Linux IR HOWTO

Werner Heuser, < *wehe@snafu.de* >
Traduction française par Mathieu Arnold < *arn_mat@club-internet.fr* > v2.4, 9 Février 1999

Une introduction à Linux et aux périphériques infrarouges, et comment utiliser les logiciels fournis par le projet Linux/IrDA. Ce paquetage utilise le standard IrDA(TM). IrDA(TM) est un standard industriel utilisé pour les communications infrarouges sans fils, et la majorité des ordinateurs portables construits après janvier 1996 sont équipés de transmetteurs infrarouges compatibles IrDA. Les ports infrarouges permettent de communiquer avec des imprimantes, modems, fax, LAN et d'autres portables. La vitesse de transmission va de 2400bps à 4Mbps. La pile Linux/IrDA supporte IrLAP, IrLMP, IrIAS, IrIAP, IrLPT, IrCOMM, IrOBEX, et IrLAN. De nombreux protocoles sont implémentés en tant que clients et serveurs. Les connections multiples via IrLAP¨sont aussi supportées, via plusieurs périphériques IrDA en même temps. Le projet Linux/IrDA à démarré à la fin de l'année 1997 et son statut est toujours expérimental, alors, n'espérez pas que tout marchera du premier coup. D'après ce que je sais, Linux/IrDA est la _seule_ implémentation libre disponible actuellement. Le contrôle à distance (Remote Control : RC) via infrarouge n'est pas le but de ce projet, malgré tout, ce sujet est traité partiellement dans ce HOWTO.

Table des matières

1 Introduction

Better red, than dead. - Unknown AuthorEss

Depuis 2.1.131 et 2.2.0, Linux/IrDA fait partie du noyau. Notez toutefois que le statut de ce projet est toujours expérimental. Si vous avez besoin d'informations à propos du support Linux/IrDA pour les noyaux 2.0.x, allez voir l'ancienne version de ce howto disponible à *http ://www.snafu.de/˜wehe/index_li.html*. Mais d'après ce que je sais, ce paquetage n'est plus maintenu.

Les compagnies et les développeurs qui aimeraient participer à ce projet devraient contacter le projet Linux/IrDA à *http ://www.cs.uit.no/˜dagb/irda* ou moi à < *wehe@snafu.de* >.

Maintenant, un peu d'histoire. Le projet démarra à la fin de l'année 1997 avec le nom Linux/IrDA. A cause de problèmes dus au fait que le nom IrDA est une marque déposée par l'Infrared Data Association IrDA *http ://www.irda.org/*, le nom a été changé en Linux/IR. A la fin de l'année 1998, les relations entre nous et eux devinrent meilleures, et le nom redevint Linux/IrDA.

Ce document est basé sur la partie "Comment l'utiliser" du site Linux/IrDA *http ://www.cs.uit.no/˜dagb/irda/howto.html*. J'ai aussi inclus des informations provenant de l'équipe Linux/IrDA, de la liste de diffusion, ainsi que d'autres sources.

Ce document fait partie du LINUX DOCUMENTATION PROJECT *http ://metalab.unc.edu/LDP*.

La dernière version de ce document est disponible à *http ://www.snafu.de/˜wehe/index_li.html* (version anglaise) et *http ://www.freenix.org/unix/linux/HOWTO/IR-HOWTO.html* (version française)

J'ai essayé de vérifier toutes les informations, mais je n'ai pas tout le matériel infrarouge nécessaire, alors, si cela ne marche pas pour vous, ne m'accusez pas.

Si vous avez des commentaires ou des questions, n'hésitez pas à me contacter. Je sais que ce document n'est ni fini ni parfait, mais j'espère qu'il vous sera utile.

<*Werner Heuser*>

2 Prérequis

- BIOS
  - Assurez vous que le port infrarouge est bien activé dans le BIOS, et vérifiez quelle interruption et quel port il utilise.
- Puce contrôleur infrarouge
  - Assurez vous que votre port infrarouge soit bien détecté par Linux. Pour plus de détails, référez vous à la section "Aperçu du matériel supporté" ci dessous.
- modutils
  - Assurez vous que vous avez bien modutils 2.1.x (`insmod -version`). J'utilise la version 2.1.121.
- Librairies partagées
  - La librairie `libc.so.5` et le chargeur `ld-linux.so.1` doivent être disponibles.
  - Mais `glibc2` aka `libc6` devrait aussi marcher.
  - Je ne suis pas sûr que vous ayez besoin de la librairie `zlib` si vous utilisez la compression de données.
- Interfaces graphiques :
  Il y a actuellement deux interfaces graphiques pour Linux/IrDA qui sont en développement :
  - GNOBEX, une application GNOME développée par Dag Brattli *http ://www.cs.uit.no/˜dagb/irda/irda.html* avec le support du glisser/déposer depuis le gestionnaire de fichier GNOME (gmc). Il indique aussi la progression du transfert en cours et donne de bons messages d'erreurs quand quelque chose tourne mal. L'interface graphique n'est pas terminée pour autant, mais si vous voulez la tester, vous aurez besoin du module Perl-GTK+.
  - KDE, une application KDE développée par Thomas Davis. Allez faire un tour sur sa page *http ://www.jps.net/tadavis/irda*.

  Pour vous autres lecteurs - voici votre chance de contribuer ! Ces deux interfaces graphiques nécessitent des icônes. Toutes les icônes doivent :

a) Avoir une taille fixe (je pense que 48x48 pixels semble être la taille commune).
b) Grosse & petite (les petites sont pour être dockées).
c) 16 couleurs.
d) Libres de tous droits.
e) SVP, ne copiez pas des icônes microsoft !
Contactez les développeurs.

– Sécurité
  - **Très important**, vous devez faire un `sync` sur vos disques ! ! ! Vous aurez peut être à rebooter votre machine. N'avez vous pas lu le désistement de responsabilité ?
– Divers
  - Autres programmes utiles : APSFILTER, EZ-Magic, MagicFilter ou quelque chose de similaire pour configurer les imprimantes.

3 Noyau

S'il vous plaît, lisez le *Kernel-HOWTO* `<http://www.freenix.org/unix/linux/HOWTO/Kernel-HOWTO.html>` pour trouver plus d'informations sur la compilation. Vous trouverez les sources de Linux/IrDA dans :

`/usr/src/linux/net/irda` (protocoles)

`/usr/src/linux/drivers/net/irda` (pilotes de périphériques)

`/usr/src/linux/include/net/irda` (fichiers d'en-têtes)

3.1 Configuration générale

- Assurez vous que vous utilisez bien les sources du noyau *noyau 2.2.x*. Si vous n'êtes pas sûr de la version de votre noyau, essayez `uname -r`.

- Récupérez le dernier patch du projet Linux/IrDA *http ://www.cs.uit.no/˜dagb/irda/snapshots/*. Mettez le dans `/usr/src` ou à l'endroit où vous avez les sources de votre noyau, et appliquez quelque chose ressemblant à (remplacez le nom du patch `patch-2_2.0-irdaXXX` par le nom du fichier que vous avez) :

```
cd /usr/src
tar xvzf patch-2_2.0-irdaXXX.tar.gz
cd linux
patch -p1 -l < ./patch-2_2.0-irdaXXX
```

- Le support pour le code expérimental doit être activé (CONFIG_EXPERIMENTAL).

- Activez sysctl dans le "General Setup" (CONFIG_SYSCTL).

- Vous devriez avoir *proc file system support* (CONFIG_PROC_FS).

- De même *serial support* pour les fonctionnalités SIR (CONFIG_SERIAL).

- Je ne suis pas certain qu'il y ait besoin d'y avoir le *printer support* pour utiliser une imprimante avec Linux/IrDA (CONFIG_PRINTER). Mais je suppose que cette fonctionnalité n'est pas nécessaire

- Le *Networking support* doit être activé (CONFIG_NET).

- Assurez vous que vous avez le *module support* (CONFIG_MODULES) dans votre noyau ! Testez en faisant `lsmod`.

- De même le support de `kerneld` (CONFIG_KERNELD) est recommandé. Mais `kmod` (CONFIG_KMOD) marche aussi. Une noyau monolithique semble aussi marcher.

- Matt Francis a écrit : "J'ai remarqué que certains modules ont besoin non seulement du support pour les modules, mais aussi du *misc user device support*." (CONFIG_UMISC).

Si vous avez juste appliqué le patch Linux/IrDA, vous ne devriez pas avoir à faire un `make clean`, alors, cela devrait aller un peu plus vite. Je vous suggère de faire un truc dans ce style :

```
make dep && make all && make modules && make install && make modules_install
```

Si vous avez des erreurs vraiment bizarres, alors, essayez de recompiler après un `make clean`.

3.2 Paramètres spécifiques à IrDA

Voici ce que j'ai suggéré pour `../linux-2.2.x/Documentation/Configure.help`, des morceaux viennent de Dag Brattli et Andreas Butz :

3.2.1 IrDA subsystem support

CONFIG_IRDA

IrDA(TM) est un protocole industriel standardisé pour les communications infrarouges sans fils. Les ports infrarouges vous permettent de communiquer avec des imprimantes, modems, fax, réseaux et ordinateurs portables. La vitesse de transmission varie de 2400bps à 4Mbps. Pour utiliser ces fonctionnalités, vous aurez besoin du paquetage irda_utils fournis par le projet Linux/IrDA (). Pour de plus amples informations, vous pouvez trouver le IR-HOWTO à . Pour l'instant, il est conseillé de compiler le support IrDA en tant que module uniquement (Référez vous à Documentation/modules.txt). Notez que le support Linux/IrDA est toujours expérimental.

Protocole IrDA

– Protocole IrLAN CONFIG_IRLAN
  Compile le pilote IrDA réseau. Utilisez "ifconfig eth0 <IP-NUMBER>" pour le configurer. - Répondez juste Y.
– Support des clients IrLAN CONFIG_IRLAN_CLIENT
  Si vous connectez des périphériques infrarouges via IrLAN, l'un doit être serveur et l'autre client. Vous pouvez utiliser les deux en même temps. Le premier à se connecter sera le client. - Répondez juste Y.
  Note : Les derniers patch incluent le support peer-to-peer à la place.
– Support pour le serveur IrLAN CONFIG_IRLAN_SERVER
  Si vous connectez des périphériques infrarouges via IrLAN, l'un doit être serveur et l'autre client. Vous pouvez utiliser les deux en même temps. Le premier a se connecter sera le client. - Répondez juste Y.
  Note : Les derniers patch incluent le support peer-to-peer à la place.
– Protocole IrOBEX CONFIG_IROBEX
  IrOBEX est un protocole pour échanger des objets (fichiers, cartes de visite, etc.) à travers une connexion infrarouge. Vous pouvez l'utiliser pour échanger des fichiers entre Linux et un PALM III. IrOBEX peut aussi être utilisé entre deux machines Linux, entre une machine Linux et une machine Windows95, etc. - Répondez juste Y.
– Protocole IrCOMM CONFIG_IRCOMM
  Via IrCOMM, il est possible de communiquer avec des téléphones cellulaires, etc. Pour utiliser ce service, vous aurez à créer un périphérique avec "mknod /dev/irnine c 60 64", il marche comme /dev/ttySx. - Répondez juste Y.
  Note : les nombres majeurs et mineurs ne sont toujours pas officiels. Pour les dernières améliorations, (IrSocket est en chemin !), allez faire un tour sur la page de Takahide Higuchi .
  Note : Actuellement, IrCOMM semble faire planter le noyau assez facilement, alors, vous devriez probablement attendre le prochain patch.

