

INSTALLATION DES DRIVERS

**CARTES MULTIVOIES PCI
XRSUNI, XRSUNI-LP ET XRSPCI**

DRIVER

MANUEL D'UTILISATION

COPYRIGHT (©) ACKSYS 2001-2017

Ce document contient des informations qui sont protégées par Copyright.

Tout ou partie du présent document ne pourra être reproduit, transcrit, stocké dans n'importe quel système informatique ou autre, traduit dans n'importe quelle langue et n'importe quel langage informatique sans le consentement préalable et écrit de ACKSYS, ZA Val Joyeux, 10, rue des Entrepreneurs, 78450 VILLEPREUX, FRANCE.

MARQUES DEPOSEES ®

- *ACKSYS* est une marque déposée de *ACKSYS*.
- Windows 10, Windows Seven, Windows Vista, Windows XP, Windows 2000, Windows NT, Windows 95, Windows 98 et Windows ME sont des marques déposées de Microsoft.
- Linux est une marque déposée de M. Linus Torvalds.

NOTICE

ACKSYS ® ne garantit en aucune façon le contenu du présent document et dégage son entière responsabilité quant à la rentabilité et la conformité du matériel aux besoins de l'utilisateur.

ACKSYS ® ne pourra en aucun cas être tenu pour responsable des erreurs éventuellement contenues dans ce document, ni des dommages quelle qu'en soit l'importance, du fait de la fourniture, du fonctionnement ou de l'utilisation du matériel.

ACKSYS ® se réserve le droit de réviser périodiquement ce document, ou d'en changer le contenu, sans aucune obligation pour ACKSYS ® d'en aviser qui que ce soit.

TABLE DES MATIERES

1.	INTRODUCTION	1
2.	INSTALLATION DU DRIVER ACKSYS POUR WINDOWS 98 & MILLENNIUM.....	3
3.	INSTALLATION DU DRIVER ACKSYS POUR WINDOWS NT 4.0	7
4.	INSTALLATION DU DRIVER ACKSYS DE WINDOWS 2000 A WINDOWS 10.....	9
5.	INSTALLATION DU DRIVER ACKSYS POUR LINUX (NOYAU 2.2.X).....	14
6.	INSTALLATION DU DRIVER ACKSYS POUR LINUX (NOYAU 2.4.X).....	15
7.	INSTALLATION DU DRIVER ACKSYS POUR NOYAUX LINUX 2.6.X.....	18
8.	ANNEXE A RETOURNEMENT DE LIGNE.....	23
9.	ANNEXE B PROBLEMES RENCONTRES	24

1. INTRODUCTION

Ce manuel décrit l'installation des drivers Windows et LINUX des cartes suivantes :

- xRSUNI-232 : Cartes multivoies série RS232 2, 4 et 8 voies, bus PCI 5V et 3V.
- xRSUNI-400 : Cartes multivoies série RS422/485 2, 4 et 8 voies, bus PCI 5V et 3V.
- xRSUNILP-232 : Cartes multivoies série RS232 4 et 8 voies, bus PCI 5V et 3V, Low profile
- xRSPCI-232 : Cartes multivoies série RS232 2, 4 et 8 voies, bus PCI 5V.
- xRSPCI-400 : Cartes multivoies série RS422/485 2, 4 et 8 voies, bus PCI 5V.

2. INSTALLATION DU DRIVER ACKSYS POUR WINDOWS 98 & MILLENNIUM

Installez la carte dans le PC puis démarrez Windows.

