



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

## Configuración de la DWL-122 (40924 lectures)

Per **Laura Primo Monlleó**, [Laura](http://linuv.uv.es) (<http://linuv.uv.es>)

Creado el 17/04/2005 00:53 modificado el 25/05/2005 17:51

*En este artículo trataré de describir los pasos que llevé a cabo para lograr la integración de la tarjeta D-Link DWL-122 en una Ubuntu Warty para la arquitectura PowerPC de un Ibook G4.  
**Actualización:** Comento cómo hacer esta misma configuración en Ubuntu Hoary solventando los problemas surgidos con la nueva versión del driver linux-wlan-ng.*

## Introducción

Actualmente los Ibook y los Power PC de la serie G4 vienen con una Tarjeta AirPort Extreme que es compatible con cualquier red inalámbrica encriptada o no, que siga el estándar [802.11b/g](#)<sup>(1)</sup>.

Lamentablemente su fabricante, la empresa [Broadcom](#)<sup>(2)</sup>, no ha creado controladores para Linux para ella ni tampoco ha facilitado los suficientes datos para que sea posible su integración en nuestro sistema operativo. En este punto cabe comentar que en los powerpc de la serie G3 la tarjeta integrada si es reconocida sin problemas irresolubles por los controladores para linux.

Por lo que los usuarios que tenemos GNU/Linux instalado en nuestro power pc no podemos hacer uso de este dispositivo. Así el único medio para acceder a las redes wifi de nuestro entorno, es adquiriendo otro adaptador de red que sí sea compatible.

La elección no es nada sencilla, ya que las particularidades de este portátil hacen difícil la operabilidad de muchas de las tarjetas que están a la venta actualmente. Por ejemplo, al menos en los portátiles tipo [Ibook](#)<sup>(3)</sup>, no se puede contar con las tarjetas PCMCIA, ya que no tenemos bahía para conectarlas.

Sólo queda la solución de encontrar una tarjeta tipo USB que se adapte a nuestro querido ibook. Haberlas haylas como dicen por ahí, pero no todas las que quisiéramos. Cuando empecé a buscar por internet me di cuenta de que en muchos foros, listas y algunas webs, relacionadas con este tema, recomendaban un par de tarjetas muy concretas, la [Netgear modelo MA111](#)<sup>(4)</sup> y la [DWL-122 de Dlink](#)<sup>(5)</sup>. Así parecía que la elección no era tan complicada.

Finalmente, yo me decanté por la de Dlink, porque en algunas páginas donde se hablaba del tema se decía que el chipset de la Netgear actual, había cambiado y ya no era tan "fácil" usarla bajo Linux.

## Características de la DWL-122

El adaptador DWL-122 de D-Link utiliza un chipset tipo Prism de Intersil, lo que permite que sea configurable con los controladores [Linux-Wlang-ng](#)<sup>(6)</sup>.

También, gracias a estos controladores se puede acceder a redes encriptadas de 64/128 bits que usen el estándar 802.11b.

La DWL-122 está preparada para conectarse a redes o routers hasta una velocidad de transmisión de 11Mbps, lo que hace que este adaptador resulte algo lento comparado con las tarjetas pcmcia y los adaptadores tipo USB para otro tipo de portátiles. El usb del que dispone sigue el estándar 1.1 cuya velocidad de transmisión es menor que si fuera un usb versión 2.



Los tipos de arquitectura que soporta la DWL según el fabricante son los modos Ad-Hoc(punto a punto sin Punto de Acceso) y el modo Managed o estructurado (redes comunicadas con Roaming via Punto de Acceso).

