

## Coronel Mansilla

coronel@lugmen.org.ar

## Groucho Marx

groucho@lugmen.org.ar

## Juan E. Vitale

vita@lugmen.org.ar

## JuanJo Ciarlante

jjo-lug-wireless@mendoza.gov.ar

## Harpo MAxx

harpo ;at; lugmen.org.ar

---

### *Tabla de contenidos*

#### [Lug-Wireless Mini-HOWTO](#)

1. [Introducción](#)
2. [Instalación de hardware](#)
  - 2.1. [Inserción del dispositivo PCI](#)
  - 2.2. [¿Funciona la placa sin antena?](#)
  - 2.3. [Actualización del firmware](#)
3. [Instalación de software](#)
  - 3.1. [Requerimientos](#)
  - 3.2. [Compilación del kernel](#)
  - 3.3. [Subsistema PCMCIA](#)
  - 3.4. [Driver linux-wlan-ng](#)
  - 3.5. [Driver hostap](#)
4. [Configuración run-time](#)
  - 4.1. [PCMCIA](#)
  - 4.2. [Configuración básica de red](#)
  - 4.3. [Configuración de red 802.11](#)
5. [Diagrama ASCII de conexionado](#)
6. [Antena](#)
  - 6.1. [Elementos irradiantes \(antenas\)](#)
  - 6.2. [Alimentador](#)
  - 6.3. [Pigtails](#)
7. [Referencias](#)
8. [Conclusiones](#)
9. [FAQ](#)

### *Lista de tablas*

1. [Denominación drivers prism2](#)

2. [Placas Prism2 802.11 largo alcance](#)
3. [Placas Prism2 802.11 \(otras\)](#)
4. [Placas NO Prism2 802.11 \(NO RECOMENDADAS\)](#)

### *Tabla de figuras*

1. [Diagrama de conexiones](#)

# Lug-Wireless Mini-HOWTO

## Configuración


Versión 0.5 (\$Id: Config-HOWTO.sgml,v 1.18 2003/02/08 20:50:14 harpo Exp \$)

---

### *Tabla de contenidos*

1. [Introducción](#)
2. [Instalación de hardware](#)
  - 2.1. [Inserción del dispositivo PCI](#)
  - 2.2. [¿Funciona la placa sin antena?](#)
  - 2.3. [Actualización del firmware](#)
3. [Instalación de software](#)
  - 3.1. [Requerimientos](#)
  - 3.2. [Compilación del kernel](#)
    - 3.2.1. [Configuración del código fuente del kernel](#)
    - 3.2.2. [Compilación del kernel](#)
    - 3.2.3. [Instalación del kernel](#)
  - 3.3. [Subsistema PCMCIA](#)
    - 3.3.1. [Compilación de pcmcia-cs](#)
    - 3.3.2. [Configuración del código fuente de pcmcia-cs](#)
  - 3.4. [Driver linux-wlan-ng](#)
    - 3.4.1. [Compilación del código fuente de linux-wlan-ng](#)
  - 3.5. [Driver hostap](#)
    - 3.5.1. [Compilación del driver alternativo hostap](#)
4. [Configuración run-time](#)
  - 4.1. [PCMCIA](#)
    - 4.1.1. [Drivers PCMCIA](#)
    - 4.1.2. [Carga automática del soporte PCMCIA](#)
  - 4.2. [Configuración básica de red](#)
  - 4.3. [Configuración de red 802.11](#)
5. [Diagrama ASCII de conexionado](#)
6. [Antena](#)
  - 6.1. [Elementos irradiantes \(antenas\)](#)
  - 6.2. [Alimentador](#)
  - 6.3. [Pigtails](#)
7. [Referencias](#)
8. [Conclusiones](#)
9. [FAQ](#)

## 1. Introducción

 Si vas a contribuir con este documento, agregate en orden alfabético a la lista de autores, indicando brevemente tu contribución. También aumentá el número de versión, subversión o revisión así podemos seguirle el rastro a los cambios.

Aumentá el número de versión si tus cambios contradicen la versión previa, volviéndola "incompatible" con la nueva versión. Aumentá el número de subversión si tus cambios agregan, sacan, reorganizan o amplían secciones sin que ocurran "incompatibilidades" que requieran aumentar el número de versión.

Aumentá el número de revisión si hacés cambios cosméticos, de redacción, ortografía, typos, etc. En otras palabras: si el archivo sigue diciendo lo mismo, pero de una mejor manera, tenés que cambiar el número de revisión.

Este HOWTO lo escribimos para documentar nuestra experiencia en la instalación y puesta a punto de nuestras placas de red inalámbrica WL200 bajo GNU/Linux, de manera que sea más sencillo para los demás realizar las mismas tareas en el futuro. Sin embargo puede ser usado como documento para instalar y configurar cualquier placa basada en los chipset PrimsII y Prism 2.5. En cierto sentido, este HOWTO vendría a ser como un resumen pasado en limpio de la lista lug-wireless. En varios lugares hay citados fragmentos de mensajes sacados literalmente de la lista.

Nos basamos en el sintético pero completo paper escrito por C0r0nel, quien a su vez se apoyó en la gentil colaboración de Vita para hacer andar su placa. Agradezco también las ampliaciones y correcciones de Juanjo.