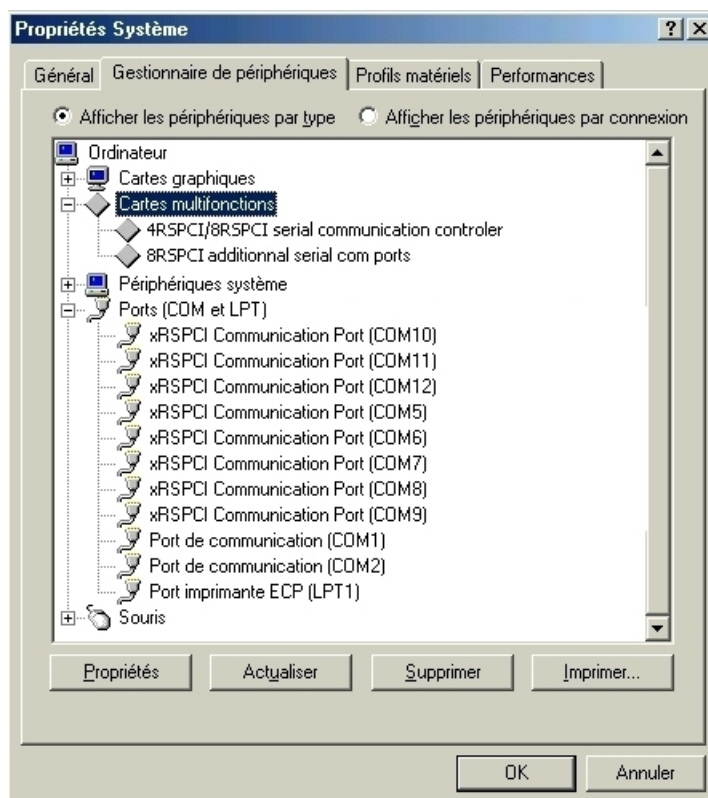
Comme il s'agit d'une carte PCI, une fois que la carte a été mise ou retirée, la configuration sera automatiquement mise à jour.

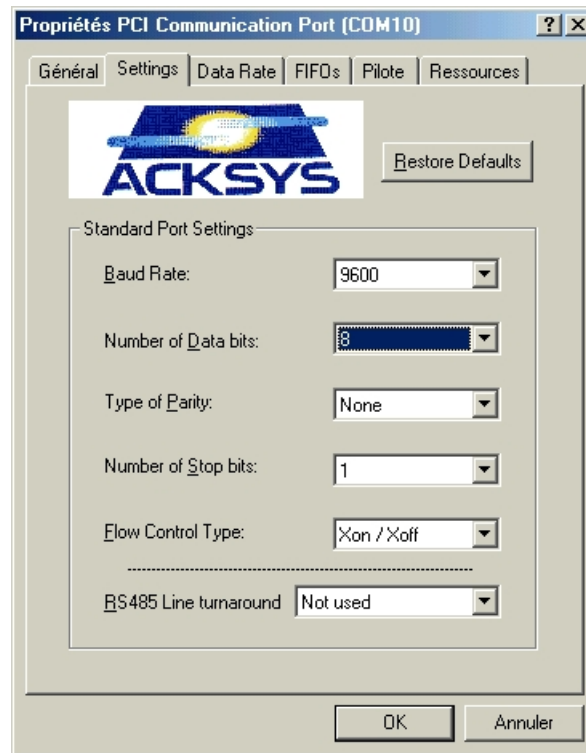
Sous Windows 98, 98 SE et Millennium, l'assistant *ajout de nouveau matériel* est automatiquement lancé lors de la première installation, dès que la carte a été détectée. Suivez les indications de l'assistant pour installer la carte.

L'installation du pilote est maintenant achevée avec succès. Si vous voulez ajouter une carte alors que le pilote a déjà été installé, Windows 98 passera l'assistant d'ajout de matériel et installera automatiquement les ports de communication PCI.

Propriétés des ports de communication

Les nouveaux ports de communication sont fonctionnels dès la fin de l'installation, et sont alors visibles dans le gestionnaire de périphérique (panneau de configuration/système). Pour éditer les propriétés d'un port, double-cliquez la ligne correspondante.





L'onglet "**Settings**" permet de définir les paramètres de communication initiaux, comme dans le cas d'un port de COM standard.

Retournement de ligne en RS485 (RS485 line turnaround)