– Support des clients IrLPT CONFIG_IRLPT_CLIENT
  Dites Y ici si vous voulez le support pour le protocole client IrLPT. Si vous voulez le compiler en tant que module, dites M ici et lisez Documentation/modules.txt. Le protocole client IrLPT peut être utilisé pour imprimer des documents sur des imprimantes compatibles IrDA telle la HP-5MP, ou sur un adaptateur IrLPT tel le ACTiSYS IR-100M. - Répondez juste Y.
– Support du serveur IrLPT CONFIG_IRLPT_SERVER
  Dites Y ici si vous voulez compiler le support pour le protocole serveur IrLPT. Si vous voulez le compiler en tant que module, dites M ici et lisez Documentation/modules.txt. Le protocole serveur IrLPT rends possible l'utilisation d'une machine Linux en tant qu'imprimante infrarouge pour d'autres portables. Alors, si votre machine Linux a une imprimante, d'autres portables pourront l'utiliser via la liaison infrarouge. - Répondez juste Y.

Options du protocole IrDA CONFIG_IRDA_OPTIONS

Vous avez la possibilité de définir certaines options du protocole IrDA
– Cache last LSAP CONFIG_IRDA_CACHE_LAST_LSAP
  Dites Y ici si vous voulez que IrLMP cache le dernier LSAP utilisé. Ceci est intéressant puisque la majorité des trames seront envoyées/reçues via la même connexion. L'activation de cette option vous fera gagner une recherche dans un tableau à chaque trame transmise. Si vous êtes indécis, dites Y.
– FAST RRs CONFIG_IRDA_FAST_RR
  Utilisez cette option si vous voulez envoyer les trames RR (Receive Ready) plus vite lorsque la queue est vide. Cela vous donnera de biens meilleurs temps de réponse, mais consommera plus de puissance, à cause des rebonds des trames RR.
– Recycle RRs CONFIG_IRDA_RECYCLE_RR
  Lorsque le protocole IrLAP fonctionne normalement, il envoie beaucoup de petites trames RR (Receive Ready) à travers la liaison (tout du moins, lorsqu'il n'a rien d'autre à faire). Répondre Y ici demandera à IrLAP de recycler ces trames, évitant par là même plein de alloc_skb et de kfree_skb. Pour réaliser cela, il va juste garder une trame ce qui est suffisant dans la majorité des cas.
– Debug information CONFIG_IRDA_DEBUG
  Dites Y ici si vous voulez que IrDA donne des informations de débogage à votre syslog. Vous pouvez changer le niveau de débogage dans /proc/sys/net/irda/debug.
  Si vous êtes indécis, dites Y (car il deviendra plus simple de dénicher les bogues).

IrDA compressors CONFIG_IRDA_COMPRESSION

Vous pouvez utiliser les méthodes de compression BZIP2 et BSD. Cela ne fait pas partie du standard IrDA. Cela permettra à deux machines Linux de compresser leur trafic. Cela devrait être compatible avec les autres périphériques IrDA, mais les communications ne seront alors pas compressées.

– Deflate compression (experimental) CONFIG_IRDA_DEFLATE
  Dites Y ici si vous voulez compiler le support pour le protocole de compression Deflate. Si vous voulez le compiler en tant que module, dites M et référez vous à Documentation/modules.txt. La compression deflate (GZIP) est exactement la même que celle utilisée par le protocole PPP. Activer cette option créera un module nommé irda_deflate.o.
– BZIP2 compression CONFIG_IRDA_BZIP2
  Pas d'aide disponible.
– BSD compression CONFIG_IRDA_BSD
  Pas d'aide disponible.

3.2.2 Pilotes de périphériques infrarouges

Trois types de pilotes sont disponibles : série, dongle et FIR. Ils apparaîtrons dans /proc/net/dev (irda0) après initialisation.

IrTTY (utilise le pilote série) La majorité des puces IrDA supportent StandardInfraRed (SIR), qui marche jusqu'à 115200 bps et émmule un port série (UART 16550A). Sur beaucoup de portables, ce port est détecté par le pilote série du noyau, référez vous a “dmesg”. IrTTY connecte les services Linux/IrDA à ce port. - Vous devriez dire Y ici.

– Support du dongle série CONFIG_IRTTY_SIR
  Dites Y ici si vous voulez compiler le support pour la IrTTY line discipline. Si vous voulez le compiler en tant que module, dites M et référez vous à Documentation/modules.txt. IrTTY rends possible l'utilisation du pilote série de Linux pour tous les ports IrDA qui sont compatibles 16550. La majorité des puces le sont, par conséquent, vous devriez aussi dire Y à cette option. L'utilisation de IrTTY limitera toutefois la vitesse de connexion à 115200 bps (mode IrDA SIR). Si vous n'êtes pas sûr, dites Y.

Support des dongles CONFIG_DONGLE

Pour l'instant 4 dongles (adaptateurs infrarouges pour port série) sont supportés. Le dongle est un périphérique infrarouge qui se connecte au port série, si vous n'avez pas de port infrarouge intégré à votre machine. Si vous utilisez un dongle sur un ordinateur portable, vous aurez certainement à désactiver le support IrDA dans le BIOS.

– Dongle ESI JetEye PC CONFIG_ESI_DONGLE
  Dites Y ici pour compiler le support pour les dongles Extended Systems JetEye PC. Si vous voulez le compiler en tant que module, dites M et référez vous à Documentation/modules.txt. Ce dongle se branche sur un port série 9 broches, et ne peut être utilisé que par IrTTY. Pour activer le support pour le dongle ESI, vous aurez à insérer “irattach -d esi” dans le script /etc/irda/drivers.
– Dongle ACTiSYS IR-220L and IR220L+ CONFIG_ACTISYS_DONGLE
  Dites Y ici pour compiler le support pour les dongles ACTiSYS IR-220L et IR220L+. Si vous voulez le compiler en tant que module, dites M et référez vous à Documentation/modules.txt. Ce dongle se branche sur un port série 9 broches, et ne peut être utilisé que par IrTTY. Pour activer le support pour les dongles ACTiSYS, vous aurez à insérer “irattach -d esi” ou “irattach -d actisys_plus” dans le script /etc/irda/drivers.
– Dongle Tekram IrMate 210B CONFIG_TEKRAM_DONGLE
  Dites Y ici pour compiler le support pour le dongle Tekram IrMate 210B. Si vous voulez le compiler en tant que module, dites M et référez vous à Documentation/modules.txt. Ce dongle se branche sur un port série 9 broches, et ne peut être utilisé que par IrTTY. Pour activer le support pour les dongles Tekram, vous aurez à insérer “irattach -d tekram” dans le script /etc/irda/drivers.
– Dongle GirBIL CONFIG_GIRBIL_DONGLE
  Dites Y ici pour compiler le support pour le dongle Greenwich Instruments GirBIL. Si vous voulez le compiler en tant que module, dites M et référez vous à Documentation/modules.txt. Ce dongle se branche sur un port série 9 broches, et ne peut être utilisé que par IrTTY. Pour activer le support pour les dongles Greenwich, vous aurez à insérer “irattach -d girbil” dans le script /etc/irda/drivers.

Support de FIR FastInfraredSupport (FIR) (support de l'infrarouge rapide) nécessite un contrôleur spécial, qui supporte jusqu'à 4Mbps. Dites juste "Y"

– NSC PC87108 CONFIG_NSC_FIR
  La puce FIR NationalSemiConductor NSC PC87108 (utilisée par exemple dans l'IBM Thinkpad 560X et le dongle ACTiSYS IR2000). La puce FIR NSC PC87338 FIR est probablement aussi supportée. Le pilote

supporte les vitesses SIR, MIR et FIR (4Mbps). - Dites juste Y.
- Winbond W83977AF (IR) CONFIG_WINBOND_FIR
  La puce FIR Winbond W83977AF (IR) (utilisée par exemple dans le Corel Netwinder PC). Le pilote supporte les vitesses SIR, MIR et FIR (4Mbps). - Dites juste Y.
- Sharp UIRCC CONFIG_SHARP_FIR
  Dites Y ici pour compiler le support pour le chipset Sharp UIRCC IrDA. Si vous voulez le compiler en tant que module, dites M et référez vous à Documentation/modules.txt. Ce chipset est utilisé dans les portables Toshiba de la gamme Tecra.

4 Utilitaires Linux/IrDA

- Utilisez la dernière version des `irda-utils` disponible à *http ://www.cs.uit.no/˜dagb/irda/irda-utils/*.
- Détarez le paquetage en faisait `tar xvzf irda-utils<VERSION>`. Je vous recommande de le mettre dans `/usr/src`.
- Faites un `make depend`.
- Faites un `make clean` (ce n'est pas nécessaire si vous compilez le paquetage pour la première fois).
- Faites un `make all` pour compiler tous les exécutables.
- Faites un `make install`, cela installe `irattach` et `irmanager` au bon endroit, et installe quelques fichiers de configuration dans `/etc/irda`.

Une recommandation de Bjoern Hansson <Bjorn.Hansson@signal.uu.se> : Si `make depend` échoue sur `stdef.h` et `stdarg.h` ajoutez juste `-I/usr/lib/gcc-lib/i586-linux/egcs-2.90.29/include/` ou le chemin adéquate avec votre configuration à la ligne `SYS_INCLUDES` du fichier `Makefile`.

5 Configuration

5.1 Configuration générale

- Tout d'abord vous devriez mettre vos périphériques infrarouges à portée l'un de l'autre. Bien que le service Linux/IrDA détecte peut être automatiquement les nouveaux périphériques, j'ai bonne expérience de la configuration quand les périphériques sont à portée l'un de l'autre.
- Gardez vos périphériques infrarouges ensembles, dans un espace d'un mètre maximum et dans un angle de 30 degrés. Ils doivent être en vis à vis. Si cela n'est pas possible, utilisez un miroir (un CD de M$ non utilisé fera l'affaire).
- Ajoutez les lignes suivantes à votre `/etc/conf.modules` :

  ```
  alias tty-ldisc-11 irtty          #
  alias char-major-60 ircomm_tty    # Si vous voulez le support IrCOMM
  ```

  Je ne suis plus tout à fait sûr de ces paramètres. Il y a des informations contradictoires sur les page de Dag Brattli et sur la liste. Ces paramètres étaient utilisés avec les anciens paquetages Linux/IrDA, ces paramètres ont certainement changés dans les versions 2.2.x des noyaux.
- Allez faire un tour dans `/etc/irda`. ces fichiers sont similaires à ceux dans `/etc/pcmcia`. Editez `/etc/irda/drivers` pour que cela soit en accord avec votre configuration. La majorité des gens utiliseront `irattach` depuis ce fichier. les fichiers sont :

  ```
  Makefile
  network*
  network.redhat*
  serial
  drivers
  ```

```
network.opts
obex
printer
```

– Lancez un `depmod -a`.

5.2 IrManager

Dag Brattli a écrit : *IrManager* [...]. est un démon utilisateur qui s'inspire et qui est assez similaire à *cardmgr* utilisé pour contrôler les cartes PCMCIA.

IrManager recevra les événements qui proviennent de l'espace noyau de la pile du protocole. Quand *IrManager* reçoit un événement, il peut exécuter des commandes shell ou des scripts, par conséquent, j'ai ajouté le répertoire `/etc/irda` pour contenir ces scripts. [...]

Par exemple, si IrLMP découvre un périphérique qui peut fournir un service IrLAN, et qu'aucun client IrLAN ne s'est enregistré, alors, IrLMP enverra un événement à IrManager et lui fera faire un "modprobe" sur le module requis. [...]

Lorsque les clients (applications utilisateurs) sont prêts à effectuer la configuration des paramètres de communication et de l'utilisateur, ils peuvent également contacter IrManager, de façon à ce qu'il exécute le bon script. Par exemple, IrLAN enverra un événement EVENT_IRLAN_START quand le canal de données sera prêt pour échanger des trames ethernet. Quand IrManager reçoit cet événement, il exécute `/etc/irda/network start <devname>` pour configurer l'interface réseau. Ce script network est d'ailleurs le même que celui utilisé par le code PCMCIA, et vu que j'utilise sa variante RedHat, il finira par exécuter `/sbin/ifup <devname>`.