Esto no intenta ser un manual de usuario sino una guía de instalación rápida pero detallada para aquellos que quieren conectarse a la LUG-Wireless. Al igual que en las versiones anteriores de este trabajo, vamos a suponer familiaridad con el proceso de compilación del kernel. Para los que no satisfacen ese requisito, les recomiendo leer el Kernel-HOWTO. La explicación está ejemplificada con Debian 3.0 "Woody" y kernel 2.4.19 tratando de usar el máximo posible de herramientas de Debian.

---

## 2. Instalación de hardware

### 2.1. Inserción del dispositivo PCI

Antes de comenzar, necesitamos instalar en una ranura PCI libre la placa WL200. Hasta esta operación tan simple puede traer problemas. Por empezar, existe un problema de compatibilidad con la función Wake-on-LAN en algunos motherboards AOpen.

En las palabras de Marcelo Roccasalva: " El problema es sencillo: ponés la placa y la máquina ni siquiera hace el beep de arranque... Nada... Silencio... Muerta... Aparentemente es un problema conocido por los chicos de AOpen: la pata A14 recibe 3.3 V para el Wake-On-LAN, pero en ciertas condiciones una placa PCI 2.2 puede provocar que este voltaje caiga hasta unos 2 V y el motherboard se congela antes de intentar arrancar... Lo que recomiendan es "aislar" dicha pata para que no reciba corriente..."

Más adelante Juanjo Ciarlante nos comentó que luego de aislar la mencionada pata, la placa no funcionó (si bien el motherboard sí arrancó) y que al final Marcelo tuvo que instalar la placa en un motherboard viejo...

Luego de instalada la placa, si todo salió bien, debería aparecer en el listado de dispositivos PCI que produce el BIOS luego del POST, o más relevantemente, en el listado producido por el programa lspci (que lee la información directamente de /proc/bus/pci) e informa la lista de dispositivos PCI que el kernel logró identificar y asignarles recursos:

La salida de **lspci** debería ser similar a esta:

```
bash # lspci
00:00.0 Host bridge: Intel Corp. 430VX - 82437VX TVX [Triton VX] (rev 02)
00:07.0 ISA bridge: Intel Corp. 82371SB PIIX3 ISA [Natoma/Triton II] (rev 01)
00:07.1 IDE interface: Intel Corp. 82371SB PIIX3 IDE [Natoma/Triton II]
00:07.2 USB Controller: Intel Corp. 82371SB PIIX3 USB [Natoma/Triton II] (rev 01)
00:09.0 VGA compatible controller: Trident Microsystems TGUI 9660/968x/968x (rev d3)
00:0a.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8139 (rev 10)
00:0b.0 PCMCIA bridge: Cirrus Logic CL 6729 (rev 07)
```

Acá viene la primera sorpresa para mí: la placa WL200 es en realidad un puente PCI-PCMCIA, y la funcionalidad de NIC (network interface controller) está dentro de un dispositivo PCMCIA embebido.

---

## 2.2. ¿Funciona la placa sin antena?

El otro aspecto digno de comentario relacionado con la instalación de hardware es si es recomendable o no dejar instalada y activa la placa de red sin antena o carga fantasma. En principio, la desadaptación de impedancia entre un transmisor y el medio (en ausencia de antena) fuerza a que parte de la onda electromagnética se refleje de regreso al transmisor en la frontera entre el conector de antena y el aire. La onda reflejada se disipa como calor en el transmisor. Esto último puede ser perjudicial para el transmisor, pudiendo causar desajuste, funcionamiento deficiente o destrucción. Mediciones realizadas por el Tano indican que la placa WL200 está perfectamente adaptada cuando opera sin antena. Esto es razonable en un dispositivo con antena removible para uso de público en general (léase idiotas), por lo tanto no hay peligro del cual preocuparse.

---

## 2.3. Actualización del firmware

Compaq ofrece un upgrade de firmware a la versión 0.8.0, obtenible con:

```
bash $ wget ftp://ftp.compaq.com/pub/softpaq/sp14501-15000/sp14796.exe
```

El upgrade del firmware es obligatorio para hacer andar correctamente el driver de la WL200 bajo Linux (confirmado con hostap, casi seguro para wlan-ng). Desafortunadamente, Compaq ofrece la actualización de firmware en la forma de "softpaq" (nombre ñoño de Compaq para los paquetes de software). Estos softpaqs son autoejecutables para DOS solamente, por lo que al menos durante unos minutos tendremos que tener a mano una box con DOS/Windows.

Compaq ofrece allí dos paquetes:

```
sp14796.exe
```

crea un diskette de arranque de 1,44 MB conteniendo las utilidades de upgrade en forma *standalone*  
*Más adecuado para servers sólo Linux* y no necesitamos instalar temporalmente la WL200 en una máquina con DOS/Windows.