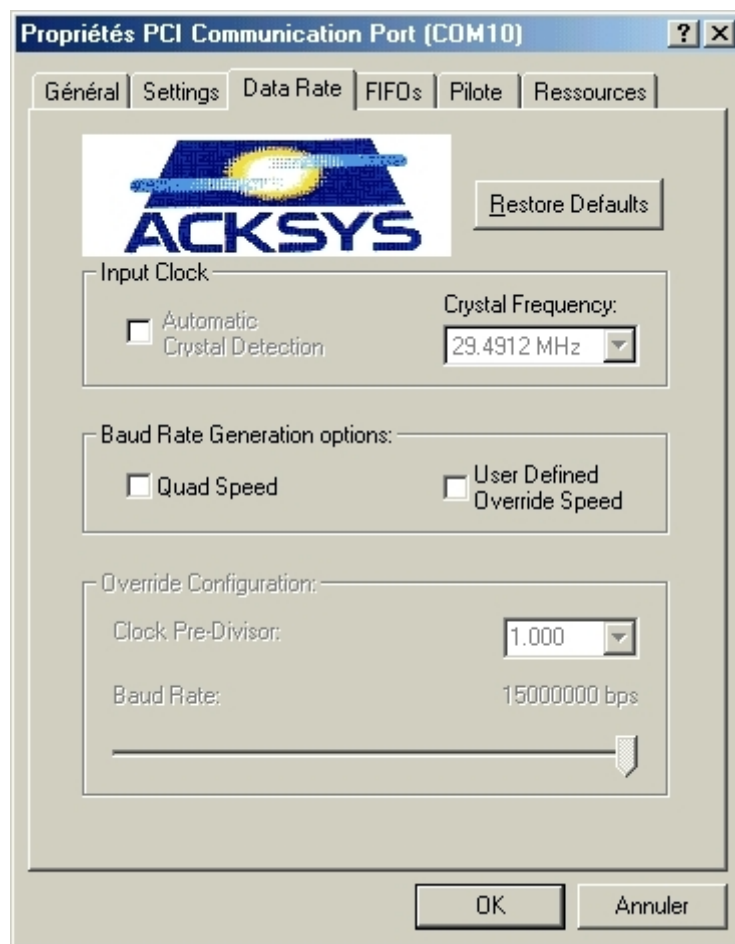
Se reporter à l'annexe A pour tout complément d'informations.

Pour les cas où le retournement de ligne n'est pas nécessaire (RS232, RS422 point à point, RS422 multipoint maître), l'option RS485 line turnaround doit être '**Not used**'.

Pour les autres cas (RS485, RS422 multipoint esclave), 2 choix : '**Driven by application**' ou '**automatic**'

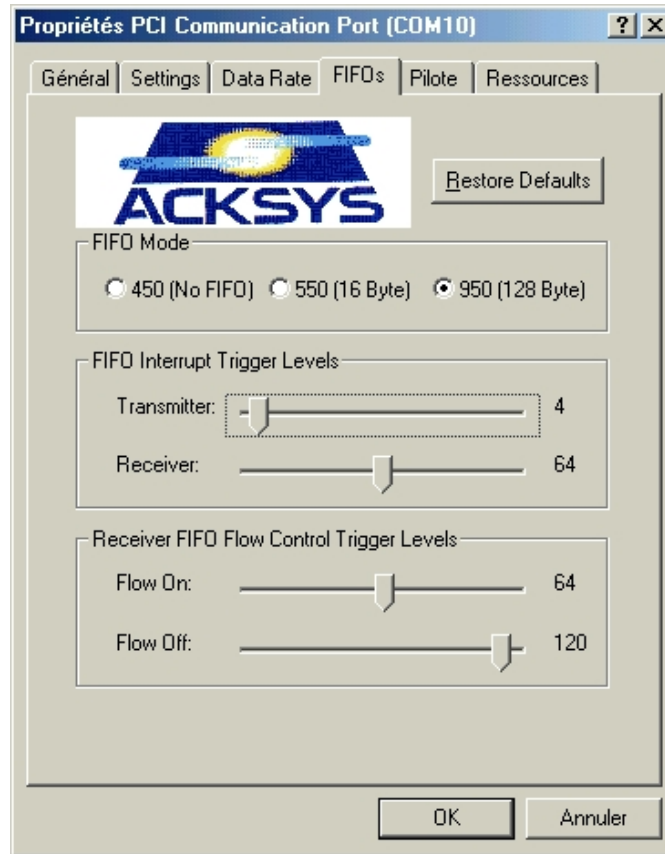
Lorsque l'option '**Driven by application**' est sélectionnée, le pilotage du signal DTR est à la charge de l'application. Cette méthode ne permet pas un contrôle précis du temps de commutation après émission complète de la trame.

Pour un contrôle plus efficace, ou si l'application ne peut pas gérer le signal DTR, sélectionnez l'option '**Automatic**' : le signal DTR sera alors piloté automatiquement par l'UART à chaque envoi de caractère, garantissant ainsi un temps optimal de commutation du sens de communication.



