraries are not available.
The GTK+ library is not available.

Configuration successful.

```
almagr001:/usr/src/pcmcia-cs-3.2.4# make all
.....
....
...
make[1]: Leaving directory `/usr/src/pcmcia-cs-3.2.4/debug-tools'
make[1]: Entering directory `/usr/src/pcmcia-cs-3.2.4/man'
make[1]: No se hace nada para 'all'.
make[1]: Leaving directory `/usr/src/pcmcia-cs-3.2.4/man'
make[1]: Entering directory `/usr/src/pcmcia-cs-3.2.4/etc'
make[2]: Entering directory `/usr/src/pcmcia-cs-3.2.4/etc/cis'
make[2]: No se hace nada para 'all'.
make[2]: Leaving directory `/usr/src/pcmcia-cs-3.2.4/etc/cis'
make[1]: Leaving directory `/usr/src/pcmcia-cs-3.2.4/etc'
almagr001:/usr/src/pcmcia-cs-3.2.4#
```

```
almagr001:/usr/src/pcmcia-cs-3.2.4# make install
.....
....
```

```
...
make[2]: Leaving directory `/usr/src/pcmcia-cs-3.2.4/etc/cis'
-> Installing PCMCIA startup script as /etc/rc.d/rc.pcmcia.N
-> Updating client scripts in /etc/pcmcia
-> Running depmod...
make[1]: Leaving directory `/usr/src/pcmcia-cs-3.2.4/etc'
```

Hasta este punto, ya tenemos instalado el driver PCMCIA, reiniciamos la máquina, aunque no es necesario, si es conveniente para evitarnos posibles errores con el montaje de la ethX, con el sistema de archivos devfs. Una vez reiniciada la máquina, la configuramos para que tenga salida a Internet, ya sea a través de la LAN o un modem, y vamos a ejecutar unos scripts que tiene el driver HermesAP.

Insertar un nuevo firmware en la tarjeta

```
almagr001:/usr/src/hermesap-0.1/firmware# ./hfwget.sh
--14:08:46-- ftp://ftp.avaya.com/incoming/Uplcku9/tsoweb/avayawireless/AV_WINXP_PC_USB_SR0201.zip
=> `AV_WINXP_PC_USB_SR0201.zip'
Resolviendo ftp.avaya.com... hecho.
Conectando con ftp.avaya.com[216.25.243.8]:21... conectado.
Identificándose como anonymous ... ¡Dentro!
==> SYST ... hecho. ==> PWD ... hecho.
==> TYPE I ... hecho. ==> CWD /incoming/Uplcku9/tsoweb/avayawireless ... hecho.
==> PASV ... hecho. ==> RETR AV_WINXP_PC_USB_SR0201.zip ... hecho.
Longitud: 6,211,983 (probablemente)
1% [>.....] 75,672 24.97K/s ETA 03:59
```

Este script lo que hace, es conectarse al ftp de Avaya y descargarse unos drives, los descomprime y extrae un binario, este es el que vamos a utilizar ahora para el firmware de la tarjeta. Nota: Este script necesita tener instalado unzip, "apt-get install unzip"

Tenemos todo, ya sólo queda hacer el proceso de flasheo de la tarjeta, por las pruebas que he hecho, he sacado la conclusión de que el firmware, no se actualiza físicamente en la tarjeta, lo hace mediante software, es decir monta el dispositivo wireless en el /dev y escribe en él, entonces linux se cree que el firmware es el que nosotros le indicamos. Para hacer esto hay un script que se encarga de ello, este es procedimiento a seguir:

Lo primero bajar la interface wireless para que se pueda escribir en la interface en el /dev

```
almagr001:/usr/src/hermesap-0.1/hfw# ifconfig eth1 down
```

Ahora lanzamos el script de flasheo de la tarjeta wireless (ethX)

```
almagr001:/usr/src/hermesap-0.1/hfw# ./hfwload eth1 ../firmware/T1085800.hfw
segm ofs 1F4800 len A400 dlen A400 unk 0
segm ofs 1FF000 len 1000 dlen 1000 unk 0
no matching PDA entry for plugrecord 00000150 2
no matching PDA entry for plugrecord 00000160 1C
no matching PDA entry for plugrecord 00000161 100
hfw_upload: 0
```

Nota: T1085800.hfw es el binario que saca el script de la descarga del zip.

Ya lo tenemos funcionando :)

```

almagr001:/usr/src/hermesap-0.1/hfw# iwconfig
lo          no wireless extensions.

eth0        no wireless extensions.

eth1        IEEE 802.11-DS  ESSID:"non-specified SSID !!"  Nickname:"pilar004"
Mode:Master  Frequency:2.422GHz  Access Point: 00:60:1D:1E:84:F8
Bit Rate:11Mb/s  Tx-Power=15 dBm  Sensitivity:1/3
Retry limit:4  RTS thr:off  Fragment thr:off
Encryption key:off
Power Management:off

```

Antes de levantar la interface wireless le asignamos un essid.

```

almagr001:/usr/src/hermesap-0.1/hfw# iwconfig eth1 essid madridwireless

```

Levantamos la interface.

```

almagr001:/usr/src/hermesap-0.1/hfw# ifconfig eth1 up
almagr001:/usr/src/hermesap-0.1/hfw# ifconfig

eth0        Link encap:Ethernet  HWaddr 00:10:5A:D4:3D:6E
            inet addr:192.168.10.129  Bcast:192.168.10.255  Mask:255.255.255.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:733 errors:0 dropped:0 overruns:0 carrier:733
            collisions:0 txqueuelen:100
            RX bytes:0 (0.0 b)  TX bytes:18702 (18.2 KiB)
            Interrupt:3 Base address:0x300

eth1        Link encap:Ethernet  HWaddr 00:60:1D:1E:84:F8
            inet addr:10.64.2.2  Bcast:10.64.2.31  Mask:255.255.255.224
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:12460 errors:0 dropped:0 overruns:0 frame:0
            TX packets:15638 errors:16 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:100
            RX bytes:3319235 (3.1 MiB)  TX bytes:2456758 (2.3 MiB)
            Interrupt:10 Base address:0x100

lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:439 errors:0 dropped:0 overruns:0 frame:0
            TX packets:439 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:43438 (42.4 KiB)  TX bytes:43438 (42.4 KiB)

almagr001:/usr/src/hermesap-0.1/hfw# iwconfig

lo          no wireless extensions.

eth0        no wireless extensions.

eth1        IEEE 802.11-DS  ESSID:"madridwireless"  Nickname:"pilar004"
Mode:Master  Frequency:2.457GHz  Access Point: 00:60:1D:1E:84:F8
Bit Rate:14Mb/s  Tx-Power=15 dBm  Sensitivity:1/3
Retry limit:4  RTS thr:off  Fragment thr:off
Encryption key:off
Power Management:off
Link Quality:0/92  Signal level:-40 dBm  Noise level:-49 dBm
Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:4
Tx excessive retries:16  Invalid misc:0  Missed beacon:0

```

Conclusiones

La versión que estamos usando de HermesAP soporta esto:

- WEP support (hope it works 8))
- hidden mode
- (set channel and ssid - but that's quite basic, isn't it? ;))

Y le falta estas opciones por solucionar:

- no monitoring mode
- no ACL support
- no WDS support yes
- not (very) well tested

Pruebas que me faltan hacer a mi:

- Probar el driver con bridge ISA, PCI y adaptador PLX.
- Probar varias tarjetas sobre la misma máquina.
- Intentar hacer funcionar el driver HermesAP en USrobotic 2450 con linuxAP y quitando el HostAP.