## 2.2. ¿Funciona la placa sin antena?

*sp15584.exe*

actualización de firmware en la forma de "softpaq", es decir: paquete autoejecutable para Windows solamente. *Más adecuado para PCs Linux/Win (drivers p/Win)*

Existe un firmware 0.8.3 para placas \*SMC\* con el mismo chipset (en el archivo se llama *s10008c3.hex*) descargable desde

[http://www.smc-europe.com/english/support/driver\\_manual/wirel/download/2632/WinUpdate\\_OEM\\_1.1.zip](http://www.smc-europe.com/english/support/driver_manual/wirel/download/2632/WinUpdate_OEM_1.1.zip) y también en: (no se cuál es la diferencia) <http://www.goonda.org/wireless/prism2/>

Varios han tenido éxito flasheando el firmware 0.8.3, como así también otras versiones *1.3.5 1.4.9* y *1.5.6* sin embargo NO han presentado mejoras notables y hasta en algunos casos no fue posible asociarse usando alguno de los firmwares mencionados. Un listado de todos estos drivers puede encontrarse en : <http://www.mendoza-wireless.net.ar/~jjo/wl200/>

El procedimiento que ha resultado más cómodo para cargar estas versiones de firmware así sido utilizando el disquete provisto por COMPAQ que creamos al ejecutar SP14796.EXE copiando el .hex conteniendo la versión de firmware que deseamos y luego editando el archivo *pci.bat*.

Tener en cuenta que en este archivo encontraremos referencias a 3 archivos .hex distintos. La línea que debemos modificar es la que hace referencia al segundo firmware.

```
delay 200
flash -5v -d p10003c0.hex
delay 200
flash -5v -d s10008c0.hex      <----- Este es el que hay que tocar
delay 200
flash -5v -d oneirq.hex
```

Entonces reemplazamos *s10008c0.hex* por la versión de firmware que necesitamos cargar en la placa y luego utilizamos el disquete de manera natural.

Por otro lado, el driver *hostap* permite cargar un nuevo firmware en la RAM de la placa, por lo tanto esto permitiría experimentar en forma SEGURA con la versión 0.8.3 (sin correr el riesgo de tostar la placa)

La utilidad es la *prism2\_srec* la cual permite cargar un firmware haciendo uso de la RAM o también permite la posibilidad de 'tostarlo' en la memoria FLASH (pero esto aún no ha sido probado) y es recomendable seguir usando el método antes mencionado.

La utilidad *prism\_srec* que viene en la versión de *hostap 2002-10-12* no permite forzar la carga en RAM por lo que es necesario usar la utilidad *prism2\_srec* que está en el CVS. Si no resulta posible cargar ningún firmware, ya que *prism2\_srec* se queja por incompatibilidades del firmware.

Por lo que Harpo ha podido probar, no todos los firmwares vienen preparados para poder ser cargados en la memoria RAM, de hecho el único con el que tuvo éxito fue el *1.5.6*



Debemos tener en cuenta todos los cuidados usuales al sobrescribir el firmware de un dispositivo de hardware: no interrumpir el proceso, evitar los cortes de energía, etc. No estoy seguro qué pasa si se graba mal el firmware, pero por las dudas, mejor no averiguarlo. ;-)

Aquí hay una referencia muy interesante (Nov-2002) en español respecto de la actualización del firmware a versiones 1.5.x+, las cuales *permiten interoperar correctamente con otros APs para crear un WDS*:

<http://manty.net/> ... yeah!

Parece estar disponible directamente desde Intersil un upgrade al firmware 1.5.x para las placas Prism2 en [http://www.intersil.com/design/prism/ss/crack-n-peel/prod-drvr/drv209/ISL3PRDSUIT-EXE\\_WIN\\_\(2-0-9\).zip](http://www.intersil.com/design/prism/ss/crack-n-peel/prod-drvr/drv209/ISL3PRDSUIT-EXE_WIN_(2-0-9).zip), adicionalmente podemos ver un howto-cito en <http://lists.shmoo.com/pipermail/hostap/2002-November/000295.html>. A propósito de ésto, los *peers* de [Sydney Wireless](#) ya han hecho la conversión mencionada en la mailing list, para la Compaq WL200 el archivo parece ser el [r1010506.hex](#).

---

## 3. Instalación de software

### 3.1. Requerimientos

Desafortunadamente, los drivers de la WL200 no están incorporados en el kernel, ni se consiguen versiones precompiladas y empaquetadas de los drivers, de manera que habrá que compilarlos por uno mismo. Peor aún, el soporte de PCMCIA (necesario para hacer andar el puente PCI-PCMCIA) incorporado en el kernel no es lo suficientemente reciente como para soportar adecuadamente la WL200.

Por lo tanto, tendremos que usar los drivers del kernel provistos por el proyecto pcmcia-cs, y no las versiones de los mismos distribuidas con Linux.

Los kernels que Debian distribuye traen activado el soporte para PCMCIA, el que como acabamos de ver, no sirve. Por lo tanto, habrá que compilar un kernel personalizado sin soporte PCMCIA.