L'onglet '**Data Rate**' permet de configurer des modes de fonctionnement spécifiques de la carte et de visualiser la fréquence de l'oscillateur, qui doit être, pour les cartes standard, de 29,4912 MHz.

- Option '**Quad Speed**' : cette fonctionnalité ne doit pas être utilisée pour les applications conventionnelles, laissez la case non cochée.
- Option '**User Defined Override Speed**' : cette option permet de forcer la vitesse de communication fixée dans le cadre '**Override Configuration**'. Dans ce cas, le pilote de la carte ignore le paramètre de vitesse passé par l'API Windows. Le '**Clock Pre-Divisor**' permet d'appliquer un facteur de division de l'horloge pour obtenir une meilleure précision lorsqu'une vitesse non standard élevée est requise.



L'onglet **FIFOs** permet de fixer les seuils de déclenchement des interruptions d'émission et de réception en fonction du nombre de caractères présents dans les tampons respectifs, ainsi que les seuils pour le contrôle de flux. Les valeurs par défaut sont satisfaisantes pour la plupart des applications classiques.

Réglages des seuils d'interruption :

La valeur définie à l'aide du curseur 'Transmitter' indique le seuil à partir duquel une interruption d'émission sera générée. Par exemple, la valeur par défaut, 4, indique qu'une interruption sera générée dès que le nombre de caractères restants dans la FIFO de transmission passera de 5 à 4. Cette valeur devrait rester petite, mais il peut être souhaitable de l'augmenter aux vitesses élevées ou avec des unités centrales peu puissantes ou chargées.

La valeur définie à l'aide du curseur 'Receiver' indique le seuil à partir duquel une interruption de réception sera générée. Dans le cas de la valeur par défaut, l'interruption se produit lorsque le nombre de caractères présent dans la FIFO de réception passe de 63 à 64. Si le nombre de caractères reçus est inférieur au seuil et n'évolue pas pendant un temps correspondant au temps d'émission de 4 caractères, une interruption 'time out' est générée de façon à avertir le pilote de périphérique.

Dans le cas d'une application effectuant des transferts de blocs de données importants, on choisira de préférence des seuils élevés de façon à réduire le nombre d'interruptions et par conséquent le taux d'occupation CPU. Il n'est pas conseillé cependant d'utiliser les valeurs maximales, notamment lorsque la vitesse de communication est élevée, afin d'éviter les écrasements en réception et les interruptions de l'émission.

3. INSTALLATION DU DRIVER ACKSYS POUR WINDOWS NT 4.0

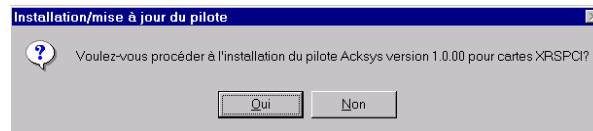
Installez la carte et démarrez Windows NT.

Pour installer le pilote :

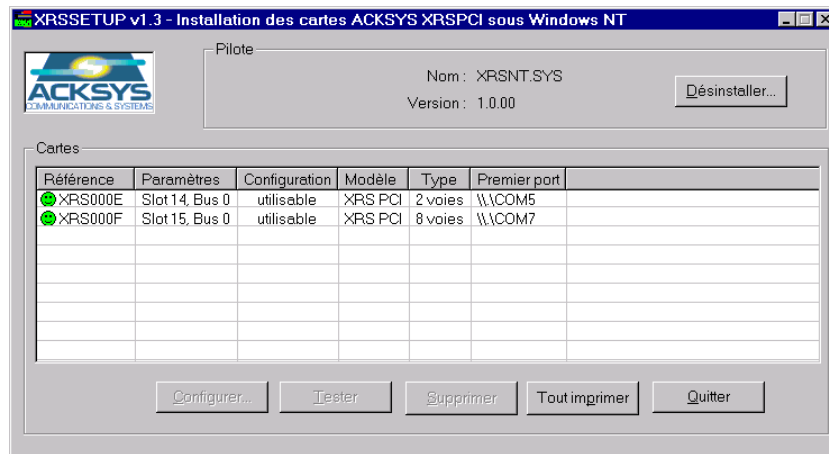
Identifiez vous comme *Administrateur*

Insérez la disquette¹ nommée xRSPCI drivers

Lancez le programme XRSSETUP.EXE à partir du sous répertoire WINNT.