## Instalación

**MacOS:** El adaptador viene acompañado de un [manual](#)<sup>(7)</sup> y de un cdrom donde están los drivers listos para instalarla, en este sistema operativo. Así que siguiendo los pasos que se describen en el libro o simplemente siguiendo el sentido común se puede llegar a instalar con gran facilidad a golpe de clics de ratón. He de hacer notar que si bien es muy sencilla su instalación y su uso en redes abiertas, es decir, sin encriptación. Si lo que se desea es acceder a una red encriptada simplemente se ha de habilitar la WEP en las propiedades de la tarjeta, dentro del programa de configuración. Tras esto, volver a conectar a la red, ya encriptada. Si se pretende conectar a una red de 64bits, elegir la encriptación de 40Bits Hex (10 dígitos), si por el contrario tienes una red de 128Bits, se ha de elegir la de 104Bits Hex (26 dígitos).

**GNU/Linux:** Mi sistema Linux, corresponde a una [Ubuntu Warty](#)<sup>(8)</sup> para powerpc, con un kernel 2.6.8.1-3-powerpc. Como he comentado antes a este adaptador de red le corresponde como controlador en Linux los [Linux-Wlang](#)<sup>(9)</sup>, por ello tras su instalación mediante apt-get, se configura siguiendo su [readme](#)<sup>(10)</sup>.

## Instalación en Debian Ubuntu

- *Módulos en el kernel*

En primer lugar se ha de pensar en si el kernel está preparado para administrar dispositivos tipo USB. Esto se puede saber leyendo el config propio del kernel que está activo. En este caso, este archivo reside en la carpeta /boot, así con un `cat /boot/config-2.6.8.1-3-powerpc |grep USB`, se puede encontrar la lista de módulos que contiene el kernel y tratan sobre el USB, preparados para activar con una simple ejecución de un **modprobe módulo\_a\_activar** siendo root.

En Ubuntu casi todos estos detalles estan previstos, para bien o para mal. Pero en otras distribuciones con kernels 2.6.x, el config debe contener los siguientes parámetros si se desea que esté soportado el USB:

```
CONFIG_USB_DEVICEFS=y
CONFIG_USB=y
CONFIG_USB_EHCI_HCD=m
CONFIG_USB_UHCI_HCD=m
CONFIG_USB_OHCI_HCD=m
CONFIG_USB_HID=m
CONFIG_USB_HIDINPUT=m
```

En caso de que no esten presentes estos módulos será necesario compilar el kernel. Lo mismo, si no tienes los módulos Prism.

```
root@ibook:/home/laura # grep PRISM /boot/config-2.6.8.1-3-powerpc
CONFIG_PRISM2=m
CONFIG_PRISM2_USB=m
CONFIG_PRISM54=m
root@ibook:/home/laura #
```

- Instalando Linux-Wlang

Para instalar este driver se puede optar bien por bajar el [código fuente](#)<sup>(11)</sup>, que tras compilar el kernel con los módulos necesarios y dejar las fuentes instaladas en el sistema, se puede instalar siguiendo estos pasos:

Ejecutar **make config** en el directorio creado tras la descompresión de los sources, tras esto aparecerán una serie de preguntas cuyas respuestas serán la base de la configuración de los controladores. La respuesta por defecto es la que está entre corchetes, para seleccionarla basta con pulsar la tecla .

```
- "Build Prism2.x PCMCIA Card Services (_cs) driver? (y/n) [y]:
Select "y" if you want to build the Prism PCMCIA driver.
```



If you select "n", the PCMCIA related questions below will not be asked.

- Build Prism2 PLX9052 based PCI (`_plx`) adapter driver? (y/n) [y]:  
Select "y" if you want to build the Prism driver for PLX PCI9052 PCI/PCMCIA adapter based solutions.

- Build Prism2.5 native PCI (`_pci`) driver? (y/n) [y]:  
Select "y" if you want to build the Prism driver for Prism2.5 ISL3874 based native PCI cards. This includes PCI add-in cards and the mini-pci modules included in some notebook computers (but not all, some use internal USB modules).

- Build Prism2.5 USB (`_usb`) driver? (y/n) [y]:  
Select "y" if you want to build the Prism driver for Prism2.5 ISL3873 based USB adapters. This includes USB add-on modules and the internal modules included in some notebook computers.

