



Publié sur **COAGUL** (<https://www.coagul.org/drupal>)

[Accueil](#) > [Rubriques](#) > [Publications](#) > [Matériel](#) > PDF imprimable

Comment faire fonctionner la carte wifi DWL-G650+ avec Kubuntu Feisty

ven, 25/05/2007 - 12:58 — jibux

[Matériel](#) ^[1]

But de ce Document

Ce document me sert de mémo. Il relate mon expérience lors de la mise à jour de Kubuntu Edgy vers Feisty. Je l'ai écrit pour me souvenir des différentes opérations que j'ai dû effectuer. Je le diffuse car cela pourra servir à d'autres personnes.

Situation initiale

Une machine sous Kubuntu 6.10. En fait celle-ci a été initialement installée avec la 6.06 puis a été mise à jour en 6.10 et donc maintenant en 7.04.

Pour plus d'informations [cet article](#) ^[2] relate l'installation, [et celui-ci](#) ^[3] la première mise à jour. Enfin la mise à jour vers la 7.04 a été faite en suivant les indications de [ce mémo](#) ^[4]. Et pour cette machine tout s'est bien passé hormis le fait que la carte PCMCIA DWL-G650+ ne fonctionne plus.

Identification et résolution du problème

Le gros avantage avec linux c'est que le système consigne énormément d'informations dans des fichiers log. La consultation du fichier `/var/log/messages` va nous donner les indications qui permettront de solutionner ce problème. Pour cela la commande `tail -f` est fort pratique, car elle permet de voir en temps réel les informations s'ajouter dans un fichier. On tape donc cette commande, puis plusieurs fois sur la touche "Entrée" afin d'ajouter quelques lignes vides sur l'écran, puis j'insère la carte WIFI et voici ce que j'obtiens :

```
$ tail -f /var/log/messages
May 17 08:59:19 localhost kernel: [ 255.088000] pccard: CardBus card inserted into slot 0
May 17 08:59:20 localhost kernel: [ 256.268000] acx: Loaded combined PCI/USB driver,
firmware_ver=default
May 17 08:59:20 localhost kernel: [ 256.268000] acx: compiled to use 32bit I/O access. I/O
timing issues might occur, such as non-working firmware upload. Report them
May 17 08:59:20 localhost kernel: [ 256.268000] PCI: Enabling device 0000:01:00.0 (0000 -> 0002)
May 17 08:59:20 localhost kernel: [ 256.272000] ACPI: PCI Interrupt 0000:01:00.0[A] -> Link
[LNKA] -> GSI 9 (level, low) -> IRQ 9
May 17 08:59:20 localhost kernel: [ 256.272000] acx: found ACX111-based wireless network card
at 0000:01:00.0, irq:9, phymem1:0x14020000, phymem2:0x14000000, mem1:0xccb00000,
mem1_size:8192, mem2:0xcc800000, mem2_size:131072
May 17 08:59:20 localhost kernel: [ 256.272000] acx: loading firmware for acx1111 chipset with
radio ID 16
May 17 08:59:21 localhost kernel: [ 256.384000] acx: firmware image 'acx/1.2.1.34/tiacx111c16'
was not provided. Check your hotplug scripts
May 17 08:59:21 localhost kernel: [ 256.392000] acx: firmware image 'acx/1.2.1.34/tiacx111' was
not provided. Check your hotplug scripts
May 17 08:59:21 localhost kernel: [ 256.392000] acx: reset_dev() FAILED
May 17 08:59:21 localhost kernel: [ 256.396000] ACPI: PCI interrupt for device 0000:01:00.0
disabled
May 17 08:59:21 localhost kernel: [ 256.412000] acx_pci: probe of 0000:01:00.0 failed with
```

```
error -5
May 17 08:59:21 localhost kernel: [ 256.412000] usbcore: registered new interface driver acx_usb
```

A première vue, ce genre d'informations peut impressionner, mais en fait en y regardant de plus près, il est 'clairement' indiqué que le système n'arrive pas à charger le firmware tiacx111c16.

```
May 17 08:59:21 localhost kernel: [ 256.384000] acx: firmware image 'acx/1.2.1.34/tiacx111c16'
was not provided
```

A partir de cette information, la démarche est toujours la même, sachant que google est ton ami, une petite recherche sur tiacx111c16 va nous conduire rapidement sur [ce site](#) [5].