Par conséquent, en utilisant l'IrManager, j'ai "juste" à faire cela quand je démarre la pile :

```
irattach /dev/ttyS2 &
irmanager -d 1           # -d 1 signifie : démarre la reconnaissance automatique
```

et quand mon portable découvre le fournisseur IrLAN (un HP Netbeamer dans mon cas) il demandera à *IrManager* de charger le module `irlan_client`. Quand la connexion est établie et prête, il demandera l'exécution de `/etc/irda/network start eth0`. Si la connexion est perdue, il lui demandera de supprimer l'interface en lançant `/etc/irda/network stop eth0`.[...]

C'est tout ce que vous avez à faire si vous utiliser une RedHat. Si vous utilisez une distribution qui n'a pas `/sbin/ifup`, alors, vous devriez copier `/etc/pcmcia/network.opts` vers `/etc/irda/network.opts` ou configurer le fichier vous même.

Si vous voulez utiliser le serveur IrLAN, vous aurez quand même à faire un `modprobe irlan_server` avant de lancer `irmanager` **sans** `-d 1`.

Et tout comme le `cardmgr`, vous aurez (et si vous le voulez) un bip quand la connexion est établie et prête et quand elle est déconnectée !!!

J'espère que nous pourrons ajouter de tels scripts pour tous les autres services/clients qui ont besoin d'une configuration dans l'espace utilisateur. Cela serait super d'avoir un script `/etc/irda/printer` pour configurer les imprimantes compatibles IrDA(TM) de telle sorte que si vous entrez dans le champ d'une telle imprimante, IrManager charge le module `irlpt_client` et configure aussi ce qui doit l'être pour utiliser cette imprimante.

J'espère aussi que nous pourrons utiliser le fichier de configuration pour configurer les ports et les pilotes IrDA(TM). Quelque chose dans le style :

```
Device Drivers
  module "irtty" script="irattach /dev/ttyS2"
  module "smc_ircc" irq=11 port=0x34f
```

de telle sorte que IrManager soit capable de charger et démarrer tout ça quand il est exécuté. De cette façon, il serait juste de lancer IrManager depuis `/etc/rc.d/init.d/irda` et le reste sera plug & play. Il ne serait plus nécessaire de démarrer manuellement les programmes et de configurer les périphériques. Pour l'instant, lorsque `irmanager` reçoit les événements suivants, il fera :

EVENT_IRLAN_START, démarre et configure le périphérique avec `/sbin/ifup <dev>`

EVENT_IRLAN_STOP, ferme l'interface avec `/sbin/ifdown <dev>`

cela peut bien sur être changé simplement par l'utilisateur, si ce n'est pas la façon qu'il a de le faire.

5.3 Pilotes de bas niveau

Il y a trois sortes de pilotes de bas niveau : SIR, dongle et FIR. Si le bon pilote est trouvé par le noyau, vous aurez un message ressemblant à :

```
IrDA irda_device irda0 registered.
```

5.3.1 SIR

- Essayez de trouver quel port série est utilisé par le périphérique infrarouge. Vous pouvez le faire en regardant ce que donne `dmesg`. Si le support série est sous forme de module vous pouvez faire un `insmod serial`. Cherchez une entrée telle :

  ```
  Serial driver version 4.25 with no serial options enabled
  ttyS00 at 0x03f8 (irq = 4) is a 16550A     #Permier port série /dev/ttyS0
  ttyS01 at 0x3000 (irq = 10) is a 16550A    #ex. port IR
  ttyS02 at 0x0300 (irq = 3) is a 16550A     #ex. port modem PCMCIA
  ```

  Si ce n'est pas le cas, soit vous n'avez pas activé le support infrarouge dans le BIOS ou votre périphérique infrarouge n'est pas détecté par le noyau. Pour l'instant, je ne connais que deux portables qui ont ce comportement, le HP OmniBook 800 et les Toshiba Libretto. Je ne suis pas certain du rôle que joue le PnP dans la détection du port IR. Si vous n'êtes pas sûr, essayez et donnez moi le résultat.
- Dans certaines situations, vous aurez peut être à faire un `setserial /dev/ttyS<0-2> port 0xNNNN irq M` pour donner les bonnes valeurs à votre port infrarouge série, surtout si le port série est sur une ligne série séparée. Normalement, il n'est pas nécessaire de modifier ces valeurs ! Pour plus d'informations, référez vous à la FAQ ci dessous.
- Si vous n'utilisez pas `kerneld` ou `kmod` insérez le module irda avec `modprobe irda`.
- Faites un `lsmod`. Les modules `irda` et `irtty` devraient apparaître.
- Un coup d'oeil dans `/var/log/messages` devrait vous afficher `"Serial connection established"` maintenant.
- Tapez `irmanager -d1`, cela lancera les programmes nécessaires, tels `irattach`.
- Donnez à `irattach` un peu de temps (ex : sept secondes), pour détecter les autres périphériques infrarouges. Ensuite, regardez ce que vous affiche le noyau dans `/var/log/messages`. Cela devrait ressembler aux lignes suivantes (j'ai supprimé quelques lignes qui n'étaient pas en relation avec Linux/IrDA) :

  ```
  Jan  2 12:57:26 japh kernel: ttyS00 at 0x03f8 (irq = 4) is a 16550A
  Jan  2 12:57:26 japh kernel: ttyS02 at 0x03e8 (irq = 4) is a 16550A
  Jan  2 12:57:26 japh kernel: Linux Support for the IrDA (tm) protocols (Dag Brattli)
  Jan  2 12:59:09 japh syslog: executing: 'echo 1 > /proc/sys/net/irda/discovery'
  Jan  2 12:59:09 japh syslog: Setting discovery to 1 exited with status 1
  Jan  2 12:59:09 japh syslog: + 0.1 Fri Jul 25 11:45:26 1997 Dag Brattli
  Jan  2 12:59:09 japh syslog: + 0.1 Fri Jul 25 11:45:26 1997 Dag Brattli
  Jan  2 12:59:09 japh syslog: Serial connection established.
  ```

```
Jan  2 12:59:09 japh kernel: IrDA irda_device irda0 registered.
Jan  2 13:01:22 japh syslog: executing: './drivers start '
Jan  2 13:01:22 japh syslog: Serial connection established.
Jan  2 13:01:42 japh syslogd: Printing partial message
Jan  2 13:01:42 japh 0.1 Fri Jul 25 11:45:26 1997 Dag Brattli
Jan  2 13:02:49 japh kernel: IrDA Discovered: japh
Jan  2 13:02:49 japh kernel:     Services: Computer
```

5.3.2 Connexion des dongles - Adaptateurs infrarouges sur port série.

Les dongles supportés actuellement sont le Extended Systems Inc. ESI-9680 JetEye, le Tekram IRmate 210B, le ACTiSYS IR220L et 2\20L+, le Greenwich GIrBIL.

Dag Brattli a écrit (modifié par wh) : "Pour utiliser un dongle, vous aurez à faire un truc comme :

```
modprobe tekram        # ou esi ou actisys
irmanager -d 1         #
irattach -d tekram     # ou -d esi ou -d actisys
```

Comme vous pouvez le voir, vous avez toujours à utiliser l'option `-d` avec `irattach` vu qu'il est possible d'avoir deux ports séries utilisant deux dongles différents au même moment (le tty que vous liez doit savoir à quel dongle il a à faire). Donc, si vous avez deux dongles et deux ports séries, vous aurez à faire un truc comme cela :

```
modprobe tekram
modprobe esi
irattach /dev/ttyS0 -d esi &
irattach /dev/ttyS1 -d tekram &
```

PS : Je n'essaierais pas de mettre les deux dongles l'un en face de l'autre, car je ne sais pas trop quelle serait la réaction de la pile :-)

Comme je n'ai pas de nouveaux dongles ACTiSYS 220L+ sous la main, je ne peut pas les tester. Et comme ces nouveaux dongles supportent une vitesse supplémentaire (28400bps), vous devez spécifier le dongle différemment avec `irattach` de telle sorte que le noyau sache quel dongle vous utilisez (et quel QoS peut être utilisé) :

```
irattach /dev/ttyS0 -d actisys     # pour le dongle 220L
irattach /dev/ttyS0 -d actisys+    # pour le dongle 220L+
```

L'implémentation actuelle du support des dongles n'a aucun état précis, il n'est donc pas possible d'utiliser les deux dongles ACTiSYS (220L et 220L+ en même temps (connecté à deux ports séries) pour l'instant. Si vous avez besoin de le faire, envoyez moi un e-mail (Dag Brattli) et j'y songerais peut être !"

Note : Quand j'ai essayé d'utiliser un modem infrarouge (Swissmod 56Ki, construit par Telelink AG) connecté à mon portable (l'IrDA n'y marche qu'avec Windows 95 du à un matériel non standard) j'ai du désactiver le support infrarouge depuis le BIOS pour le faire marcher !

5.3.3 Connexion des dongles - Adaptateur infrarouge pour carte mère

Le support pour le dongle ACTiSYS IR2000 a été implémenté dans un fichier appelé `px87108` que vous pouvez soit compiler dans le noyau, soit en tant que module :

```
irmanager -d 1
modprobe pc87108
```

ou insérez `modprobe pc87108` dans le fichier `/etc/irda/drivers` (je pense).

5.3.4 Infrarouge rapide (FIR = Fast InfraRed)

Le standard IrDA(TM) reconnaît *trois* vitesses différentes :

1. SIR = IrDA standard (Standard IrDA), jusqu'à 115kbps IrDA,
2. MIR = IrDA à vitesse moyenne (Medium Speed IrDA),
3. FIR = IrDA rapide (Fast IrDA) (4Mbps),
4. VFIR = IrDA très rapide (Very Fast IrDA) (16Mbps), semble devenir le standard futur

Jusqu'à 115,200 bps, beaucoup de contrôleurs infrarouges fonctionnent comme un port série. Jusqu'à 4Mbps, ils marchent dans le mode "FIR". Actuellement, seuls deux contrôleurs sont supportés : NationalSemiConductor NSC PC87108 (utilisé dans les IBM Thinkpad 560X) et la puce Winbond W83977AF (utilisé dans le PC Corel Netwinder). Vous pouvez lancer le service FIR en chargeant le modules approprié. Linux/IrDA se chargera de détecter vos périphériques à ce moment là.

6 Protocoles et connexions spécifiques

6.1 Connexion imprimante - IrLPT

IrLPT est en phase de développement active en ce moment. La partie cliente devrait marcher, mais le serveur pourrait planter.

- Supprimer toutes les impressions en attente avec `lprm "*"`.
- Si vous n'utilisez pas `kerneld` faites un `modprobe irtty`.
- Faites un `modprobe irlpt_client`.
- Vérifiez que le module est bien chargé avec `lsmod`. Il devrait y avoir `irda`, `irtty` et `irlpt_client`
- `cat /proc/misc` vous donne le *numéro mineur* du périphérique. C'est le premier chiffre sur la ligne avec irlpt0.
- faites un `su` pour devenir root, et faites un `mknod /dev/irlpt0 c 10 <numéro mineur>`. Note : quelque chose comme `./MAKEDEV irlpt0` n'est toujours pas possible, mais un `load_misc irlpt` marche, bien que je ne puisse pas le confirmer.
- Essayez d'écrire un petit fichier vers /dev/irlpt0 en `cat FILE >/dev/irlpt0` (ne pensez pas au format, c'est juste un test). Pour moi, ça n'a pas toujours marché, mais je n'ai jamais compris pourquoi.
- La meilleure façon est de changer votre `/etc/printcap` pour utiliser plutôt `/dev/irlpt0` ou même à la place de `/dev/lp1`. Référez vous au `Printing-HOWTO` pour plus de détails.
- Pour imprimer plus facilement, utilisez un logiciel d'impression tel APSFILTER, MagicFilter EZ-Magic (avec la RedHat il devrait aussi y avoir une interface graphique pour cette chose là). Faites toujours une copie de `/etc/printcap` avant :-).
- Exemple pour APSFILTER avec une imprimante HP 6P (non postscript, HP 6MP est postscript). Les deux questions utiles sont :
  "Avez vous une interface (s)série ou (p)parallèle ?" Réponse "p"
  "Quelle est le nom du périphérique pour votre imprimante parallèle ?" Réponse "`/dev/irlpt0`"
- Redémarrez le démon d'impression avec `kill -HUP <PID de lpd>`. Si vous utilisez un autre démon d'impression, utilisez la commande appropriée.
- Regardez si l'indicateur de connexion de votre imprimante montre des signes d'activité, par exemple, sur les HP 6P/6MP le petit voyant vert au dessus du port infrarouge s'allume (en bas à gauche, près du bac à papier).
- Je n'ai jamais réussi à imprimer des documents de plus d'environ 10 pages jusqu'à présent. Mais cela dépend peut être de la mémoire de mon système (16Mo). Mais il semble qu'il y ait un problème logiciel aussi, Thomas Davis a écrit : "Je ... limiterais IrLPT pour qu'il ne prenne pas trop de mémoire lors de l'impression d'un gros fichier.".