- Linux source directory [`/usr/src/linux`]:  
The config script will attempt to automatically find your kernel source directory. If found, the kernel source directory will be presented as the default selection. If the default selection is wrong, you may correct it here.

- pcmcia-cs source dir [`/usr/src/pcmcia-cs-3.1.29`]:  
If the "`_cs`" driver is selected above, the configure script will attempt to present a reasonable default for the pcmcia source directory. If the presented directory is incorrect, you may change it here. If the "`_cs`" driver is not selected, this prompt will not appear.

- PCMCIA script directory [`/etc/pcmcia`]:  
If the "`_cs`" driver is selected, this prompt allows you to change the location where the pcmcia scripts will be installed. Only do this if you have installed the rest of the `pcmcia_cs` scripts to a non-default location.

- Alternate target install root directory on host []:  
This prompt allows you to specify an alternative root directory for the install process.

- Module install directory [`/lib/modules/2.6.x`]:  
Select where you want the driver modules to be installed. The script constructs a default location using the output of `uname`. If you have not yet installed the kernel you will run `linux-wlan` with, and the new kernel has a different version string, you will need to change this value.

- Prefix for build host compiler? (rarely needed) []:  
When cross-compiling or using different compilers for kernel and user-mode software, it is sometimes (but rarely) necessary to specify a different compiler prefix to use when compiling the `_tools_` that are built to run on the build host during the `linux-wlan-ng` build process.

- Build for debugging (see `doc/config.debug`) (y/n) [y]:  
This option enables the inclusion of debug output generating statements in the driver code. Note that enabling those statements



requires the inclusion of insmod/modprobe command line arguments when loading the modules. See the document doc/config.debug for more information.

Una vez respondidas las preguntas sin que se haya escapado del programa por algún error, se construye el paquete con un **make all**. Ahora si todo marcha bien, se ha de instalar haciendo **make install**, siendo root.

En mi caso, al tener un kernel precompilado no pude completar este cuestionario, ya que no podía encontrar las fuentes de un kernel que no había compilado personalmente. Así intenté en vano instalar los headers de mi kernel mediante apt-get:

```
apt-get install linux-headers-2.6.8.1-3
```

Como tampoco parecía funcionar el tema, así que decidí darle una oportunidad al apt-get, me puse a buscar y hallé lo que buscaba, es decir, el paquete linux-wlang para debian-ubuntu:

```
root@ibook:/usr/src/# apt-cache search linux-wlan
hostap-source - Host AP driver for Intersil Prism2/2.5/3
linux-wlan-ng - utilities for wireless prism2 cards
linux-wlan-ng-doc - documentation for wlan-ng
```

Como se ve en la búsqueda hecha por apt-cache hay posibilidad de instalar 3 programas relacionados con los controladores linux-wlan, opté por el linux-wlan-ng:

```
root@ibook:/usr/src/# apt-get install linux-wlan-ng
```

Se instaló y se configuró en unos segundos, tras los cuales ya era operativo el programa, ahora restaba configurarlo adecuadamente en mi sistema.

- Cargando los módulos del Kernel

Siguiendo las instrucciones de los drivers Linux-wlang, se activa el módulo **prism2\_usb** pasándole el parámetro **prism2\_doreset=1**, de este modo:

```
modprobe prism2_usb prism2_doreset=1
```

En efecto se comprueba que se carga dicho módulo:

```
root@ibook:/home/laura # lsmod |grep usb
prism2_usb 84748 0
p80211 42580 1 prism2_usb
usbhid 53952 0
usbcore 139092 6 ehci_hcd,prism2_usb,usbhid,ohci_hcd
```

Con la finalidad de que los módulos sean cargados tras cada reinicio y antes de que trate de conectar a internet, se modificará el /etc/modules, añadiendo, al listado, el nombre y los parámetros del módulo a cargar. De este modo, el sistema leerá este archivo y creará el /etc/modules.conf que a su vez será leído por el sistema de arranque de debian, consiguiendo que esos módulos esten preparados para el usuario de manera automática.