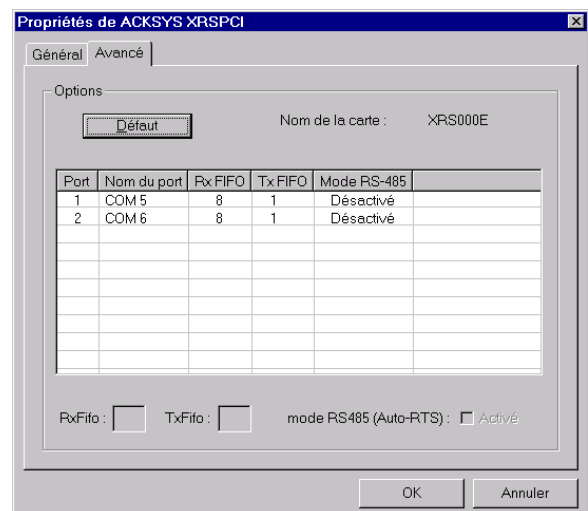
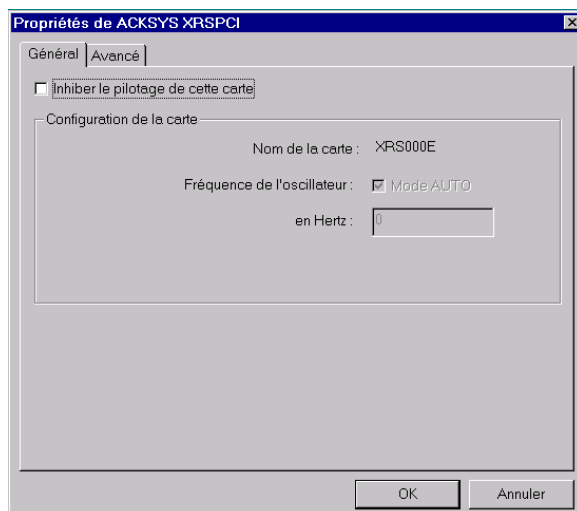


Cliquez sur le bouton *Oui*



Dans cette fenêtre, vous verrez apparaître la liste des cartes détectées par le pilote.

Vous pouvez alors cliquer sur le bouton **Quitter** pour terminer l'installation ou sélectionner une carte et cliquer sur le bouton "**C**onfigurer ..." pour ouvrir la fenêtre *Propriétés de ACKSYS xRSPCI*.



¹ Les drivers pourront éventuellement être fournis sur CD. Dans ce cas, il conviendra de sélectionner le répertoire contenant les drivers xRSPCI.

Dans la fenêtre *Propriétés de ACKSYS XRSPCI* :

Vous pouvez désactiver une carte en cliquant sur la case « *Inhiber le pilotage de cette carte* ».

Vous pouvez spécifier, pour chaque port, le niveau de déclenchement du RxFIFO, la taille du TxFIFO et le mode RS485.

Mode RS485 AutoRTS (comprendre line Turnaround)

Se reporter à l'annexe A pour tout complément d'informations.

Pour les cas où le retournement de ligne n'est pas nécessaire (RS232, RS422 point à point, RS422 multipoint maître), la case à cocher doit être désactivée.

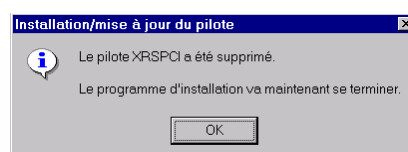
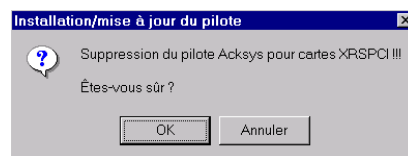
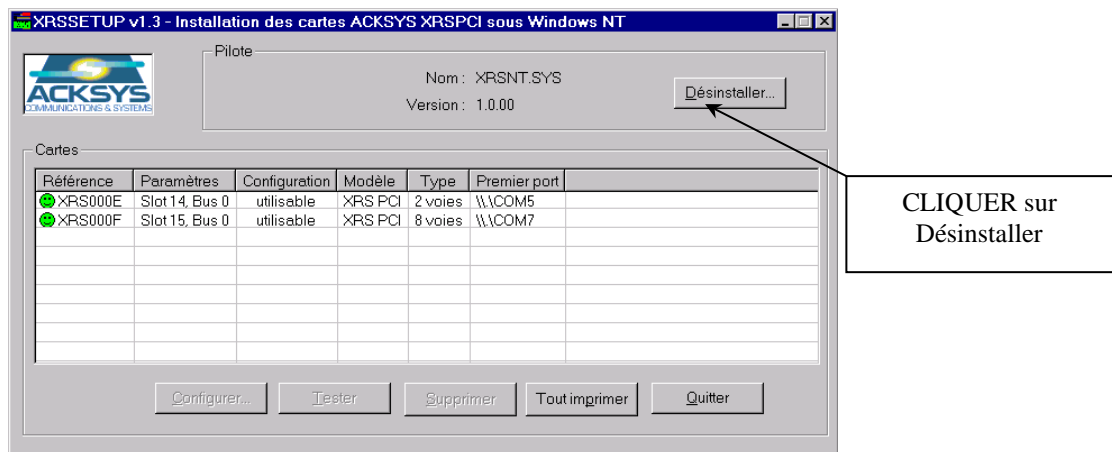
Pour les autres cas (RS485, RS422 multipoint esclave), les 2 choix restent possibles.