Takahide Higuchi a rapporté : "J'ai testé IrCOMM avec une imprimante (Canon BJC-80v) ayant un port IrDA et le protocole IrCOMM (pas IrLPT). J'ai pu imprimer de petits textes, mais il se peut que cela cause facilement un blocage total lorsque j' essaye d'imprimer un fichier postscript avec gs.".

Sur la page de Thomas Davis *http ://www.jps.net/tadavis/irda*, on peut lire que pour utiliser le serveur IrLPT, il faut :

```
/sbin/insmod irlpt_server
/sbin/mknod /dev/irlptd c 10 `grep irlptd /proc/misc|cut -f 1`
```

A ce moment là, le serveur IrLPT est prêt à recevoir des fichiers à imprimer. Maintenant, tout ce dont vous avez besoin est un script :

```
#!/bin/sh
#
while (true)
do
cat /dev/irlptd | lpr
done
```

6.2 Connexion LAN - IrLAN

- Vous pouvez connecter votre ordinateur sous Linux en utilisant IrLAN avec un autre périphérique, par exemple, une autre machine Linux avec IrLAN, un HP NetBeamer ou une machine Windows9* avec une interface infrarouge.
- Dag Brattli a écrit : "Si vous voulez utiliser IrLAN, vous devez faire un `modprobe irlan_client` avant `ifup eth0`. J'ai du supprimer toute la partie request_module() car cela nécessite un contexte non disponible dans le noyau.".
- Lancez `ifconfig eth0 up <addresse_ip> netmask <masque_sous_réseau>` pour le configurer avec une adresse IP et d'autres paramètres. Si le protocole est toujours en mémoire, vous pourrez commencer à communiquer. Il est possible d'utiliser l'outil `netcfg` de la RedHat pour le faire, cela rend les choses plus faciles. La prochaine fois, vous aurez juste à faire un `/sbin/ifup eth0`.
- Testez l'interface réseau en la pingant. Pour plus de détails, référez vous au *NET3-HOWTO*.
- N'oubliez pas d'ajouter une route, ex : `route add default gw <ip_passerelle>` ou `route add -host <cible> dev eth0`.
- Pingez une autre IP, pour tester la connexion.
- Pour des raisons de test, je recommande de n'utiliser qu'un seul portable et une seule interface infrarouge dans la même pièce. Si il y a des problèmes, regardez quels sont les différents modes possibles pour l'interface infrarouge. Testez les...

Pour une carte ou un dongle ACTiSYS FIR, vous aurez à faire :

```
irmanager -d1
/sbin/modprobe pc87108  # Supprimer irattach de /etc/irda/drivers, ou
                        # substituer irattach par modprobe !
```

Sur la machine 1 :

```
modprobe irlan_client # Pas vraiment nécessaire, vu que irmanager devrait le faire !
```

Sur la machine 2 (si vous n'avez pas un point d'accès)

```
modprobe irlan_server
```

Ne compilez pas `irlan_server` dans le noyau, car il n'aime pas ça du tout pour l'instant ! Vous devriez avoir configuré `/etc/sysconfig/network-scripts/ircfg-eth0` avec le bon réseau si vous utilisez deux machines. Si vous avez un point d'accès, alors, le paramétrage par défaut devrait être bon.

Remarquez que dans le dernier patch (2.2.0-irda1), irlan_client devrait appeler le périphérique `irlan0` par défaut, mais il est possible de modifier ce comportement en donnant `eth=1` comme option à `irlan_client` (`modprobe irlan_client eth=1` ou `options irlan_client eth=1` dans `/etc/conf.modules`).

La prochaine version d'IrLAN ne sera qu'un seul module, alors, vous n'aurez pas à penser au fait qu'il vous faut le client et/ou le serveur d'installé.

Il est possible de faire un `ifconfig irlan0 -broadcast` pour empêcher l'AP de vous submerger avec des trames de broadcast ! Cela peut être un problème si vous êtes connectés à un segment ethernet très large. Le seul problème sera que votre machine aura à démarrer toutes les communications, et donc, ne pourra pas marcher en tant que serveur (bon, vous devriez pouvoir trouver une machine fixe qui pourrait répondre aux requêtes ARP à votre place).

6.3 Connexion Palm III - IrOBEX

La partie IrOBEX semble être en constant développement. Par conséquent, les applications changent aussi. Par conséquent, je ne peux pas vous donner des informations exactes. Allez voir la page de Dag Brattli à *http ://www.cdpubs.com/hhsys/archives/66/10brattl.pdf*.

Le script `/etc/irda` est vraiment juste bon pour configurer les périphériques, en faisant le bon mknod pour `/dev/irobex` etc, pas pour démarrer des applications.

- Palm III -> Linux
  1) Terminal 1> `irattach /dev/ttyS<x>`
  2) Terminal 2> `load_misc irobex`
  3) Terminal 3> Démarrez `irobex_app` depuis le répertoire irobex. Je suppose que `irobex_app` ne marche plus maintenant. Maintenant, vous devez utiliser le programme `gtk/irobex` ! Vous avez besoin que la librairie gtk soit installée pour pouvoir utiliser ce programme. Une interface graphique devrait être développée par quelqu'un. Le programme à utiliser devrait être `irobex_receive`.
  4) Récupérez quelque chose depuis votre Palm III.
  5) Si tout va bien, vous pouvez aller voir le fichier qui vient d'arriver dans le répertoire depuis lequel a été lancé irobex_app (ou /tmp pour irobex_receive). Le nom de ce fichier dépendra de l'objet que vous avez transféré.
- Linux -> Palm III Cela devrait être possible, mais je n'ai pas d'informations pour l'instant.
- PPP Rui Oliveira a écrit : "Grâce au dernier patch IrCOMM (050998) de Takahide Higuchi, j'ai réussi à synchroniser et à établir une connexion PPP entre mon Palm III et ma machine Linux. J'utilise IRLink (provenant de IsComplete) pour rediriger le port série vers un port IR. La communication avec pilot-xfer marche sans aucun problème. Bien que j'aie réussi à établir une connexion PPP, je n'arrive toujours pas à récupérer des emails et à naviguer sur le Web. Cela est probablement du à des timeouts de connexion. Je suis en train d'essayer de résoudre ce problème.". Référez vous au *PPP-HOWTO* pour de plus amples informations sur le PPP.
- IrCOMM Jon Howell a écrit : "Je pensais essayer IrCOMM, car le Palm III peut être configuré pour rerouter les infos du port série vers le port IR (en utilisant IrLink de IS/Complete, disponible sur *www.palmcentral.com*), et ensuite, vous pouvez lancer un programme terminal (tel "PalmTelnet" en mode série) à travers IrDA. Je peux seulement supposer qu'il utilise le protocole IrCOMM. J'ai testé cette configuration entre deux Palm Pilots, mais là, je ne peux pas savoir quel protocole est utilisé.
  Je n'ai pas réussi à trouver une application au code IrCOMM. Je vois sur la page d'avancement qu'il y a un "support client (devrait commencer à être utilisable)" mais je ne trouve aucune documentation sur comment l'utiliser.".

6.4 Connexion avec un téléphone cellulaire.

D'après ce que je sais, quelques téléphones utilisent le standard IrCOMM, ex : Ericsson SH888 et NOKIA 8110. Benny Amorsen a écrit : "J'ai utilisé minicom avec le SH888, et j'ai réussi à établir une connexion (CONNECT 9600...). Je n'ai pas réussi à lui faire faire quoi que ce soit d'autre après la connexion, et raccrocher en arrêtant la porteuse ne marche pas. +++ J'ai bien travaillé, maintenant, il sait raccrocher. Je dois dire que le SH888 n'est pas loin de marcher très bien avec Linux/IrDA, et devrait même marcher avec les noyaux 2.0.x. Je n'utilise que les noyaux 2.1.".

Il se peut que d'autres téléphones utilisent le standard IrOBEX, allez faire un tour dans la section Palm III pour apprendre à faire une connexion.

Carlos Vidal a écrit : "Corrigez moi si j'ai tout faux, mais il me semble que les téléphones NOKIA ne contiennent pas de modem matériel, mais quelque chose assez similaire aux WinModems pour les PC. Car à chaque fois que NOKIA écrit un truc à propos de communication via modem, ils utilisent le même nom de "Modem Windows logiciel" (ou quelque chose de similaire). Ce qui est actuellement confirmé par le fait qu'il faut utiliser les logiciels NOKIA pour Windows (nommés : NOKIA Cellular Data Suite).

Joonas Lehtinen a écrit : "Cela est vrai pour les modèles 61xx. Les modèles 8810, 9000(i) et 9110 devraient marcher (ils ont un modem intégré). Mon N9000 marche avec IrCOMM et Linux de la façon suivante :

Pour démarrer une session de communication avec `/dev/irnine`, par exemple, faites :

```
dip -t
> port irnine
> term
```

Vous utiliserez peut être `cu` à la place de `dip`. Il y a aussi du travail de fait avec l'Ericsson GF768 et le modem IR DI 27.

Benny Amorsen a écrit : Le SH888 émule le port IrDA quand vous vous connectez via un câble série, alors que beaucoup penseraient que c'est une chose plutôt obscure comme cela l'est pour moi. Mais c'est la seule manière de le faire fonctionner avec Windows, bien que je n'ai jamais réussi à le faire marcher avec Windows.

6.5 Connexion avec un appareil photo numérique

Markus Schill a écrit : "Heureusement qu'il y a d'autre personnes qui sont intéressées à faire fonctionner l'adaptateur IR SONY DSC-F1 sous Linux. Jusqu'à maintenant, je n'ai fait que jouer avec les logiciels de la suite Linux/IrDA et l'adaptateur série qui est fournit avec l'appareil photo. Voici où cela en est. J'utilise un noyau 2.0.33 et les derniers Linux/IrDA... Si je fait :

```
insmod irda
insmod irtty
irattach /dev/cua0
```

L'adaptateur commence à discuter avec l'appareil photo. `/var/log/messages` dit que le SONY-DSC-F1 a été trouvé, mais qu'aucun service n'a été démarré. (notez que ce comportement ne s'applique peut être pas aux noyaux 2.2.x).

Il y a deux programmes pour Linux qui sont disponibles et qui peuvent être utilisés avec l'appareil photo avec le câble : (1) `chotplay` et (2) `stillgrab`. Ils prennent tous les deux un tty comme ligne de commande, par conséquent, je suppose qu'ils devraient marcher si la couche irtty et la pile de protocole fonctionnent correctement... Je n'ai pas jeté de coup d'oeil au code Linux/IrDA pour l'instant ! Je ne suis pas certain de comprendre la pile, mais irtty ne devrait-il pas faire en sorte que cela ressemble à un vrai tty ? Quel service devrait être démarré ? "

Dag Brattli a écrit : "Je ne suis pas certain de savoir quel protocole l'appareil photo utilise, mais il est possible qu'il implémente les spécifications du protocole IrDA(TM) transfert d'images par infrarouge (Infrared Transfer Picture IrTran-P). Allez jeter un coup d'oeil à *http ://www.irda.org/standards/pubs/IrTran-P_10.pdf*, vous verrez que le protocole est implémenté par dessus IrCOMM (pas IrTTY !). IrTTY est quelque chose que nous utilisons juste pour communiquer avec le pilote série de Linux. "

6.6 Window$95 et Linux/IrDA

Introduction

Pourquoi cette section ? Malheureusement, les utilisateurs de Linux ne sont pas toujours capable de savoir tout sur leur configuration matérielle. Quelquefois, il est possible de trouver ces informations depuis Window$95.