- Configuración básica de los Linux-wlang

Los archivos de configuración de wlang se hallan en /etc/wlan. Uno de los que es necesario modificar es el /etc/wlan/wlan.conf . Los cambios necesarios se efectuan sobre estas líneas:

```
WLAN_DEVICES="wlan0"
SSID_wlan0="nombred"
ENABLE_wlan0=y
#SSID_wlan1=""
#ENABLE_wlan1=n
```



```
#SSID_wlan2=""
#ENABLE_wlan2=n
```

En definitiva sólo hay que decirle que nombre tiene la red a la que queremos conectarnos y el nombre del dispositivo que se ha de activar, en este caso será wlan0. El resto de líneas no se han de modificar si no se dispone de más dispositivos wireless.

Luego creamos un nuevo archivo que será una copia modificada del que hay como ejemplo en el mismo directorio /etc/wlan/. Así haremos:

```
root@ibook:/etc/wlan $ cp /etc/wlan/wlancfg-DEFAULT
/etc/wlan/wlancfg-nombrered
```

Donde el nombre de la red ha de coincidir con la que se ha detallado en el /etc/wlan/wlan.conf en las líneas que hemos modificado.

Este es mi archivo /etc/wlan/wlancfg-nombrered:

```
#=====SELECT STATION MODE=====
IS_ADHOC=n # y|n, y - adhoc, n - infrastructure
#===== INFRASTRUCTURE STATION =====
# What kind of authentication?
AuthType="opensystem" # opensystem | sharedkey (requires WEP)
#===== ADHOC STATION =====
BCNINT=100 # Beacon interval (in Kus)
CHANNEL=6 # DS channel for BSS (1-14, depends
# on regulatory domain)
###BASICRATES="2 4" # Rates for mgmt&ctl frames (in 500Kb/s)
BASICRATES="22" # Rates for mgmt&ctl frames (in 500Kb/s)
OPRATES="2 4 11 22" # Supported rates in BSS (in 500Kb/s)
-----
```

En este archivo de configuración, le decimos a los controladores que queremos que nuestra red sea de tipo *infrastructure o managed*, es decir, todos los ordenadores de la red se comunican a través del punto de acceso (AP), se trata de una red centralizada. Si quisiéramos que nuestra red no dependiera de un solo AP, se marcaría con la opción *y*, para habilitar el modo Ad Hoc.

Si se lee el readme de los controladores, se encuentra con que para agilizar la configuración de la tarjeta, se puede copiar un script de inicio en la carpeta correspondiente, por ejemplo en nuestro sistema se hace lo siguiente:

```
root@ibook:/home/laura # cp /usr/share/doc/linux-wlan-ng/examples/rc.wlan
/etc/init.d/wlan
```

Tras lo cual ejecutamos el script del modo habitual, es decir, nombre servicio y su PATH luego la opción deseada: start, stop, reload... iniciamos el servicio:

```
root@ibook:/home/laura # /etc/init.d/wlan start
```

Este script busca y ejecuta el /etc/wlan/shared que a su vez, activa los dispositivos wlan con la configuración indicada a tal fin en los archivos de configuración arriba descritos. Este script se puede añadir a la lista de servicios que arrancan con el sistema, aunque se podría decir que no es muy conveniente, pero explicaré como podría haberlo hecho.

El programa que se ha de usar es el *rcconfig*, en *debian* es tan fácil como esto:

```
apt-get install rcconf
Ejecutarlo desde terminal como root así:
root@ibook:/home/laura # rcconf
```

Aparecerá un menú en *ncurses* desde el cual se podrá añadir o eliminar cualquier programa de la lista.