Lorsque la case à cocher est désactivée, le pilotage du signal DTR est à la charge de l'application : Cette méthode ne permet pas un contrôle précis du temps de commutation après émission complète de la trame.

Lorsque la case à cocher est activée, le signal DTR sera alors piloté automatiquement par l'UART à chaque envoi de caractère, garantissant ainsi un temps optimal de commutation du sens de communication.

Pour désinstaller le pilote,

lancez XRSSETUP.EXE et cliquez sur le bouton « Désinstaller ».



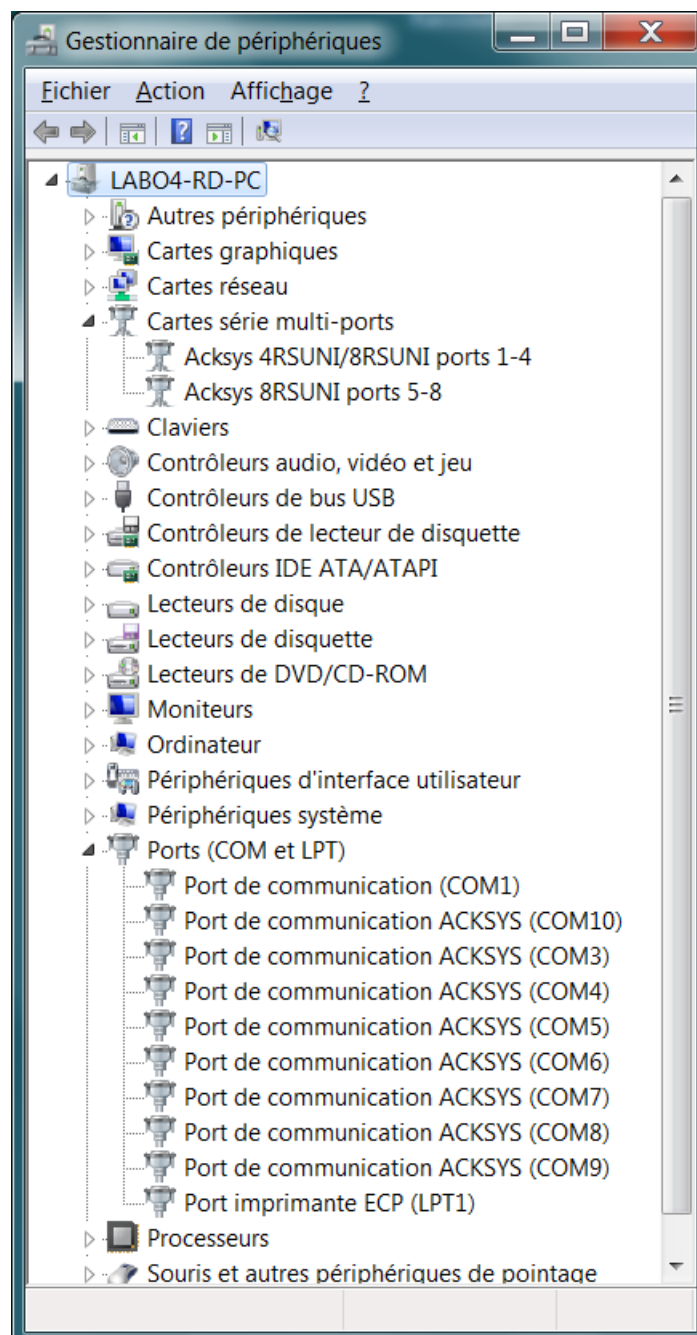
4. INSTALLATION DU DRIVER ACKSYS DE WINDOWS 2000 A WINDOWS 10

Ces systèmes d'exploitation détectent automatiquement la carte. Un assistant d'installation de matériel est automatiquement lancé au démarrage du système, dès que la carte a été détectée. Avant tout, naviguez sur le CD ACKSYS pour trouver votre produit, et exécutez l'installateur correspondant à votre version de Windows. Vous pouvez installer la carte avant ou après. Ensuite suivez les indications de l'assistant d'installation de périphériques.