Où commencer ? Sur *http ://www.microsoft.com/windows95/info/irda.htm* vous trouverez un support pour "Infrared Transfer 2.0". C'est une archive auto extractible (`W95IR.EXE`) qui fait 331Ko.

Avec certaines machines telles le HP Omnibook 800, il est nécessaire d'utiliser les spécifications du fabricant (pour cet Omnibook, cela se trouve sur votre CD de secours).

Les fichiers les plus intéressants pour trouver des détails sur la configuration sont les `..\windows\inf\*.inf` et les gestionnaires de périphériques.

D'après ce que je sais, Window$NT ne supporte pas IrDA(TM). Et pour Windows98, je n'ai pas encore entendu parler de support IrDA(TM) pour l'instant.

Il y a aussi quelques périphériques non M$ disponibles. Notez toutefois que certains utilisent des protocoles infrarouges propriétaires :

- CounterPoint : QuickBeam 1.15
- LapLink 7.5
- CarbonCopy 32 4.0
- pc ANYWHERE 7.5
- Puma Technology : TRANXIT pro 4.0

Connexion entre Linux/IrDA et Window$95 IrDA(TM)

Je pense qu'il y a `trois` façons de connecter Linux/IrDA et Window$95 :

Une *connexion réseau* entre les deux PC. Si vous avez mis en place *Infrared Transfer 2.0*, vous devriez trouver une interface réseau IrDA(TM) dans la liste des interfaces réseaux. Mais je n'ai pas encore réussi à faire une connexion.

Il est certainement aussi possible d'utiliser le *protocole IrOBEX*. Mais je ne sais pas quel logiciel utiliser et où le trouver. Je pensais que les logiciels nécessaires viendraient avec le Palm III, mais il s'avère que ce n'est pas le cas.

Takahide Higuchi *<thiguchi@pluto.dti.ne.jp>* a fournit le *support IrCOMM*. Depuis sa page *http ://www.pluto.dti.ne.jp/~thiguchi/irda/* j'ai récupéré la description suivante (que j'ai légèrement modifiée) : "Avec IrCOMM, il est possible d'envoyer et de recevoir de petits messages entre une machine Linux et un programme de terminal sur un portable sous Win95 ! Ajoutez cette ligne à votre `/etc/conf.modules` :

```
alias char-major-60 ircomm_tty
```

Ensuite, créez un fichier de périphérique `mknod /dev/irnine c 60 64`. Maintenant, les services Linux/IrDA peuvent être démarrés avec `irattach /dev/ttyS? &`. `/dev/irnine` peut être utilisé comme port série. Les modules `ircomm` et `ircomm_tty` seront chargés automatiquement par `kerneld/kmod` quand un programme utilise `/dev/irnine`. NOTE : Je pense que le programme "setserial" ne devrait pas être utilisé sur `/dev/irnine`. Quelques petits trucs :

– Pour autoriser les login via IrCOMM, utilisez cela en tant que root : Premièrement, activez IrDA et IrCOMM, Ensuite, éditez `/etc/inittab` et ajoutez une ligne comme celle là :

```
T1:23:respawn:/sbin/getty -L -w irnine 38400 vt100
```

Et faites ceci en tant que root : `init q`. Et `init` commencera à attendre les connexions IrCOMM. Vous verrez apparaître votre prompt de login préféré depuis un émulateur de terminal depuis Win95 !

– Si vous essayez `pppd`, pensez à utiliser l'option -crtscts pour désactiver le contrôle de flux. J'ai implémenté une petite émulation de contrôle de flux, mais cela n'a pas été testé.

– Maintenant, mon patch montre quel fonctionnalités sont nécessaires à l'autre interface infrarouge. Des messages de ce style seront écrits dans le syslog :
`Sep 4 10:01:02 monolith kernel: parse_control:instruction(0x12)`
`Sep 4 10:01:02 monolith kernel: data:03`

– Je voudrais spécialement connaître les messages donnés par SH888 (ou tout autre type de périphériques sauf les PC sous windows 95). Donc, s'il vous plaît envoyez moi vos syslog générés durant les connexions IrCOMM ! Si vous avez une copie des spécifications IrCOMM écrites par IrDA(TM), référez vous aux pages 34 ou 38, et vous comprendrez ce que ces messages signifient."

6.7 Connexion de Linux à Linux

6.7.1 Méthodes de connexions

Il devrait y avoir *trois* façons de connecter deux machines Linux via Linux/IrDA.

– Dag Brattli a écrit à propos du *support IrOBEX* : "Le lecteur averti se demandera ce qui peut empêcher de transférer des fichiers d'une machine Linux à une autre machine Linux. Et bien, rien !! (mais je n'ai pas encore essayé ;-)). Cela signifie que nous avons maintenant une manière "simple" de transférer des fichiers entre deux portables Linux. Je pense que cela devrait être la "fonctionnalité qui tue" que nous attendions tous !" Essayez de faire `load_misc irobex` des deux cotés, et ensuite, essayer `iroabex_app get` sur l'une des deux machines et `irobex put <file>` sur l'autre.".

– Via une connexion *réseau Linux/IrDA*. Je suppose que vous aurez à charger le module `irlan_client` sur l'une des machines et le module `irlan_server` sur l'autre.

– Avec le support *IrCOMM*, c'est à dire via une liaison série, ce qui signifierait `minicom`, `pppd`, etc.

6.7.2 Compression

Notez que cette fonctionnalité est toujours vraiment expérimentale ! Dag Brattli a écrit : "Je voulais juste vous dire que je viens juste d'ajouter de la COMPRESSION à IrLAP ! Comme vous le savez peut être, cela ne fait pas partie du standard IrDA(TM), mais Linux peut négocier avec son voisin et savoir si il est capable de compresser les données. Donc, si vous discutez avec une machine Win95, Palm III ou quoi que ce soit d'autre, vous n'aurez pas de compression !!! C'est une fonctionnalité propre à Linux d'autant que je sache ! Le standard IrDA(TM) dit que les périphériques doivent ignorer les champs inconnus dans les en-têtes de négociations, donc, nous sommes toujours compatibles avec le standard IrDA(TM) (j'ai juste emprunté une valeur non utilisée dans les en-têtes).

Si vous voulez essayer d'utiliser la compression (Linux <-> Linux) vous aurez à insérer le module `irda_deflate` quelques temps avant de faire la connexion. Personnellement, je le fait avant `irattach`.

Le standard de compression que j'ai ajouté est le format deflate utilisé par la librairie zlib qui est décrit par les RFC (Request for Comments) 1950 à 1952 dans les fichiers `ftp ://ds.internic.net/rfc/rfc1950.txt` (*format zlib*), `rfc1951.txt` (*format deflate*) et `rfc1952.txt` (*format gzip*).

L'interface de compression est similaire à celle qu'utilise PPP, par conséquent vous pouvez ajouter autant de compresseurs différents. Pour l'instant, seul GZIP est supporté, mais la compression BSD sera ajoutée plus tard."

6.8 Instances multiples

Dag Brattli a écrit : "La couche IrLAP a été améliorée de telle sorte qu'elle autorise plusieurs instances (de cette façon, je peux utiliser mon port infrarouge pour IrLAN et communiquer avec mon Pilot via un dongle IrDA en même temps)... Comment faire pour avoir plusieurs connexions Linux/IrDA ? Et bien, facile, lancez tout simplement `irattach` autant de fois que vous avez de ports... procédez comme il suit : "

```
irattach /dev/ttyS0 &    (mon dongle ESI)
irattach /dev/ttyS2 &    (mon port IrDA intégré)

insmod irlan_client
insmod irobex
```

7 Matériel supporté par Linux/IrDA

7.1 Récupérer des informations à propos des ports infrarouges des portables

Pour faire marcher le port IrDA de votre portable avec Linux/IrDA vous pouvez utiliser StandardInfraRed (SIR) ou FastInfraRed (FIR).

7.1.1 SIR

Jusqu'à 115.200 bps, le port infrarouge émule un port série tel l'UART 16550A. Ceci sera détecté par le pilote série du noyau au boot ou lorsque vous chargerez le module `serial`. Si le support infrarouge est activé dans le BIOS, pour la majorité des portables, vous aurez un message du noyau dans le style :

```
Serial driver version 4.25 with no serial options enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A     #Premier port série/dev/ttyS0
ttyS01 at 0x3000 (irq = 10) is a 16550A    #ex port infrarouge
ttyS02 at 0x0300 (irq = 3) is a 16550A     #ex port modem PCMCIA
```

7.1.2 FIR

Si vous voulez utiliser jusqu'à 4 Mbps, votre machine doit être équipée d'une puce FIR spécifique. Vous aurez besoin d'un pilote Linux/IrDA spécial pour cette puce. Par conséquent, il est nécessaire de savoir exactement quelle puce vous avez. Vous pouvez trouver ces informations de différentes manières :

1. Lisez les spécifications de votre machine, mais bon, il est très rare que vous y trouviez suffisamment d'informations.
2. Faites un `cat /proc/pci`, mais bon, les informations PCI sont souvent incomplètes. Vous trouverez les dernières informations à propos des périphériques PCI et des numéros de fabricants sur la page de Craig Hart *http ://members.hyperlink.net.au/˜chart* . Depuis les noyaux 2.1.82, vous pouvez utiliser `lspci` fourni avec les `pci-utils`. Les fichiers correspondants pour les noyaux 2.2.x sont dans `/proc/bus/pci`.
3. Si vous avez installé les logiciels Linux/IrDA, chargez le module FIR et regardez ce que donne `dmesg`, pour savoir si votre FIR est détectée ou pas.

4. Bien que je ne les utilise pas encore dans ce but, les outils `isapnp` devraient être utiles.
5. Une autre manière de trouver, expliquée par Thomas Davis : "Foncez sur le site FTP (du fabriquant), trouvez les pilotes FIR, et ils ont (pour la puce SMC) :

```
-rw-rw-r--   1 ratbert  ratbert      743 Apr  3  1997 smcirlap.inf
-rw-rw-r--   1 ratbert  ratbert    17021 Mar 24  1997 smcirlap.vxd
-rw-rw-r--   1 ratbert  ratbert     1903 Jul 18  1997 smcser.inf
-rw-rw-r--   1 ratbert  ratbert    31350 Jun  7  1997 smcser.vxd
```

   Si vous avez un doute, regardez toujours les fichiers .inf/.vxd de Win95 ; Win95 n'est livré avec **aucun** pilotes FIR (ils viennent tous d'ailleurs, pour la plupart de Counterpoint, qui a été absorbé par ESI)."
6. Utilisez l'utilitaire DOS `CTPCI330.EXE` fourni au format ZIP par le magazine informatique allemand CT *http ://www.heise.de* . Les informations fournies par ce programme sont parfois plus intéressantes que celles des programmes Linux.
7. Il y a aussi un petit utilitaire DOS fait par SMC appelé "FindChip". Le paquetage fournit aussi un utilitaire "FirSetup" qui est supposé pouvoir tout régler à part l'adresse de la puce. Référez vous à *http ://www.smsc.com/ftppub/chips/appnote/ir_utils.zip.*

   Attention : Ce paquetage n'est pas fait pour l'utilisateur final, et certains de ces utilitaires peuvent faire très mal. La seule documentation est au format M$ Word. Les utilisateurs de Linux pourront le lire avec `catdoc`, disponible à *http ://www.fe.msk.ru/~vitus/catdoc/* .
8. Utilisez le gestionnaire de périphérique de Window$95/98/NT.
9. Vous pouvez aussi vous référer à la section suivante.