Los comandos que se van a utilizar para la configuración de la conexión serán únicamente los derivados del `wlanctl-ng`, para conocer la lista de posibles comandos e instrucciones, basta ejecutar: `wlanctl-ng commands`

```
root@ibook:~ # wlan

wlancfg wlanctl-ng wlan

root@ibook:~ # wlancfg
Query, show, or set configuration settings.
Usage: wlancfg query dev
wlancfg show dev [all]
wlancfg set dev
wlancfg list
wlancfg version
where: dev - Name of device (e.g. wlan0).
```

Con esto ya está configurada la red para la conexión sin encriptación. Ahora hay que realizar una serie de pasos sistemáticos para realizar la conexión, estos son los comandos con sus opciones necesarios:

```
root@ibook:~ # modprobe prism2_usb prism2_doreset=1
root@ibook:~ # wlanctl-ng wlan0 lnxreq_ifstate ifstate=enable
message=lnxreq_ifstate
ifstate=enable
resultcode=success
root@ibook:~ # wlanctl-ng wlan0 lnxreq_autojoin ssid=MyWlan
authtype=opensystem
message=lnxreq_autojoin
ssid='nombred'
authtype=opensystem
resultcode=success
```

La prueba de que la conexión es operativa la muestra la propia tarjeta, ya que sus dos diodos se encienden sin parpadear. Pero veamos que dice el comando `iwconfig`:

```
root@ibook:~ # iwconfig
lo no wireless extensions.
eth0 no wireless extensions.
sit0 no wireless extensions.
wlan0 IEEE 802.11-DS ESSID:"nombred" Nickname:"nombred"
Mode:Managed Frequency:2.462GHz Access Point: XX:XX:XX:XX:XX:XX
Bit Rate:2Mb/s Tx-Power:2346 dBm
Retry min limit:8 RTS thr:off Fragment thr:off
Encryption key:off
Link Quality=92/92 Signal level=61/154 Noise level=4/154
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

Efectivamente tenemos un adaptador en modo managed conectado a una red llamada `nombred`, usando el usuario `nombred`, ya que `iwconfig` toma como nickname el valor del ESSID. El punto de acceso tiene una dirección MAC con el formato `XX:XX:XX:XX:XX:XX`. La conexión está sin encriptar y la velocidad de transmisión es de unos 2Mb/s. Curioso que no pueda acceder a una velocidad mayor, siendo como es la tarjeta de 11Mb/s (esto ocurre cuando no se arranca el servicio wlan anteriormente citado).

Si repetimos la comprobación con `ifconfig`, obtenemos lo siguiente:

```
root@ibook:~ # ifconfig wlan0
wlan0 Link encap:Ethernet HWaddr YY:YY:YY:YY:YY:YY
BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```



```
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
```

Hasta aquí la configuración básica para conseguir conectarse a una red sin encriptar con una Dlink DWL-122. En algunas webs y foros también aconsejan que se modifique el `/etc/network/interfaces` del siguiente modo:

```
#RED WIRELESS DWL 122
iface wlan0 inet dhcp
name Targeta de la xarxa sense fils
wireless_essid nombrered
#auto wlan0
up /sbin/wlanctl-ng wlan0 lnxreq_ifstate ifstate=enable
#para recargar la red y que las modificaciones tomen efecto ejecutar:
/etc/init.d/networking restart
```

- Encriptación de la red a 64bits

Aunque es sabido que no hay suficiente seguridad como para evitar ser vulnerados tarde o temprano, nunca está de más tratar de cerrar un poco nuestros sistemas de redes wireless.

Bien, pues este es el proceso más complejo que tiene lugar a la hora de configurar una tarjeta de este tipo. Aquí detallo los pasos a seguir si se pretende tener una red encriptada a 64bits. Para encriptación a 128bits lo explico un poco más abajo.