Remarque : Les pilotes pour les versions anciennes de Windows signalent que le pilote n'est pas signé. Ignorez cet avertissement et forcez l'installation du pilote.

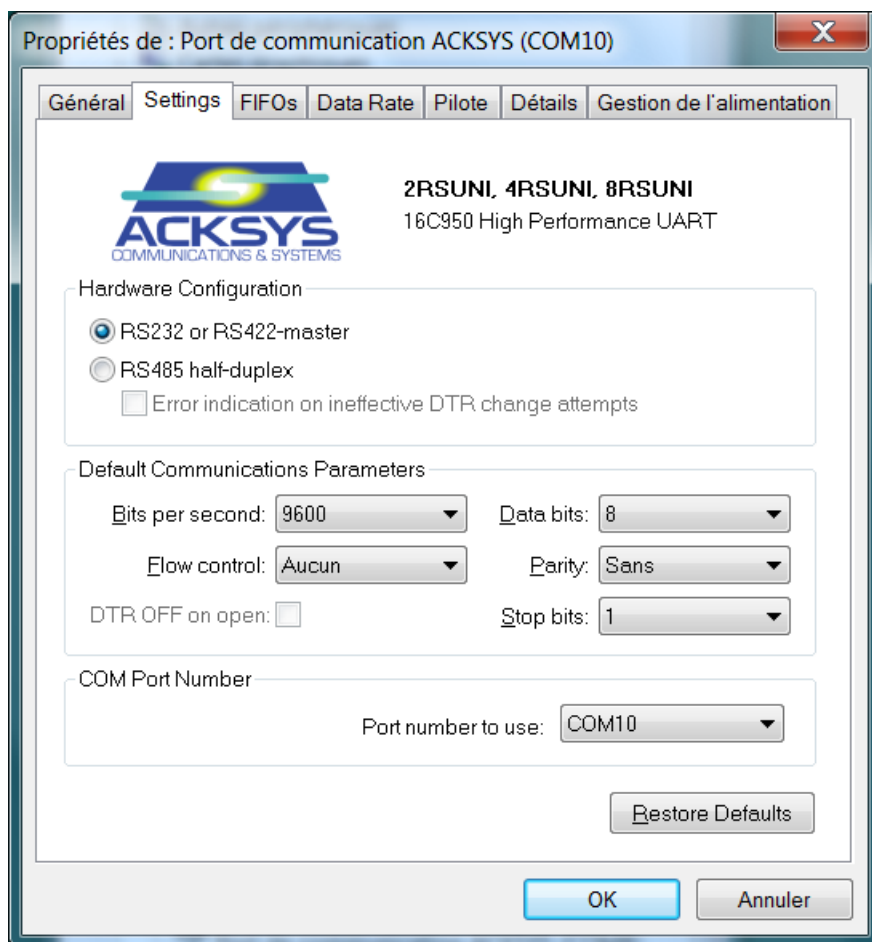
Propriétés des ports de communication

Les nouveaux ports de communication sont visibles dans le gestionnaire de périphériques (panneau de configuration / matériel). Double-cliquez sur un des ports pour éditer ses propriétés.



Onglet « Settings »

Cet onglet permet de définir les paramètres de communication utilisés par défaut dans certaines applications, comme dans le cas d'un port de COM standard. Il doit aussi être utilisé pour définir le type d'interface et valider le retournement automatique de ligne en RS485



« Hardware configuration »

RS232 or RS422-master

L'émission et la réception simultanées ('full duplex') sont autorisées, l'application peut gérer les signaux RTS et DTR.