Il ne reste donc plus qu'à télécharger le fichier tiacx111c16 dans le répertoire **`lib/firmware/2.6.20-15-generic/acx/1.2.1.34`**

Et maintenant, lors de l'insertion de la carte les LEDs de celle-ci clignotent et le système tente de se connecter. J'en profite pour installer knetworkmanager qui est beaucoup plus évolué que kwifimanager que j'utilisais jusqu'à présent.

```
sudo aptitude install knetworkconf knetworkmanager
sudo aptitude remove kwifimanager
```

Il est impératif d'avoir un fichier `/etc/network/interfaces` épuré comme ceci

```
# The loopback network interface
# This is a list of hotpluggable network interfaces.
auto lo
# They will be activated automatically by the hotplug subsystem.
iface lo inet loopback
auto eth0
auto wlan0
```

pour que la ligne **Se loguer à un autre réseau sans fil** apparaisse dans le menu de KNetworkManager.

Finalement, mes essais de connexion aussi bien en WPA qu'en WEP ne fonctionnent pas. Voilà les informations qu'on peut trouver dans `/var/log/messages`

```
Jun 17 09:33:42 localhost kernel: [ 190.680000] acx v0.3.36: net device wlan0, driver compiled
against wireless extensions 21 and Linux 2.6.20-16-generic
Jun 17 09:33:42 localhost kernel: [ 190.684000] usbcore: registered new interface driver acx_usb
Jun 17 09:33:43 localhost kernel: [ 191.200000] ADDRCONF(NETDEV_UP): wlan0: link is not ready
Jun 17 09:33:45 localhost kernel: [ 193.660000] wlan0: association FAILED: peer sent Status
Code 12 (Assoc denied (reason outside of 802.11b scope) -- maybe MAC filtering by peer?)
Jun 17 09:33:47 localhost kernel: [ 195.164000] wlan0: association FAILED: peer sent Status
Code 12 (Assoc denied (reason outside of 802.11b scope) -- maybe MAC filtering by peer?)
...
```

Visiblement, il va donc falloir encore utiliser ndiswrapper

```
sudo aptitude install ndiswrapper-common ndiswrapper-utils-1.9
sudo ndiswrapper -i /usr/local/D-Link/GPLUS.inf
```

Il faut blacklister le driver acx en ajoutant **acx** à la fin du fichier `/etc/modprobe.d/blacklist`.

Lors de l'insertion de la carte wifi le fichier `/var/log/messages` enregistre les informations suivantes :

```
Jun 17 09:48:42 localhost kernel: [ 1090.908000] pccard: CardBus card inserted into slot 0
Jun 17 09:48:42 localhost kernel: [ 1091.432000] acx: Loaded combined PCI/USB driver,
firmware_ver=default
Jun 17 09:48:43 localhost kernel: [ 1091.432000] acx: compiled to use 32bit I/O access. I/O
timing issues might occur, such as non-working firmware upload. Report them
Jun 17 09:48:43 localhost kernel: [ 1091.436000] usbcore: registered new interface driver
acx_usb
Jun 17 09:48:43 localhost kernel: [ 1091.612000] ndiswrapper: driver gplus (D-
Link,04/09/2004,6.0.0.18) loaded
Jun 17 09:48:43 localhost kernel: [ 1091.612000] PCI: Enabling device 0000:01:00.0 (0000 ->
```

0002)

Jun 17 09:48:43 localhost kernel: [1091.612000] ACPI: PCI Interrupt 0000:01:00.0[A] -> Link [LNKA] -> GSI 9 (level, low) -> IRQ 9

Jun 17 09:48:43 localhost kernel: [1092.356000] ndiswrapper: using IRQ 9

Jun 17 09:48:44 localhost kernel: [1092.896000] wlan0: ethernet device 00:0f:3d:58:d2:60 using NDIS driver: gplus, version: 0x5000200, NDIS version: 0x501, vendor: 'TNET1130', 104C:9066.5.conf

Jun 17 09:48:44 localhost kernel: [1092.896000] wlan0: encryption modes supported: WEP; TKIP with WPA

Conclusions

Licence Creative Commons by-sa 3.

URL source: <https://www.coagul.org/drupal/publication/comment-faire-fonctionner-carte-wifi-dwl-g650-kubuntu-feisty>

Liens:

- [1] <https://www.coagul.org/drupal/rubrique/materiel>
- [2] http://www.coagul.org/article.php3?id_article=448
- [3] http://www.coagul.org/article.php3?id_article=533
- [4] http://www.coagul.org/article.php3?id_article=597
- [5] <http://acx100.sourceforge.net/wiki/Firmware>