Para usar este adaptador con encriptación de la señal, se debe modificar el `/etc/wlan/wlancfg-nombrered`, que se ha generado anteriormente. Este es un ejemplo, que creo será válido para muchos casos:

```
#====USER MIB SETTINGS=====
# You can add the assignments for various MIB items
# of your choosing to this variable, separated by# whitespace.
# The wlan-ng script will then set each one.
# Just uncomment the variable and set the assignments# the way you want
them.
#USER_MIBS="p2CnfRoamingMode=1 p2CnfShortPreamble=mixed"

#=====WEP=====
# [Dis/En]able WEP. Settings only matter if PrivacyInvoked is true
lnxreq_hostWEPEncrypt=true # true|false
lnxreq_hostWEPDecrypt=true # true|false
dot11PrivacyInvoked=true # true|false
dot11WEPDefaultKeyID=3 # 0|1|2|3
dot11ExcludeUnencrypted=true # true|false, in AP this means WEP is
required.

# If PRIV_GENSTR is not empty, use PRIV_GENTSTR to generate
# keys (just a convenience)PRIV_GENERATOR=/sbin/nwepgen
# nwepgen, Neesus compatiblePRIV_KEY128=false
# keylength to generatePRIV_GENSTR=""
# or set them explicitly. Set genstr or keys, not
both.dot11WEPDefaultKey0=3F:45:4I:3B:36
# format: xx:xx:xx:xx:xx or
dot11WEPDefaultKey1=4F:3C:4F:13:42
# xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
dot11WEPDefaultKey2="4A:5H:G5:54:23"
# e.g. 01:20:03:40:05 or dot11WEPDefaultKey3="22:B2:93:OT:CW"
# 01:02:03:04:05:06:07:08:09:0a:0b:0c:0d
#==SELECT STATIONMODE==#
```



```
IS_ADHOC=n # y|n, y - adhoc, n - infrastructure
#===== INFRASTRUCTURE STATION ==#
# What kind of authentication?
AuthType="sharedkey" # opensystem | sharedkey (requires WEP)
#=====ADHOC STATION =====BCNINT=100
# Beacon interval (in Kus)
CHANNEL=6 # DS channel for BSS (1-14, depends
# on regulatory domain)
###BASICRATES="2 4" # Rates for mgmt&ctl frames (in 500Kb/s)
BASICRATES="22" # Rates for mgmt&ctl frames (in 500Kb/s)OPRATES="2 4 11
22"
# Supported rates in BSS (in 500Kb/s)
```

- Encriptación de la red a 128bits

Esta es por definición algo más segura que la de 64 bits, pero como siempre hay que saber que ningún sistema es suficientemente seguro. Estos son los cambios que se han de hacer respecto a la anterior para obtener una encriptación de 128.

Conseguir las 4 claves, que serán del estilo a esta:

```
01:02:03:04:05:06:07:08:09:0a:0b:0c:0d
```

Editar el `/etc/wlan/wlancfg-nombrered`, descomentar y cambiar la opción `false` por `true` en la línea:

```
PRIV_KEY128=true # keylength to generate
```

En el mismo archivo poner las 4 claves generadas para conectarse a la red:

```
dot11WEPDefaultKey0=34:10:99:52:G2:80:42:6D:41:4C:V2:D0
# format: xx:xx:xx:xx:xx or
dot11WEPDefaultKey1=F6:92:13:31:OP:69:89:GC:D9:PL:DB:A1:85
# xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
dot11WEPDefaultKey2=F8:B2:97:FT:PF:DW:69:71:3A:49:8D:5F:28
# e.g. 01:20:03:40:05 or
dot11WEPDefaultKey3=72:D8:80:C1:FF:77:72:F8:90:84:60:C7:77
# 01:02:03:04:05:06:07:08:09:0a:0b:0c:0d
```

Modificar también `/etc/network/interfaces` con una de las claves para 128bits:

```
wireless_key 82:F8:P6:OD:F9:27:LA:UR:A2:8P:62:G7:97
```