7.2 Aperçu du matériel supporté

Il y a quelques pages qui parlent des périphériques infrarouges supportés par Linux sur le WWW :

- Le Linux/IrDA Project - rapport de matériel à *http ://www.cs.uit.no/~dagb/irda/hardware.html*
- Takahide Higuchi à *http ://www.pluto.dti.ne.jp/~thiguchi/ir/product.html.* Cette page est en japonais.
- J'ai aussi mis en place un rapport du matériel à *http ://www.snafu.de/~wehe/index_li.html.* Cette liste contient aussi des informations se rapportant aux périphériques qui supportent l'infrarouge et qui ne sont pas mentionnés ici (souris, imprimantes, télécommandes, transmetteurs, etc.).

  Pour que cette liste soit la plus complète possible, il est nécessaire de collecter le plus d'informations possible à propos des différents périphériques infrarouges. Vous m'aiderez beaucoup en m'envoyant un court email contenant le nom exact du périphérique et quel type de contrôleur infrarouge est utilisé.

  Tenez moi aussi au courant sur la façon dont Linux/IrDA a marché, sur quel tty, port, interruption ça a marché, et le périphérique infrarouge correspondant (imprimante, téléphone cellulaire) que vous utilisez.

  Vous pouvez aussi aider en donnant des détails technologiques à propos de certains périphériques infrarouges, ce qui est nécessaire pour développer un périphérique correspondant pour Linux.

8 Interface graphique

Si vous voulez essayer une jolie interface graphique, vous pouvez essayer `ircp.pl`. Vous aurez besoin du module Perl-GTK+ (*http ://www.gnome.org*) pour le faire marcher. Cette interface graphique est loin d'être finie, alors, n'espérez pas trop :-)

9 Économie d'énergie

Dans les spécifications de mon HP Ominbook 800, il est recommandé d'éteindre le port infrarouge si il n'est pas utilisé parce qu'il peut consommer jusqu'à 10 % de la batterie.

Si nécessaire, vous pouvez aussi essayer de désactiver la fonctionnalité `Fast RRs` dans la section IrDA du noyau. Cette option vous donnera de bien meilleurs temps d'attente, mais consommera bien plus de puissance.

10 Résolution des problèmes, liste de diffusions

10.1 Informations générales

Si vous rencontrez des problèmes, essayez ce qui suit :

- Lisez la FAQ ci dessous.
- Allez faire un tour dans `/var/log/messages` et/ou `/var/log/kern`.
- Tapez `dmesg`.
- Allez voir les différents fichiers dans `/proc/irda`.
- Allez faire un tour sur les *archives de la liste de diffusion* à *http ://www.ita.chalmers.se/˜svinto/hypermail/irda/*, pour savoir si votre problème est déjà recensé.
- Posez votre question sur la *liste de diffusion Linux/IrDA*. Pour vous abonner à la liste, envoyez un email à `linux-irda-request@list.uit.no` avec "subscribe" (s'inscrire) dans le corps du message. Utilisez l'adresse `linux-irda@list.uit.no` pour poster un message. Vous êtes le bienvenu sur cette liste pour poser des questions, répondre aux questions, faire des rapports de bugs, envoyer des patchs, faire des suggestions et des commentaires. Pour éviter le spam, la liste est dorénavant modérée, par conséquent, il se peut qu'un certain temps se passe avant que votre message soit distribué aux autres membres de la liste.

10.2 Technique de résolution des problèmes

Bien que je ne soit pas franchement un hacker, j'ai récupéré quelques erreurs possibles et quelques bugs de la suite Linux/IrDA.

- Vous pouvez régler le niveau de debug dans `/proc/sys/net/irda/debug` à 1, 2, 3 ou 4.
- Utilisez les fichiers dans `/proc/sys/net/irda` pour essayer différents paramètres comme `echo 0 > /proc/sys/net/irda/discovery`.
  Les différents fichiers présents dans `/proc/*/irda` sont :

```
root@duckman:~# ls /proc/sys/net/irda/* /proc/net/irda/*
/proc/net/irda/discovery      /proc/net/irda/irlmp             /proc/sys/net/irda/devname
/proc/net/irda/irda_device    /proc/net/irda/irttp             /proc/sys/net/irda/discovery
/proc/net/irda/irias          /proc/sys/net/irda/compression
/proc/net/irda/irlap          /proc/sys/net/irda/debug
```

- Il est aussi possible de débogger le code. Mais je ne sais pas le faire. Si vous voulez utiliser le code de débogage SKB, vous pouvez éditer `irda.h` et changer `/include/linux/skbuff.h` (référez vous à l'historique des révisions du code datant du 2/10/98).
- Pour les problèmes avec le module `irda`, un module provenant du *paquetage de modules* `kdstat` pourra être utile. Mais je n'ai pas été en mesure de l'essayer.
- "Vous pouvez dorénavant changer le nombre de paquets de découverte utilisés (1, 6, 8 ou 16) et le timeout entre deux émissions (2-8 * 10 ms) depuis `/proc/sys/net/irda`. Faites des tests si vous avez des problèmes à faire reconnaître votre périphérique. Mon Palm III semble préférer 16 `discovery_slots` et 8 (*10 ms) pour `slot_timeout`." ... "Le minimum absolu pour une découverte du IR-610 semble être 9."
- Si l'un d'entre vous récupère un Oops du noyau, alors envoyez le dans le programme `../linux/scripts/ksymoops/ksymoops`, pour que nous puissions trouver d'où vient le problème. Copiez juste l'Oops depuis votre syslog et coller le dans un fichier, ensuite, lancez `ksymoops <fichier>`.

– Dag Brattli a écrit : "J'ai découvert que la carte son cs4232 me faisait plusieurs centaines d'interruptions par secondes ! Je l'ai supprimé du noyau, et la machine est approximativement 4 fois plus rapide !
Linux/IrDA peut rencontrer des problèmes si vous exécutez le serveur esound (esd) sur votre machine. Aucune de mes deux machines, un portable P166 et un PPro 200, ne peut exécuter Linux/IrDA quand esd est lancé. La raison est que esd fait faire plus de 300 interruptions par seconde à la carte son, ce qui fait que le pilote série perd des données quand il en reçoit. Cela est du au fait que le pilote série utilise maintenant des interruptions lentes dans le noyau 2.2 (tout utilise des interruptions lentes dans le 2.2), par conséquent le gestionnaire d'interruptions s'affole un peu. La bonne chose à propos des interruptions lentes est que les paquets sont délivrés de manière plus rapide, puisque vous n'avez pas à attendre le tick suivant. La seule exception pour cela est le pilote pc87108 qui marche toujours bien car il utilise le DMA et ne fera que deux interruptions par paquets.

11 Bugs connus

Si vous trouvez un bug, s'il vous plaît, envoyez un rapport à la liste, incluant la sortie de `dmesg`, la version du noyau, et le matériel que vous utilisez. Merci !

Quelques fois, IrCOMM n'arrive pas à se connecter (spécialement quand les deux périphériques se découvrent mutuellement. Vous pouvez désactiver la découverte avec `echo 0 >/proc/sys/net/irda/discovery`).

Un caractère CR (retour à la ligne) ne peut pas être transféré entre deux machines Linux via IrCOMM avec `cat file >/dev/irnine` et `cat /dev/irnine`. Cela fait des drôles de choses et bloque votre machine Linux.

La compilation du pilote pc87108 dans le noyau le fait planter au boot. Solution temporaire : le compiler en tant que module.

IrOBEX peut détruire des données en réception. Ce bug est certainement dans la partie utilisateur de IrOBEX.

12 FAQ

– Q1 - Question : Je ne sais pas du tout quels sont les ports utilisés et les irq. Que faire ?
– Réponse :
Premièrement : Réglages matériels
- 1 Allez regarder les spécifications de votre matériel. Si ce n'est pas disponible, référez vous à la page de support de votre constructeur, ou contactez son support technique. Vous pourrez peut être trouver les informations dans les aperçus du matériel mentionné plus haut.
- 2 Utilisez un BIOS à jour. D'habitude sur la page du support technique de votre fabricant.
- 3 Essayez de faire `setserial /dev/ttyS? -g -a | egrep 16550A`. L'un des périphériques mentionnés est probablement celui que vous cherchez. D'habitude, c'est le deuxième, mais ce n'est pas toujours le cas.
- 4 Note : Ce qui semble être un UART est physiquement un contrôleur IrDA, Pour mon HP Omnibook 800, c'est le contrôleur VLSI VL82C147 PCI - IrDA. Ces contrôleurs semblent pouvoir se connecter jusqu'à 115 200 bps comme les UART 16550A. Mais c'est souvent difficile de trouver la bonne configuration.
Deuxièmement : Comment donner au noyau les paramètres nécessaires pour votre matériel ?
- 4 `cat /proc/ioports` pour voir quels ports sont déjà utilisés.
- 5 `cat /proc/interrupts` pour voir quelles interruptions sont déjà prises.
- 6 Libérez les ports et interruptions pour le périphérique IR, par exemple : arrêtez le service PCMCIA ou incluez cette ligne dans `/etc/sysconfig/pcmcia` : `PCIC_OPTS="irq_list=3,4,5,7,9,10,12,14,15"`
- 7 Maintenant, essayez de deviner quel port et quelle interruption est utilisée. Utilisez un `setserial /dev/ttySx irq M port 0xNNNN` pour le dire au noyau. Si il y a plusieurs possibilités, essayez les toutes

(Note : comme il est dit dans le *Serial-HOWTO*, vous ne devriez pas essayer les interruptions 0, 1, 6, 8, 13 et 14).
- 8 Si vous avez réussi, envoyez les paramètres utiles à l'auteur, car il aimerait bien les inclure dans son aperçu du matériel.
- 9 Bonne chance.
Il sera peut être aussi nécessaire d'affiner les réglages du port série infrarouge avec setserial, ex : `setserial /dev/ttyS0 spd_vhi` (vitesse 115200).

- Q2 - Question : `irattach` se fige, mais reconnait mon imprimante. `/var/log/messages` montre que irattach a trouvé mon HP LaserJet 6P.
- Réponse : Le "figement" est normal pour irattach. Tout marche bien si vous voyez l'imprimante dans les logs. Le "figement" signifie que irattach est en attente de connexion entrante. Si vous le tuez avec un <CTRL C> irattach plante et `/dev/ttySx` ne marche plus du tout. Le problème est dans le module irda, et pas dans le programme irattach. La seule chose à faire est de rebooter ! La prochaine fois mettez irattach en tâche de fond en utilisant `irattach &`. Stoppez le si nécessaire avec un `killall irattach`. Recommandation d'Andreas Butz : "D'après ce que je sais, <CTRL Z> devrait marcher aussi, mais je ne l'ai pas testé dans ce cas précis. Normalement, cela a exactement le même effet que de rajouter un `&` à la commande.
- Q3 - Question : Je reçois un message dans le style "tcsetattr read/write error" dans `/var/log/messages`.
- Réponse : Probablement causé par un mauvais `/dev/ttyS*`, une mauvaise interruption ou un mauvais port.
- Q4 - Question : tout semble marcher car j'ai les bons messages. Mais ça ne marche toujours pas.
- Réponse : Placez les périphériques à une distance de moins de 0.5 mètre (1.5 pied). Vérifiez qu'une seule application utilise le port infrarouge. Vérifiez que les deux interfaces utilisent le même protocole comme IrOBEX ou IrCOMM.
- Q5 - Question : J'ai téléchargé la dernière version des pilotes, et ça a très bien compilé sous Linux 2.0.33 sur mon IBM Thinkpad 560E. En l'absence d'autre machine IrDA pour le tester, est-ce qu'il est possible d'affirmer qu'une fois que le module a été inséré et que syslog rapporte "irattach : Serial connection established.", que l'IR marche vraiment, et se mettra-t-il à répondre quand il aura une autre machine pour discuter ?
- Réponse de Dag Brattli : Désolé, cela signifie juste que irattach a fait son boulot, qui est juste de démarrer irda-tty. Peut être que le message aurait du être différent, mais comme je l'ai dit, il signifie que la connexion série entre la puce irda et le pilote irda est établie.
  Note : le support des noyaux 2.0 s'arrête. Vous êtes vivement encouragés à passer aux versions 2.2 et à utiliser les derniers patchs IrDA disponibles à *http ://www.cs.uit.no/˜dagb/irda/snapshots/*.
- Q6 - Question : Au démarrage `modprobe -a` vérifie `/lib/modules/<uname -r>/net/irda.o` et donne le message : "IrLAP ; Missing IrTTY /IrLMP Error no IrLAP connection" (dans `/var/log/messages` et sur la console).
- Réponse de Werner Heuser : Truc pour les système style SYSTEM V : Mettez un script par exemple "ir_rmmod" contenant :

```
#!/bin/sh
echo "$0 : remove irda module"
rmmod irport.o
rmmod irtty.o
rmmod irda.o
```