Otras herramientas distribuidas por Debian son adecuadas, tales como: pcmcia-cs (los drivers del kernel vienen aparte en kernel-pcmcia-modules) y wireless-tools.

```
bash # apt-get install pcmcia-cs wireless-tools
```

Como vamos a compilar el kernel, instalamos el código fuente y la herramienta kernel-package para generar un bonito paquete .deb personalizado.

```
bash # apt-get install kernel-source kernel-package
```

Necesitamos también el código fuente de pcmcia-cs (para usar los drivers PCMCIA del kernel incluidos allí) y el driver específico para la WL200:

```
bash # apt-get source pcmcia-cs
bash # wget ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/linux-wlan-ng-0.1.15.tar.gz
```

De esta forma, el código fuente que obtenemos coincide con el que se usó para generar los paquetes que instalamos en el paso previo, y obtenemos el fuente debianizado para poder generar un paquete .deb personalizado de pcmcia-cs. Si no, se puede obtener los distintos trozos de código fuente en:

- Linux 2.4.19 en <http://www.kernel.org>
  - pcmcia-cs 3.2.1 en <http://pcmcia-cs.sourceforge.net>
  - linux-wlan-ng 0.1.15 en <http://www.linux-wlan.org>
  - hostap-2002-10-12 en <http://hostap.epitest.fi> (driver alternativo)
-

## 3.2. Compilación del kernel

### 3.2.1. Configuración del código fuente del kernel

Para compilar el driver es necesario tener el código fuente del kernel ya configurado, recordando tener desactivado CONFIG\_PCMCIA y activado CONFIG\_NET\_RADIO esto ultimo necesario para activar las extensiones wireless a travez de entradas en el directorio /proc/wireless. Hay que tener cuidado de no reactivar el soporte PCMCIA del kernel accidentalmente al activar alguna otra opción de configuración que inadvertidamente requiera PCMCIA.

```
bash # cd /usr/src/kernel-source-2.4.19
bash # make menuconfig
```

---

### 3.2.2. Compilación del kernel

Todos sabemos compilar el kernel, pero para que quede completa la explicación recordemos que si usamos kernel-package lo hacemos con algo como:

```
bash # vi /etc/kernel-package.conf (ponemos nuestro nombre y otras trivialidades)
bash # make-kpkg clean
bash # make-kpkg --revision=custom.1.0 kernel_image
```

De esta forma, generamos un paquete .deb con nuestro kernel personalizado sin soporte para PCMCIA.

---

### 3.2.3. Instalación del kernel

Simplemente:

```
bash # dpkg -i kernel-image-2.4.19-custom.1.0-i386.deb
```

Creamos o actualizamos la imagen initrd si hiciera falta, corremos LILO, rebooteamos, y listo: tenemos un kernel sin soporte PCMCIA.

---

## 3.3. Subsistema PCMCIA

### 3.3.1. Compilación de pcmcia-cs

Uno de los programas incluidos en pcmcia-cs es un cliente X11, por lo tanto, para compilar el paquete completo necesitamos las bibliotecas y cabeceras de xlib. Para ello será suficiente:

```
bash # apt-get install xlibs-dev
```

---

### 3.3.2. Configuración del código fuente de pcmcia-cs

También debemos configurar el código fuente de pcmcia-cs. Cuando bajamos código fuente usando apt-get, obtenemos el archivo .tar.gz prístino original del autor upstream, más un parche con las diferencias aplicadas por el mantenedor de Debian.

Por lo tanto, tenemos la opción de compilar los fuentes genéricos, o construir un paquete .deb configurado a nuestro gusto.

```
bash # cd /usr/src/pcmcia-cs-3.2.1
```

```
bash # export KSRC=/usr/src/kernel-source-2.4.19
bash # debian/rules binary
bash # debian/rules binary-modules
```

Acá usamos el método de Debian, consistente en correr el programa `debian/rules` para compilar. `debian/rules` es un `makefile` que envuelve el sistema de compilación original del autor upstream, y agrega unos targets adicionales para generar el paquete `.deb`. En nuestro caso, usamos los targets ``binary'` y ``binary-modules'` para crear 2 paquetes `.deb` `pcmcia-cs` y `pcmcia-modules` idénticos a los suministrador por Debian excepto que usan los módulos que vienen con `pcmcia-cs` y no los que vienen con Linux. (Esto último es autodetectado al compilar, fijándose en el kernel en ejecución o en los fuentes a los que señala `KSRC`).

Además, permite la configuración no interactiva del código fuente previo a la compilación.

Si no instalamos los binarios de Debian, habrá que compilar el paquete e instalarlo:

```
bash # cd /usr/src/pcmcia-cs-3.2.1
bash # make config
```

y seleccionaríamos todas las opciones por defecto. Ojo: en la configuración pide el path a los fuentes del kernel; pónganle el correcto (`/usr/src/kernel-source-2.4.19` por ejemplo). Luego:

```
bash # make install
```

Entre otras cosas, esto instala binarios, archivos de configuración, documentación, drivers del kernel y scripts de inicio estilo Debian.

---

## 3.4. Driver `linux-wlan-ng`

### 3.4.1. Compilación del código fuente de `linux-wlan-ng`

Vamos a compilar este paquete con todas las opciones por defecto. El único dato que hace falta cambiar es el path al código fuente del kernel (`/usr/src/kernel-source-2.4.19` por ejemplo).

```
bash # cd /usr/src/linux-wlan-ng-0.1.15
bash # make config
bash # make all
bash # make install
```

No debería haber problemas.