- Automatizando el proceso de conexión

El primer paso es recoger todos estos comandos que se han ejecutado en un pequeño script en BASH:

```
#!/bin/bash

#Arrancando el servicio wlan

/etc/init.d/wlan start
#pasando parámetros al wlanctl-ng para activar la tarjeta 802.11 que
tenemos, para más detalles leer el man wlanctl-ng

wlanctl-ng wlan0 lnxreq_ifstate ifstate=enable

#Si se opta por no tener encriptación, descomentar esta línea

###wlanctl-ng wlan0 lnxreq_autojoin ssid=nombrered authtype=opensystem

#Si por el contrario sí quieres encriptarla, descomenta esta otra. Las dos
al mismo tiempo son #incompatibles
```



```
###wlanctl-ng wlan0 lnxreq_autojoin ssid=nombred red authtype=sharedkey

#Configuramos la tarjeta segun los parámetros indicados en
/etc/network/interfaces

ifup wlan0

#hacemos que nos muestre la información de que dispone sobre la wlan0

iwconfig wlan0

#Se comunica con el AP para que le facilite una dirección IP dinámica.

dhclient3 wlan0
```

Cerramos el archivo que contiene el script, guardando y dando permisos de ejecución:

```
chmod u+x script.sh.
```

Como el contenido implica la ejecución de algunos binarios a los que los usuarios normales no tienen acceso, el único que podrá ejecutar este script será el root del sistema o un usuario al que le esté permitido ejecutar comandos mediante la orden `sudo`.

- Comentarios

Actualización 25 de Mayo de 2005. Hace poco reinstalé el sistema usando en esta ocasión Ubuntu Hoary, la siguiente versión estable de Ubuntu para ppc.

La instalación de los módulos y del driver en la hoary, no presenta ninguna novedad respecto de la versión que comento en este artículo, pero hay problemas con la versión del driver que de manera automática se instala con `apt-get` empleando los sources que vienen con hoary. Para solventar este pequeño problema he recurrido a un truco, que quizá alguno considere algo burdo, pero que de momento está funcionando, a continuación relato los pasos más importantes que seguí.

### Instalación de linux-wlan-ng con Hoary, pero usando la versión que venia con Warty

1. Realizar una copia de seguridad del fichero de configuración de las fuentes de paquetes dada con Ubuntu Hoary.

```
root@/etc/apt/$cp sources.list sources.list-hoary
```

2. Editar el `sources.list`, como se observa los servidores tienen la orden de tomar los paquetes de la versión Hoary.
3. Comentar o borrar los servidores de seguridad, para evitar conflictos entre las versiones en programas demasiado vitales.
4. Cambiamos hoary por warty, en los servidores que mantengamos activos.
5. Actualizamos la configuración del apt:
 

```
root@/etc/apt/$apt-get update
```
6. Ejecutamos el instalador del apt
 

```
root@/etc/apt/$apt-get install linux-wlan-ng
```
7. Una vez instalado el controlador de la tarjeta wireless, es muy muy conveniente volver el apt a su configuración original, pues habría graves problemas de no hacer este paso, en posteriores instalaciones de programas. Este proceso de vuelta a Hoary se realiza deshaciendo los pasos comentados anteriormente, o bien restaurando la copia de seguridad que hemos hecho.
 

```
root@/etc/apt/$cp sources.list-hoary sources.list
```
8. Para acabar, volver a actualizar la configuración del apt:
 

```
root@/etc/apt/$apt-get update
```

De este modo hemos instalado en el sistema la versión anterior del driver con copia en los servidores de Ubuntu. Esto no quita que también se pueda [compilar el driver](#)<sup>(12)</sup>, con normalidad usando



cualquier otra versión intermedia, hasta ver cual es la última que no presenta este bug. No he probado con otras versiones, así que lo único que puedo asegurar es que siguiendo este minitruco, al ejecutar el script, que también en este artículo y si no hay ningún otro problema, la conexión a la red wifi será una realidad.

Añadir que estuve haciendo pruebas con programas como Wavemon y Kismet. El wavemon es para observar la calidad de la señal recibida y el kismet para detectar redes en los alrededores.