  dans le processus de démarrage (`/etc/init.d` et un lien symbolique par exemple "S100ir_rmmod" dans `/etc/rc3.d` vers "ir_rmmod"). (Vérifiez le chemin vers "sh"). pour les systèmes style BSD, essayez l'approche correspondante.
- Q7 - Question de Ho Chin Keong : Y a-t-il une autre manière de faire communiquer deux machines sans créer un réseau entre les deux ?
- Réponse de Dag Brattli : Oui et non ! L'un des standards IrDA, IrCOMM permet d'émuler un câble

série entre deux machines, donc, vous pouvez utiliser n'importe quel programme écrit pour les ports série (terminaux, PPP, slip, etc.). Cela n'est, par ailleurs, pas encore implémenté dans Linux/IrDA. Le support IrLPT (imprimante) utilise un sous ensemble de IrCOMM, donc des morceaux fonctionnent !

- Q8 - Question de Ho Chin Keong : Si je bloque délibérément le faisceau infrarouge pendant plus de 10 secondes, la connexion ne peut être rétablie. Je dois tuer irattach et redémarrer toute la procédure pour créer la connexion infrarouge. La connexion peut toutefois être maintenue, si la coupure dure moins de 10 secondes. Est ce que cela fait partie de l'implémentation, ou est-ce un bug ? Y a-t-il un moyen d'augmenter cette limite de 10 secondes à plus, voir même à l'infini ?
- Réponse de Thomas Davis : Cela semble être un bug dans la partie principale du code de IrLAP/IrLMP. Il semble que la notification de remise à zéro/déconnexion ne remonte pas jusqu'en haut de la pile. On peut remarquer quand IrLPT se fige dans le mode query quand vous essayez de discuter avec une imprimante, et se déconnecte/interrompt quand il établit la connexion. (et maintenant, cela arrive aussi dans la partie IrLAN)
- Q9 - Pierre-Guillaume Raverdy a demandé : Dois-je mettre à jour mon Palm Pilot, et mettre le système 3.0.2 ?
- Réponse de Dag Brattli : Il n'est pas nécessaire de mettre à jour votre Pilot, mais cela ne peut pas faire de mal. C'est par contre nécessaire si vous voulez utiliser la librairie IrCOMM de IsComplete.
- Q10 - Pierre-Guillaume Raverdy a demandé : De plus, un exemple simple de code source (spécialement du coté du Palm Pilot) serait très pratique.
- Réponse de Dag Brattli : Récupérez le Pilot SDK de Palm. Décompressez le fichier `examples.zip` et regardez le programme beamer.
- Q11 - Kai Poehlmann a écrit : J'ai un téléphone GSM Ericsson SH 888 et une machine Linux sans port irda. J'ai entendu dire qu'il était possible de discuter avec le protocole IrDA du téléphone en utilisant un câble série... :-/ Est-il maintenant possible d'utiliser ce téléphone avec Linux/IrDA et un câble série ?
- Réponse de Matthias Lemke : Oui, c'est possible. J'ai essayé il y a 6 semaines. Mais j'ai eu le même problème qu'avec juste l'IrDA. Je peux me connecter au téléphone, le remettre à zéro avec ATZ, mais après ATDT, plus rien ne se passe, que ce soit avec ou sans le câble. Mais je pense que vous devriez ressayer avec la dernière version de Linux/IrDA.
- Q12 - Jonah Peskin a écrit : Est ce que quelqu'un a réussi à augmenter les capacités d'un émetteur IrDA ? J'utilise un portable Dell Inspiron 7000, et il semble que son rayon d'émission soit d'environ 1 mètre. Je sais que cela rentre dans les spécifications de l'IrDA, mais y a-t-il des portables ou des périphériques qui permettent de faire plus d'un mètre ? Pourquoi ne pas prendre un câble d'un mètre ?
- Réponse de tzeruch@ceddec.com : Parce qu'une connexion manuelle (à part le réceptacle pour le Palm) est difficile et prend du temps et crée souvent des problèmes (accéder à l'arrière de mon ordinateur de bureau nécessite quelques contorsions).

  Le problème principal est qu'il est nécessaire de rendre le récepteur plus sensible. Dans les bases de la physique on apprend la loi du carré inverse : L'intensité diminue proportionnellement au carré de la distance, donc, passer d'un mètre à 5 demande 25 fois plus de puissance (et la surcharge courant sur un ordinateur portable), ou 25 fois plus sensible (et la distance dynamique - il doit toujours être possible de recevoir à 10 centimètres). Et si vous voulez le faire de l'autre côté, il ne doit pas être simplement 25 fois plus sensible, il doit pouvoir discerner le faible faisceau infrarouge des néons, économiseurs d'écrans et autres ombres en mouvement...

  Quelqu'un a essayé avec une mise à jour du Palm III : *http ://home.t-online.de/home/PSPilot/ppppiii.htm*

13 Contrôle à distance via infrarouge

13.1 Ressources

La prise de contrôle à distance par infrarouge n'est pas le but du projet Linux/IrDA mais est incluse dans ce HowTo pour couvrir de manière exhaustive "Linux et l'infrarouge". J'ai trouvé *trois projets* qui oeuvraient dans cette direction. Vous pouvez trouver des liens vers des informations plus récentes à *http ://www.snafu.de/˜wehe/index_li.html*.

- LIRC LInux Remote Control LIRC est un paquetage qui supporte la réception et l'émission de signaux infrarouges des protocoles de contrôle à distance IR les plus courants. Il contient un pilote pour les périphériques connectés aux ports séries, un démon qui décode et encode les signaux IR en utilisant ce pilote, un démon de souris qui traduit les signaux IR en mouvements de souris et quelques programmes utilisateurs qui vous permettent de contrôler votre ordinateur avec une télécommande. Takahide Higuchi a écrit à propos de LIRC : "C'est génial, et j'ai l'impression que c'est un projet très complet, mais j'ai aussi l'impression que quasiment aucun des périphériques du marché n'est supporté (vous aurez besoin d'un fer à souder et de circuits spéciaux... un peu dur pour le néophyte). Je suppose que LIRC deviendra plus populaire si le support infrarouge est implémenté dans les pilotes FastIR et qu'une API commune (par exemple un IrSocket brut et des ioctls communs) est disponible !". Vous trouverez LIRC à *http ://www.thp.uni-koeln.de/˜rjkm/lirc/*.
- Télécommande infrarouge série Voici un périphérique simple et bon marché qui peut être connecté à tout port série pour contrôler presque tous les appareils qui possèdent une télécommande IR. Il a été conçu et fabriqué sur une carte d'extension à sertir et est à présent disponible sous forme d'une carte ISA. Vous trouverez de plus amples informations à *http ://www.armory.com/˜spcecdt/remote/remote.html*.
- Outils infrarouges pour le COREL Netwinder PC Ryan Shillington a écrit quelques outils pour contrôler le COREL NetWinder par infrarouge, par exemple :
  Serveur pour l'administration Corel Palm. Il nécessite simplement d'avoir le port infrarouge installé et en état de marche. Avec lui, vous pouvez changer l'adresse IP, l'adresse de la passerelle, configurer eth1, etc. Vous pouvez aussi lancer quelques commandes simples ET vérifier la température, mémoire, charge de la machine.
  Client pour l'administration Corel Palm. Vous pouvez aussi lancer quelques commandes simples ET vérifier la température, mémoire, charge de la machine.
  Un pilote IR très simple. Il ne supporte pas IrDA (seulement des transferts non fiables). Il ne cherche que des signaux venant de télécommandes (et claviers, etc.). Il bloque et passe les données d'une façon très différente.
  Vous trouverez tout ceci à *http ://www.netwinder.org/˜ryansh/*

13.2 Commande a distance IR - IrDA

Deux des projets décrits plus haut utilisent une sorte de dongle fait à la main pour la commande à distance. Il y a aussi une description sur comment construire un dongle IrDA vous même dans l'ELEKTOR allemand de mai 1997 p28. Peut être que quelqu'un pourrait transformer ces dongles en un seul.

Pour une discussion sur les relations entre les télécommandes IR et IrDA, j'ai fait une compilation de la liste Linux/IrDA (raccourcie et modifiée par wh) :

Ryan Shillington a écrit : "Les télécommandes IR et ASK-IR sont très différentes de FIR, MIR ou SIR.

Les télécommandes IR et ASK-IR ont des vitesses très basses et des fréquences très basses (mais de longues portées). Elles opèrent à environ 2400bps.

SIR opère à des vitesses supérieures, et est voué à des transmissions longues portées où vous avez besoin que plus d'un ou deux caractères passent dans la liaison (pas comme les télécommandes).

MIR est légèrement plus rapide (mais a moins de portée), mais sa vitesse peut monter jusqu'à 1.15Mbps et FIR (ou les périphériques doivent presque se toucher) a un débit de 4Mbps. La portée est inversement proportionelle à la vitesse de transmission.

Je travaille sur des pilotes pour les télécommandes infrarouges, mais vous devriez déjà le savoir si votre matériel IR le supporte. Cherchez des protocoles tels NEC, RC-5 ou RC-0 (ce sont les plus communs).

Vous pouvez utiliser SIR pour recevoir des signaux de télécommandes. Réglez bien votre débit au plus bas et les données commenceront à affluer. MAIS d'après mon expérience, ce ne sont pas les BONNES données. Ce n'est pas analysé de la bonne façon, et par conséquent, vous ne pouvez ni calculer la somme de contrôle ni la vérifier avec son complément.

J'ai réussi à récupérer des données (en utilisant SIR) avec des télécommandes. On m'a dit que SIR lisait les données venant d'une télécommande différemment en fonction de la température (malgré tout, je n'en ai jamais fait l'expérience)."

Lichen Wang <lwang1@ix.netcom.com> a écrit en réponse : "Le truc que l'on appelle ASKIR dans tant de portables n'est pas fait pour les télécommandes infrarouges. ASKIR est fait pour le Sharp Wizard, les PDA Zauaus et quelques portables Sharp. Sharp avait créé cela bien avant l'arrivée d'IrDA et le supporte toujours pour maintenir une compatibilité. Le Newton d'Apple avait aussi cette capacité là à un moment.

Brièvement, ASKIR utilise un format de transmission asynchrone à 9.6 Kbps (19.2 et 38.4 Kbps sont aussi possibles) avec 8 bits de données, 1 bit d'arrêt et une parité impaire. Le bit de démarrage ainsi que tous les bits 0 sont transmis comme une onde carrée de 500 KHz (sous porteuse DASK). Le bit "arrêt" ainsi que tous les bits 1 sont représentés par l'absence de transmission.

Comme vous pouvez le constater, c'est totalement incompatible avec les télécommandes IR.

[..]