---

## 3.5. Driver `hostap`

### 3.5.1. Compilación del driver alternativo `hostap`

Existe un driver alternativo a `wlan-ng`, llamado `hostap`. Las ventajas de este último son:

- Soporta modo AP :-)
- El código es más limpio y elegante (Juanjo's)
- Trae un excelente soporte de las Wireless Extensions, lo que permite manejar mediante el comando `standard iwconfig` la totalidad de las funcionalidades de la placa
- Funciona bien con el soporte PCMCIA de los fuentes del kernel (Coronel's), aunque para algunas

placas (como la Compaq WL200 esto no es aprovechable ya que no andan con el módulo i82365.o provisto por el kernel).

Un artículo muy interesante respecto del uso y las capacidades del hostap lo podés encontrar en [http://trekweb.com/~jasonb/articles/hostap\\_20021012.shtml](http://trekweb.com/~jasonb/articles/hostap_20021012.shtml), vale la pena leerlo.

Antes de proseguir, es importante aclarar el problema de denominación que surge entre ambos drivers:

**Tabla 1. Denominación drivers prism2**

Paquete	Módulo de kernel	Nombre coloquial
linux-wlan-ng-0.1.15	prism2_cs.o	"wlan-ng"
hostap-AAAA-MM-DD	hostap_cs.o	"hostap"

Es decir, si decimos "prism2" no podemos diferenciar a cual driver nos referimos (de hecho ESE es el nombre del chipset), por lo tanto vamos a hablar de driver "wlan" o "hostap". El hostap lo pueden obtener de <http://hostap.epitest.fi/> Según Juanjo:

```
" Se puede ver como la gente de linux-wlan rompió el K.I.S.S.; de
ahí que el hostap me parezca MUUCHO más prolijo.
Apropos... les cuento más o menos (no soy muy ordenado con la
memoria) lo que hice para compilar el HostAP :"
```

Es mejor (si no cuasi-necesario) tener el pcmcia-cs andando de acuerdo a las instrucciones anteriores. Luego de descargar el paquete hostap-AAAA-MM-DD.tar.gz hacemos:

```
bash # tar zxvf /paso/de/descarga/hostap-AAAA-MM-DD.tar.gz
bash # cd hostap-AAAA-MM-DD
bash # vi Makefile (editar Makefile)
      KERNEL_PATH=/usr/src/kernel-source-2.4.19
      PCMCIA_PATH=/usr/src/pcmcia-cs-3.2.1
bash # make pccard
bash # make install_pccard
```

---

## 4. Configuración run-time

### 4.1. PCMCIA

#### 4.1.1. Drivers PCMCIA

##### 4.1.1.1. Driver wlan-ng

Editamos el archivo /etc/pcmcia/wlan-ng.conf y agregamos esta línea al final:

```
module "prism2_cs" opts "prism2_ignorevcc=1"
```

Si no hacemos esto, el driver no carga. Quizás no lo necesiten, pero a nosotros no nos anduvo sin esta opción.

---

#### 4.1.1.2. Driver hostap

En el `/etc/pcmcia/hostap_cs.conf` debemos agregar:

```
module "hostap_cs" opts "iw_mode=2 essid=lugmen ignore_cis_vcc=1"
```

Donde:

```
iw_mode (/usr/src/linux/include/linux/wireless.h):
#define IW_MODE_AUTO      0      /* Let the driver decides */
#define IW_MODE_ADHOC     1      /* Single cell network */
#define IW_MODE_INFRA     2      /* Multi cell network, roaming, ... */
#define IW_MODE_MASTER    3      /* Sync. master or Access Point */
#define IW_MODE_REPEAT    4      /* Wireless Repeater (forwarder) */
#define IW_MODE_SECOND    5      /* Secondary master/repeater (backup) */
ssid
El SSID "lugmen" para el hostap DEBE ser puesto como parámetro en
la carga del módulo si el driver no es reciente, las últimas versiones
soportan el uso de
    iwconfig wlan0 essid lugmen
(XXX:jjo aparentemente sería posible entonces editar
/etc/pcmcia/wireless.opts y poner allí 'ESSID="lugmen"')
```

El Coronel tuvo un problema para hacer andar el driver hostap, relacionado más bien con el driver para el puente PCI-PCMCIA:

" Hay que levantar el módulo `i82365` con la opción `irq_mode=1`. Yo lo hice con el pcmcia detenido y haciendo un "

```
modprobe i82365 irq_mode=1
```

luego levanté el pcmcia y me dijo que los módulos ya estaban arriba; cargó el `hostap_cs` y salió andando como un toro."

En principio, la opción necesaria se puede colocar en `/etc/modules.conf` agregando una línea así:

```
options i82365 irq_mode=1
```

En el caso de Debian, el archivo `modules.conf` es generado por el script `update-modules` concatenando los fragmentos presentes en `/etc/modutils`. Por lo tanto, para que la distribución no sobrescriba nuestros cambios, tendríamos que crear un archivo en `/etc/modutils` llamado como queramos (`i82365` es un buen nombre), agregar allí la línea mencionada y correr:

```
bash # update-modules
```

para regenerar `modules.conf`.

Todo esto suponiendo que el script de inicio `/etc/init.d/pcmcia` use **modprobe** y no **insmod**, como parece que hace.