El proceso completo descrito aquí ha sido posible gracias a una labor de búsqueda a través de diversos medios como son listas de correo ([bulmailing](#)<sup>(13)</sup> de Bulma.net, [linux-ppc-es](#)<sup>(14)</sup> de Hispalinux, ...) y sobre todo un uso intensivo del buscador [google.com](#)<sup>(15)</sup>, donde pude hallar perlas como las siguientes:

<http://linuxbcn.homeip.net/node/88><sup>(16)</sup>  
<http://z.cwriter.org/index.php/wifi/Documentations/RSF+avec+Linux><sup>(17)</sup>  
<http://www.trustonme.net/didactels/296.html><sup>(18)</sup>  
<http://www.linuca.org/body.phtml?nIdNoticia=204><sup>(19)</sup>  
<http://www.seigan.org/entry.jsp?entry=484><sup>(20)</sup>

- Otros enlaces interesantes

<http://www.adslayuda.com/PNphpBB2+file-viewtopic-t-34940.html><sup>(21)</sup>  
[http://support.dlink.com/faq/view.asp?prod\\_id=480](http://support.dlink.com/faq/view.asp?prod_id=480)<sup>(22)</sup>  
[http://www.arturosoria.com/eprofecias/art/wireless\\_seguridad.asp](http://www.arturosoria.com/eprofecias/art/wireless_seguridad.asp)<sup>(23)</sup>  
[http://support.dlink.com/faq/view.asp?prod\\_id=480](http://support.dlink.com/faq/view.asp?prod_id=480)<sup>(22)</sup>  
<http://mandrake.badopi.org/node/view/1760><sup>(24)</sup>

- [tarjetas y chipsets wireless soportados en Linux](#)<sup>(25)</sup>

#### Lista de enlaces de este artículo:

1. [http://es.wikipedia.org/wiki/IEEE\\_802.11](http://es.wikipedia.org/wiki/IEEE_802.11)
2. <http://www.broadcom.com/index.php>
3. <http://www.apple.com/es/ibook/>
4. <http://www.netgear.com/products/details/MA111.php>
5. [http://www.dlink.com/images/products/DWL-122/DWL-122\\_main.gif](http://www.dlink.com/images/products/DWL-122/DWL-122_main.gif)
6. <http://linux-wlan.com/>
7. <http://www.dlink.com.au/ArticleDocuments/89/DWL-122%20Manual%20v1.01.pdf>
8. <http://www.ubuntulinux.org/>
9. <http://www.linux-wlan.com/>
10. <ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/README>
11. <ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/>
12. <http://www.ubuntulinux.org/wiki/HowToInstallTheWlanNgDriverInHoary>
13. <http://listes.bulma.net/pipermail/bulmailing/>
14. <https://listas.hispalinux.es/mailman/listinfo/linux-ppc-es>
15. <http://www.google.com>
16. <http://linuxbcn.homeip.net/node/88>
17. <http://z.cwriter.org/index.php/wifi/Documentations/RSF+avec+Linux>
18. <http://www.trustonme.net/didactels/296.html>
19. <http://www.linuca.org/body.phtml?nIdNoticia=204>
20. <http://www.seigan.org/entry.jsp?entry=484>
21. <http://www.adslayuda.com/PNphpBB2+file-viewtopic-t-34940.html>
22. [http://support.dlink.com/faq/view.asp?prod\\_id=480](http://support.dlink.com/faq/view.asp?prod_id=480)
23. [http://www.arturosoria.com/eprofecias/art/wireless\\_seguridad.asp](http://www.arturosoria.com/eprofecias/art/wireless_seguridad.asp)
24. <http://mandrake.badopi.org/node/view/1760>
25. [http://linux\\_wless.passys.nl/](http://linux_wless.passys.nl/)

E-mail del autor: [laura\\_ARROBA\\_linuv.uv.es](mailto:laura_ARROBA_linuv.uv.es)

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2179>