Vrai. Non seulement il est possible d'utiliser le matériel SIR pour *recevoir*, mais vous pouvez aussi transmettre. Bien entendu, il y a quelques limitations.

La majorité des télécommandes utilisent une sous porteuse à 38KHz. 3 fois 38 ça fait 114, très proche de 115.2. Il est donc possible de régler l'UART pour opérer à 115.2 Kbps, 7 bits de données, 1 bit d'arrêt et pas de parité - soit 9 bits. Tous les 3 cycles de la sous porteuse à 38 KHz peuvent être *reçus* ou *émis* en tant qu'un octet Ox5B.

Il y a quelques limitations physiques en plus du fait que la sous porteuse doive être de 38KHz. Le *récepteur* SIR n'est pas aussi sensible au 38KHz que la télécommande. Le *transmetteur* SIR a un cycle d'utilisation beaucoup plus faible, et par conséquent, ne peut émettre une trop forte sous porteuse.

La télécommande encode le signal de contrôle en ajoutant un patron spécifique à la sous porteuse. Maintenant que vous pouvez *émettre* et *recevoir* la sous porteuse, tout ce qu'il reste à faire, c'est de tout synchroniser.

Pour *émettre*, il est nécessaire de savoir combien d'octets de 0x5B consécutifs il sera nécessaire d'envoyer pour chaque coup de sous porteuse, et combien de temps rester silencieux pendant les émissions.

Pour *recevoir*, il sera nécessaire de savoir combien d'octets 0x5B vous recevez consécutivement, et quelle est la durée entre les trous d'octets consécutifs.

[..]

Mon expérience avec les distances de transmissions IrDA de SIR, MIR et FIR sont légèrement différentes de ce que Ryan a dit.

[..]

SIR, MIR et FIR devraient toujours marcher de 0 à 100 cm, mais dans la pratique :

(a) Certains périphériques ont des problèmes avec les *LONGUES* distances.

Lorsque cela est possible, placez les périphériques à moins de 50 cm l'un de l'autre. Les périphériques ayant peu de puissance, comme les pagers et les téléphones cellulaires, etc. auront peut être des distances encore plus petites bien qu'ils utilisent SIR à la place de MIR ou FIR.

(b) Certains périphériques ont des problèmes avec les *COURTES* distances.

Placez les deux périphériques à au moins quelques centimètres l'un de l'autre. Les mettre plus près peut causer des problèmes.

Il est un peu intuitif de rapprocher les deux périphériques lorsque la connexion n'est pas bien stable. Mais cela va à l'encontre de toute intuition que trop près n'est pas bon non plus. La raison est que l'intensité lumineuse à 1 cm est 10 000 fois plus forte qu'à 100 cm. A 0.5 cm, 40 000 fois, etc. Les constructeurs de récepteur IR ont du mal à gérer cette distance dynamique. Nous avons tous du mal à lire avec une lampe de 10 W, imaginez comment ça serait avec une lampe de 100 000 W !

[..]

La couche physique IrDA est totalement incompatible avec la modulation DASK utilisée dans les télécommandes IR. Par conséquent, il n'est pas possible d'utiliser la même fonction de contrôle avec FIR et les télécommandes. Toutefois, presque tous les contrôleurs FIR incluent des fonctions additionnelles pour supporter les télécommandes IR. National, SMC, et Winbond (pour ne citer que ceux là) ont tous de telles puces.

Les transmetteurs IR pour FIR et les télécommandes sont très semblables. J'ai essayé avec un transmetteur FIR standard. Il est capable d'atteindre 10 mètres dans un but de télécommande. Par conséquent, il est aussi performant que les transmetteurs inclus dans les télécommandes.

Les récepteurs IR pour FIR et les télécommandes sont par contre assez différents. Un récepteur FIR peut recevoir un signal de télécommande, mais ceci seulement à 1 mètre alors que les récepteurs intégrés dans les télécommandes atteignent 10 mètres.

J'ai un adaptateur ISA avec une puce National qui supporte aussi bien FIR que les télécommandes. J'ai aussi des dongles IR qui incluent les deux types de récepteurs (ainsi qu'un transmetteur pour les deux modes). Je n'ai pour l'instant trouvé aucun logiciel supportant les fonctions de télécommandes. J'ai fait mes propres expériences sous DOS (je ne peux pas encore installer Linux). Quelqu'un est intéressé ? "

Benny Amorsen a écrit : "J'ai un portable qui est supposé supporter ASKIR. Le mode du port infrarouge peut être changé en ASKIR dans le BIOS. Avoir à rebooter pour changer le mode dans le BIOS le rend inutile, donc, quelqu'un pourrait essayer de trouver un moyen de le changer à la volée. "

Dag Brattli a écrit : "Il devrait être possible d'utiliser IrControl (auparavant IrBus) pour utiliser les télécommandes compatibles IrDA. Je ne connais aucune télécommande utilisant le standard IrControl, mais il devrait en exister quelques unes (quelqu'un en sait plus ?). Vous devriez aller jeter un coup d'oeil au site IrDA (*http ://www.irda.org*) et récupérer le standard de la couche physique (qui inclue IrControl je pense).

IrDA "Normal" (en utilisant IrLAP) n'est **pas** une bonne chose pour l'utilisation avec les télécommandes à cause de la trop grande différence de nature (et il ne supporte que le 9600 bps pour les travaux hors connexions). La raison de cette limite de portée est la protection des yeux qu'ils disent (mais je ne sais pas pourquoi CIR marche mieux en utilisant la même puissance). J'ai par ailleurs vu des portables connectés avec 4 à 5 mètres entre eux (mais je ne pense pas qu'une grande vitesse de connexion soit possible).

La majorité des chipset IrDA marchent en mode CIR, et il est assez facile de modifier les pilotes pour qu'ils parlent CIR. Takahide Higuchi a commencé à regarder IrSockets et il serait assez bien si il était possible d'ouvrir une socket Ir(DA) brute qui permettrait d'envoyer et de recevoir des paquets CIR. Ensuite, toutes les applications CIR pourraient rester dans l'espace utilisateur.

Je sais que CIR intéresse Corel dans le but de contrôler le NetWinder (et ils ont en ce moment du code qui marche). Allez jeter un coup d'oeil à *http ://www.slashdot.org/articles/98/12/05/0916216.shtml* ou *http ://www.netwinder.org/˜ryansh.*

14 Infrarouge et protection des yeux.

Cette section reprend quelques idées et pensées qui ont été échangées sur la liste Linux/IrDA. Rien ici n'est fondé sur des connaissances médicales, par conséquent toute personne qui aurait des informations plus complètes et plus sures est encouragée à contribuer à ce HowTo.

Les spécifications d'IrDA disent que la portée des périphériques IrDA a été limitée à 1 mètre dans le but de protéger les yeux des utilisateurs. Une autre explication possible serait que la consommation ainsi que la pollution IR soient aussi la cause de cette limitation. En principe, il devrait y avoir un danger pour l'oeil, car les IR ne sont pas détectés par l'oeil, et donc, la pupille ne se ferme pas pour protéger la rétine des sources IR fortes. C'est la même situation avec les UV, qui causent un aveuglement nébuleux, mais contrairement aux UV, les IR contiennent beaucoup moins de puissance destructrice à cause de leurs longueurs d'ondes plus grandes.

Les seuls restrictions légales et conseils médicaux que nous ayons pu trouver sur le web étaient les émissions infrarouges des lampes à bronzer ou dans les processus de soudage. Cela suggère que la lumière IR émise par les périphériques IrDA serait sans aucun danger, vu que même le pic de puissance des plus puissantes LED IR est d'environ 300mW, ce qui n'est rien comparé aux lampes a chauffer à usage médical qui peuvent monter jusqu'à 500W. Pour celles ci, il est nécessaire de porter des lunettes protectrices, alors, il se peut que regarder en face 1 000 LED IR en une seule fois ait le même effet. Les effets de la lumière IR sont majoritairement la chaleur, pas une altération ou une destruction des structures cellulaires comme le font les UV. Toutefois, dans les spécifications du HP OmniBook 800, Hewlett-Packard recommande de ne pas regarder directement dans les LED IR.

Comme je l'ai dit plus haut, cette discussion n'est basée que sur des présomptions et des déductions faites sur les données trouvées dans les spécifications des LED IR et des lampes chauffantes. Si quelqu'un a des connaissances médicales à partager, faites le !!!

15 Crédits

Merci à :

– Dag Brattli - au coeur de Linux/IrDA
– Thomas Davis - au coeur de Linux/IrDA
– Takahide Higuchi - au coeur de Linux/IrDA
– Ralf Zabka
– Benny Amorsen
– Lichen Wang
– Ryan Shillington
– Richard Titmuss
– Fons Botman
– Rui Oliveira
– Jon Howell
– Carlos Vidal
– Joonas Lehtinen
– Markus Schill
– Bjoern Hansson
– Pawel Machek
– Ho Chin Keong

– Bjoern Mork
– Andreas Butz
– Tang Ning
– Les membres de la liste Linux/IrDA.
– Les écrivains d'autres HowTo qui m'ont donné tout plein d'inspiration.
– Les développeurs des SGML-Tools qui m'ont permis d'écrire ce HowTo.

Désolé, je n'ai pas commencé à faire les crédits quand j'ai démarré ce HowTo, j'ai certainement oublié quelqu'un.

16 Historique des versions

– v0.1 à v0.4a, 19 mars 1998 au 4 août 1998, brouillons, non inclus dans le LDP.
– v1.0, 14 août 1998, ajout au LDP.
– v1.1, 18 août 1998, ajout des informations à propos du patch IrCOMM de Takahide Higuchi, changements mineurs.
– v1.2, 24 août 1998, mise à jour pour les pilotes `linux-irda-1998-08-20`, ajout de la section FIR et de cette section, changements mineurs.
– v1.3, 27 septembre 1998, ajout des sections sur les instances multiples, téléphones cellulaires, appareils photo numériques, connexion Linux à Linux, le superbe CVS, économie d'énergie, quelques changements dans la configuration système générale, changements dans l'aperçu du matériel, changements mineurs.
– v1.4, 11 octobre 1998, meilleure description du support IrCOMM, changement de la section sur les dongles, changements dans la section Palm III, changements mineurs.
– v1.5, 12 octobre 1998, changements mineurs.
– v1.6, 26 octobre 1998, ajout de la section à propos de IrManager, mis à jour pour les pilotes `linux-irda-1998-10-21`, changements dans la section dongle, changements mineurs.
– v1.7, 1 novembre 1998, ajout de la section contrôle à distance, changements dans la section dongles, changements mineurs.
– v2.0, 9 janvier 1999, réécriture quasi complète et un réarrangement en accord avec la nouvelle structure de Linux/Ir qui est maintenant inclus dans le noyau depuis la version 2.1.131, ajout des informations à propos du support BIOS dans la section dongle, ajout de la section outils de configuration et suppressions de la section CVS.
– v2.1, 13 janvier 1999, changements mineurs.
– v2.2, 26 janvier 1999, changement du nom du projet de Linux/IR en Linux/IrDA, extension de la section résolution des problèmes, changement dans l'ordre du chapitre des bugs connus, suppression de quelques coquilles.
– v2.3, 4 février 1999, ajout de la section Infrarouge et protection des yeux écrite par Andreas Butz, correcteur d'orthographe..., remaniements dans la section paramètrage du noyau, ajout de quelques informations par Andreas Butz, changements mineurs.
– v2.4, 9 février 1999, changement des informations sur comment appliquer un patch.

17 Copyright et dénégation

Les informations contenues dans ce document sont exactes d'après ce que je sais, mais il y a toujours une chance pour que j'aie commis quelques erreurs, donc, ne suivez pas toutes les manipulations trop aveuglément, spécialement si cela vous semble faux. Rien ici ne devrait pouvoir causer de dommages à votre ordinateur, mais juste au cas où, les auteurs ne sont responsables d'aucun dommage encouru du fait d'actions effectuées en se basant sur les informations comprises dans ce document.