---

#### 4.1.1.3. Coexistencia de wlan y hostap

Es posible tener una configuración "genérica" de PCMCIA y con links simbólicos poder probar uno u otro driver. El PCMCIA asocia la tarjeta con el driver en base a `/etc/pcmcia/*.conf`, leyéndolos en orden alfabético; por lo tanto si están

```
:
/etc/pcmcia/hostap_cs.conf
:
/etc/pcmcia/wlan-ng.conf      <-- último encontrado => USADO
:
```

Por lo tanto, es suficiente para

*usar hostap:*

```
mv /etc/pcmcia/wlan-ng.conf /etc/pcmcia/wlan-ng.conf.NO
```

*usar wlan (nuevamente)*

```
mv /etc/pcmcia/wlan-ng.conf.NO /etc/pcmcia/wlan-ng.conf
```

Obviamente, es necesario hacer un "pcmcia restart" entre cambios.

 Las líneas de configuración de módulos PCMCIA deben ir en el archivo .conf respectivo y no en el config.opts general, por ej:

```
wlan-ng.conf
module "prism2_cs" opts "prism2_ignorevcc=1"

hostap_cs.conf
module "hostap_cs" opts "iw_mode=2 essid=lugmen ignore_cis_vcc=1"
```

---

#### 4.1.1.4. PCMCIA irq\_mode

Para que sea ejecutado al inicio la carga de los módulos pcmcia con el parámetro irq\_mode se puede simplemente editar el archivo /etc/defaults/PCMCIA y agregar:

```
PCIC_OPTS="irq_mode=1"
```

que por lo menos en Debian ha funcionado perfecto.

---

#### 4.1.2. Carga automática del soporte PCMCIA

Para que el soporte PCMCIA sea cargado automáticamente, editamos los links simbólicos estilo SystemV (HELO Nw2o!) a gusto. En Debian podemos usar:

```
bash # update-rc.d pcmcia defaults
```

Finalmente ejecutamos:

```
bash # /etc/init.d/pcmcia start
(la máquina va a hacer unos beeps y, en la terminal tiene que aparecer algo así:)
Starting PCMCIA services: modules cardmgr.
cardmgr[1985]: watching 2 sockets
```

A mí me pasó que tenía instalado el paquete hotplug y cuando cargaba los módulos PCMCIA, la máquina se colgaba mal: ni siquiera respondía a Alt-Ctrl-Fn para cambiar de consola virtual. Al desinstalar ese paquete (que había instalado por curioso no más), todo comenzó a andar bien. Al hacer un lsmod podemos ver lo siguiente (wlan):

```

bash # lsmod
...
prism2_cs          94320  1
p80211             18456  1 [prism2_cs]
ds                 6720   2 [prism2_cs]
i82365             22416  2
pcmcia_core        41472  0 [prism2_cs ds i82365]
...

```

Si es así, vamos bien.

Le ponemos una dirección IP (para probar)

```
bash # ifconfig wlan0 alguna-ip
```

Ahora tenemos la placa arriba y con una dirección IP asignada. Para ver que sabe el kernel de nuestra placa:

```

bash # cat /proc/net/wireless
Inter-| sta-|   Quality           |   Discarded packets           | Missed
face  | tus  | link level noise | nwid  crypt  frag  retry  misc | beacon
wlan0: 0000   0.  27.  27.      0    0    0    0    0

```

Otros comanditos utiles son:

```

bash # iwconfig wlan0
wlan0      IEEE 802.11-b  ESSID:"non-spec"
          Mode:Managed  Frequency:2.422GHz  Access Point: 44:44:44:44:44:44
          Link Quality:0/100  Signal level:27/100  Noise level:27/100
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0  Missed beacon:0

```

... y el viejo y querido

```

bash # ifconfig wlan0
wlan0      Link encap:Ethernet  HWaddr 00:50:8B:D0:34:92
          inet addr:10.1.1.1  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING 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:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:3 Base address:0x100

```

Si van obteniendo resultados similares, ya estamos con la placa andando. si no, a revisar... :-)

## 4.2. Configuración básica de red

La configuración de una interfaz inalámbrica requiere más datos que los requeridos por una interfaz cableada.

Primero configuramos la parte común a todas las interfaces, editando el archivo `/etc/network/interfaces` y agregando algo como:

```
iface wlan0 inet static
```





Alimentador: Lo/4=30,966mm  
Largo mínimo del tarro: 3/4 Lg=191,723mm"

Es importante que la guía de ondas no tenga la rebaba que queda en las latas de conserva al abrirlas.

