



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

## Configuración de tarjetas wireless PCMCIA con drivers de Hasefroch XP en Linux (973 lectures)

Per **Dani Garcia**, [yunwebo](http://www.danigarcia.com) (<http://www.danigarcia.com>)

Creado el 23/04/2004 12:14 modificado el 23/04/2004 12:14

*En este artículo explicaremos cómo conseguir configurar en Linux una tarjeta wireless PCMCIA utilizando sus correspondientes drivers para Hasefroch XP. ¿Cómo lo haremos? Pues con el wrapper del proyecto [ndiswrapper](#)<sup>(1)</sup> (Sourceforge). Para este mini-howto usaremos una Debian SID con kernel 2.6.5 sobre plataforma x86.*

**NOTA:** Utilizar este método **sólo** cuando no hayan drivers nativos Linux.

A continuación mostramos los pasos a seguir. En el ejemplo configuraremos una tarjeta wireless PCMCIA USRobotics 100Mbs(sólo en Hasefroch) (USR5410) con chipset Texas Instruments. Ésta es la salida del lspci: **0000:02:00.0 Network controller: Texas Instruments ACX 111 54Mbps Wireless Interface**

1. Primero miramos si nuestro chipset está soportado [aquí](#)<sup>(2)</sup>. En mi caso es éste: **TI ACX111**. Aquí mismo vemos también de donde bajar los drivers si no los tenemos.
2. Una vez comprobada la compatibilidad bajamos las "userspace utilities" y las fuentes del módulo de la web del proyecto. Para los que usen debian, aquí hay un repositorio:

```
deb http://rigtorp.se/debian/ unstable/
deb-src http://rigtorp.se/debian/ unstable/
```

Sólo falta hacer `apt-get install ndiswrapper ndiswrapper-source` y ya tenemos los paquetes.

3. Accedemos a nuestro directorio `/usr/src` y hacemos `tar xzf ndiswrapper-source.tar.gz` que nos descomprimirá las fuentes del módulo `ndiswrapper`.

4. Para continuar hemos de tener las fuentes de nuestro núcleo actual en `/usr/src/linux` y el paquete `kernel-package` instalado. Si todo está bien entramos al directorio y escribimos como root: `make-kpkg modules-image` que generará la imagen del módulo en el directorio `/usr/src`

5. A continuación hacemos un `dpkg -i ndiswrapper-modules*.deb` y ya tendremos el módulo instalado. Probamos a cargarlo con `modprobe ndiswrapper`. A mi no me funcionó y como no tenía ganas de recompilar el kernel escribí esto al final del archivo `/lib/modules/2.6.x/modules.dep` (Sustituid x por vuestra versión del kernel): `"/lib/modules/2.6.x/misc/ndiswrapper.ko:"` ahora ya podemos cargarlo. Una vez comprobado que el módulo carga, lo descargamos con `rmmmod ndiswrapper`

6. Seguidamente copiamos el archivo `.inf` del driver y sus dependencias (están especificadas dentro de éste) en un directorio cualquiera, y desde ahí hacemos un `ndiswrapper -i nombredelinf.inf` lo cual nos



instalará el driver en el sistema. Para comprobarlo podemos hacer un `ndiswrapper -l`. En mi caso devuelve:

```
Installed ndis drivers:
usr11g    present
```

Los archivos que tuve que copiar yo fueron: `usr11g.inf` `usr11g.sys` y `tiacx111.bin`

7. Para finalizar sólo tenemos que cargar el módulo y mirar los mensajes que éste nos da. Hacemos: `modprobe ndiswrapper`. Y si carga sin problemas hacemos un `dmesg` y miramos las últimas líneas. A mi me salió esto al final:

```
ndiswrapper version 0.6 loaded
ndiswrapper adding usr11g.sys
ndiswrapper: Buggy ndis driver trying to use unutilized spinlock. Trying
to recover...ok.
wlan0: ndiswrapper ethernet device 00:c0:49:d6:ce:ba using driver
usr11g.sys
```

Con esto ya tenemos nuestra tarjeta instalada como `wlan0`. ¡Ahora cada uno que se la configure a su gusto!

---

#### Lista de enlaces de este artículo:

1. <http://ndiswrapper.sf.net>
2. [http://ndiswrapper.sourceforge.net/supported\\_chipsets.html](http://ndiswrapper.sourceforge.net/supported_chipsets.html)

---

E-mail del autor: [jo\\_ARROBA\\_danigarcia.com](mailto:jo_ARROBA_danigarcia.com)

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2019>