Esta es la lista de recursos de Internet donde se documentan antenas que realmente algunos de nosotros construimos y probamos:

- \* <http://debaser.ath.cx/atroz/docs/antennas/cantenna/>
- \* <http://grocities.com/populind/>
- \* <http://www.frars.org.uk/cgi-bin/render.pl?pageid=1066>
- \* <http://members.gbonline.com/~multiplx/wireless/parabolic.main.cgi>
- \* <http://www.saunalahti.fi/elepal/antennie.html>
- \* <http://www.wireless.org.au/~jhecker/helix/helical.html>
- \* <http://www.pymes.com/radiolink.htm>
- \* [http://www.qsl.net/n9zia/wireless/inside\\_lance.html](http://www.qsl.net/n9zia/wireless/inside_lance.html)
- \* <http://www.qsl.net/ki7cx/wgfeed.htm>
- \* <http://www.frars.org.uk/cgi-bin/render.pl?pageid=1058>
- \* <http://trevormarshall.com/waveguides.htm>
- \* <http://www.saunalahti.fi/elepal/antenna4.html>

Las antenas tipo guía de ondas abierta ("tarrito") son bastante fáciles de construir, tienen una ganancia y directividad decentes (10 dBi) y con un poco de suerte, un ROE aceptable en al menos 1 o 2 canales (1,3 a 1,6). Las antenas con reflector parabólico han funcionado muy bien, en especial las comerciales (no caseras) con ganancias de 20 dBi, y ROE 1,3 en todos los canales.

Las antenas helicoidales no sirven porque irradian con polarización elíptica y las antenas omnidireccionales del lado del AP irradian con polarización vertical. Las otras antenas no han dado buenos resultados (todavía).

En el lado del AP estamos usando por ahora una antena "colineal" vertical de 8 o 21 elementos, de lóbulo de irradiación omnidireccional y entre 6 dBi a 10 dBi. Se trata de antenas frágiles que requieren un soporte, pues de lo contrario se quiebran de sólo manipularlas.

El soporte debe ser transparente a las microondas, con lo cual queda excuido cualquier compuesto metálico. La elección obvia es entonces un tubo de plástico, como los utilizados en canalizaciones domiciliarias de agua. Algunos de estos caños (los de color naranja/marrón) contienen un compuesto ferroso que los vuelven opacos a las microondas. Lo ideal es probar el comportamiento del soporte colocando un recipiente con agua dentro del tubo bajo prueba y tratar de calentarlo con el horno microondas.

---

## 6.2. Alimentador

El tema acá es lograr la mínima atenuación posible, lo que se logra utilizando cable de baja atenuación y alta frecuencia de corte, como el LMR-400. Desafortunadamente, se trata de cable muy caro y difícil de conseguir. Un cable con peores características (pero no tanto) que es más accesible y altamente disponible es el RG-213 (hasta 12m debería andar bien). Según Pollux:

" Por cada 100 pies el RG-213 atenúa unos 9dB a 1GHz, tenés el RG-8 (FOAM) que atenúa unos 5.7dB a la misma frecuencia. Mejor que ese está el RG-18 que atenúa unos 4.3dB. Después de esos cables ya vienen los "hardlines" que obviamente son muchísimo más caros. El compuesto de FOAM es parecido a un telgopor muy condensado, de color blanco y un poco poroso."

Según Marcos:

" Averigüé el precio del cable RG-8: si compramos por cantidad sale \$2,40 el metro, el rollo de 100m"

Estos son los recursos de Internet donde pueden encontrar información pertinente sobre cables.

- \* [http://www.wirelesscorp.net/cable\\_specs.htm](http://www.wirelesscorp.net/cable_specs.htm)
- \* <http://www.coronel.org.ar/DL/22.pdf>
- \* <http://debaser.ath.cx/~atroz/docs/antenas/cantenna/>

El cable lleva conectores tipo N macho en los extremos, de los que tienen que comprar la variedad para crimpear (queda más prolijo, pero requiere una pinza especial).

El Coronel ha tenido buenos resultados con los conectores marca Amphenol, específicamente:

```
Amphenol RFX
82-202-RFX
Type N CLAMP-SOLDER PLUG FOR RG-8, 213, 214
```

---

### 6.3. Pigtaills

Existen pigtaills comerciales de cerca de U\$S 30 (damn!) por lo que vamos a construirlos caseramente.

El pigtail consta de un cable que debe ser lo más corto posible para reducir al mínimo la atenuación que la señal experimenta al atravesarlo, pero no tan corto que sea inmanejable el rígido cable del alimentador de antena. Entre 20 y 50 cm estaría bien.

El tipo de cable que estamos usando es el coaxil RG-58C, muy similar al RG-58 utilizado en el cableado de redes Ethernet pero con el conductor central monofilar en lugar de multifilar. En las casas de electrónica tienen que pedirlo como "coaxil para celular".

Otra dificultad con los pigtaills es que el conector que va del lado de la placa (conector SMA macho) debe tener la rosca levógira (al revés que las roscas normales) y tales conectores son "propietarios" y no se consiguen fácilmente sueltos. No queda más remedio que limarles ("birlarles") la rosca con una lima cutter hasta que enrosque, o llevarlos a un tornero (tiene que ser uno muy bueno) que les haga una rosca levógira superpuesta a la dextrógira original.

No es nada recomendable limar la rosca del conector SMA soldado a la placa de red. Se trata de un conector cubierto de oro (ventajas: mejor conductor eléctrico que el cobre, e inoxidable) y si le limamos la rosca, desaparece la película dorada y el conector se arruina.

El siguiente diálogo entre Tano y el Coronel es ilustrativo:

```
_¿Vos sabés que me emperné con unos adaptadores SMA/TNC rosca
derecha en Electrónica Mendoza? ¿Me podrías comentar en qué consiste
el "birlado"?
_Simplemente eso, le hice mierda la rosca derecha con un cutter...
_Ahora entra en la placa y engancha con las últimas espiras forzándolo
un poco... :-)
```

Con respecto a los conectores TNC, no sirven porque son para frecuencias más bajas, creo que hasta 1 GHz.

El conector del pigtail que se conecta al alimentador de antena es del tipo N hembra para cable RG-58 y para crimpear (si tienen la pinza necesaria).

Estos son los recursos de Internet relacionados con la construcción de pigtails:

- <http://www.wireless.org.au/~jhecker/pigtail/pigtail.html>
- [http://www.epcom.net/epcom\\_espanol/conectores\\_RF.htm#sma](http://www.epcom.net/epcom_espanol/conectores_RF.htm#sma)
- [http://catalogo.syscom.com.mx/conectores\\_RF.htm#sma](http://catalogo.syscom.com.mx/conectores_RF.htm#sma)
- <http://www.superinventos.com/S160160.htm>
- <http://www.dayton.com.ar/fichas-sma.htm>
- <http://www.tecnindustria.com.ar/surplus.htm>
- <http://www.itenorio.com/principal.html>

---

## 7. Referencias

Vita ha conseguido estos libros interesantes para la puesta a punto de una red inalámbrica:

- Hack Proofing Your Wireless Network en <http://www.syngress.com/download/course/hpwireless.exe>
- Designing a Wireless Network en <http://www.syngress.com/download/course/WirelessNetwork.exe>
- Webmaster's Guide to the Wireless Internet en <http://www.syngress.com/download/course/wirelessinternet.exe>

---

## 8. Conclusiones

Bueno, como verán no es complicado, pero sí bastante hinchado, sobre todo con el tema de que la placa está detrás de un puerto PCMCIA, y dependemos de ese maldito y pedorro estándar.

Lo importante es ser muy prolijo y probar hasta que todo ande, porque TIENE que andar. Have Fun.

---

## 9. FAQ

- 9.1.1. [¿ Qué placa compro ?](#)
- 9.1.2. [Compré la placa, cargué los drivers, pero cuando quiero usarla \(ping\), me dá errores ...](#)
- 9.1.3. [De acuerdo a los timings de 802.11, cuál es la máxima distancia alcanzable](#)

### 9.1.1. ¿ Qué placa compro ?

Estas son algunas consideraciones importantes a la hora de comprar la placa:

- Chipset Prims2: *fundamental* por el soporte en Linux y la capacidad de crear APs
- Potencia de transmisión mayor a 20dbm y sensibilidad menor a -90dbm: debido a que vamos a cubrir distancias considerables
- Posibilidad de conexión de *antena externa*

Para más detalles ver <http://www.seattlewireless.net/index.cgi/HardwareComparison>

Algunos ejemplos de placas

**Tabla 2. Placas Prism2 802.11 largo alcance**

Marca – Modelo	Potencia TX	Sensib. RX	Conector Antena	Comentarios
Compaq – WL200	20 dbm (100mW)	aprox. –95 dbm @ 1mbps	SMA inverso	muy buen <a href="#">knowHOW</a>
Netgate – NL2511 (PCMCIA)	23 dbm (200mW)	–95 dbm @ 1mbps	desconocido	Ver <a href="http://www.netgate.com/ordershort.html">http://www.netgate.com/ordershort.html</a>

**Tabla 3. Placas Prism2 802.11 (otras)**

Marca – Modelo	Potencia TX	Sensib. RX	Conector Antena	Comentarios
Netgear – MA311 (PCI)	??	??	SMA reverso	
DLINK – DWL520 (PCI)	16 dbm (40mW)	–84 dbm	SMA inverso	Poca potencia y sensibilidad. Se consigue en MZA
CNET – CNWLC–811 (PCMCIA)	13 dbm (20mW)	–80 dbm	parece que NO	MUY poca potencia y sensibilidad. Se consigue en BSAS

**Tabla 4. Placas NO Prism2 802.11 (NO RECOMENDADAS)**

Marca – Modelo	Potencia TX	Sensib. RX	Conector Antena	Comentarios
Samsung – SWL 2100P (PCI)	17dbm	–86 dbm @ 11mbps	SMA inverso	Se consigue en BSAS. Driver para linux provisto por el fabricante. Ver <a href="http://www.connectronics.com/samsung/SWL_2100NEP">http://www.connectronics.com/samsung/SWL_2100NEP</a>
D–Link 520+ Wireless PCI 22Mbps (PCI)				No tiene soporte en Linux
D–Link 650+ Wireless PCMCIA 22Mbps (PCMCIA)				No tiene soporte en Linux

**9.1.2.** Compré la placa, cargué los drivers, pero cuando quiero usarla (ping), me dá errores ...

Verificar

- Confirmá haber hecho el upgrade de firmware como indica [Sección 2.3](#)

Problemas con IRQ: asegurate que el módulo PCMCIA se está cargando con `irqmode=1` de acuerdo a [Sección 4.1.1.4](#)

**9.1.3.** De acuerdo a los *timings* de 802.11, cuál es la máxima distancia alcanzable

<http://www.personaltelco.net/index.cgi/MaximumDistance>