Si une application est conçue pour gérer elle-même le retournement de ligne (sur une carte d'interface MR400ISO ou un convertisseur RS485 ACKSYS externe), ce mode laisse l'application piloter elle-même le signal RTS. Le signal doit être activé avant l'émission et désactivé lorsque le dernier caractère a été émis. Cette méthode ne permet pas un contrôle précis du temps de commutation après émission complète de la trame.

RS485 or RS422-slave

Dans ce mode, l'UART génère automatiquement un signal de « retournement de ligne » qui est rendu disponible sur les signaux de sortie DTR et RTS.

Utilisez ce mode quand les signaux TxD (AB) sont connectés à un bus en mode esclave, l'émission et la réception étant alternées ('half duplex'). Les cartes d'interface MR400ISO et les convertisseurs RS485 ACKSYS externes utilisent le signal RTS pour libérer le bus quand la carte n'émet pas. Au repos (pas de transactions), la ligne est en réception; dès qu'un caractère ou qu'un groupe de caractères doit être émis, le RTS est activé et la ligne est commutée en mode émission.

Error indication on ineffective DTR changes attempts

Quand le retournement est automatique, le pilotage du signal RTS n'a plus d'effet et le signal DTR est réservé dans le pilote. Si l'application tente de modifier le DTR, il n'y aura aucun effet réel. Cette case indique si l'application doit recevoir ou pas une indication d'erreur dans ce cas.

« Default configuration parameters »

Attention, de même que pour les ports COM de base, ces paramètres ne sont utilisés que par certaines applications Windows. Vérifiez toujours les options de configuration de votre application.

DTR OFF on open

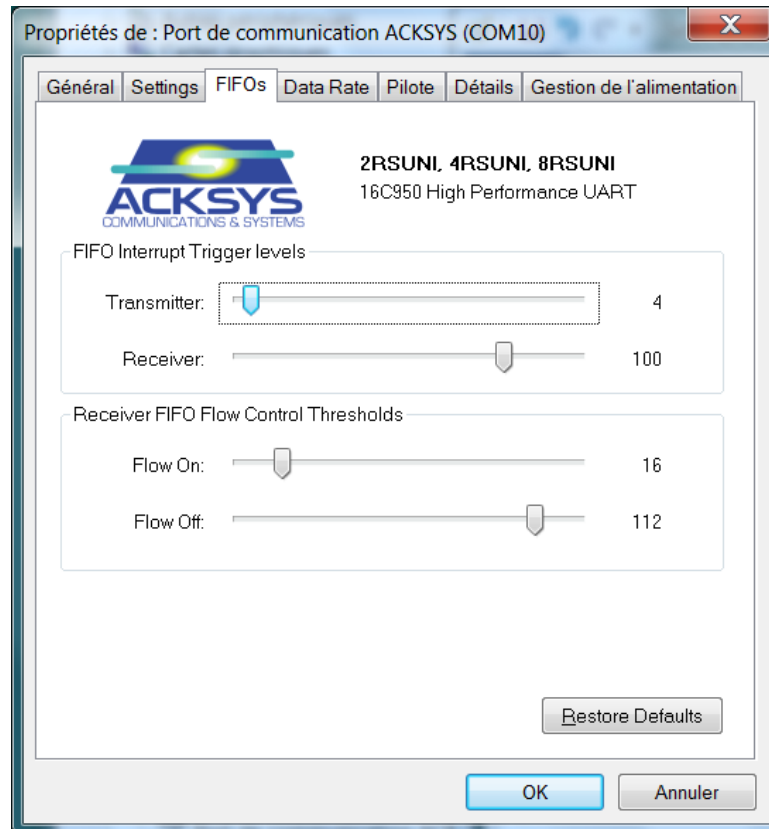
Le comportement par défaut des pilotes série de Windows est d'activer le signal DTR dès l'ouverture du port série. Ceci peut être gênant dans les cas où le DTR est utilisé pour activer des fonctions sur l'équipement série, par exemple pour démarrer l'émetteur d'un MODEM radio. Cocher cette case permet d'assurer que le DTR reste inactif jusqu'à une action volontaire de l'application.

« COM port number »

Le nom du port COM peut être changé ici.

Onglet « FIFOs »

Cet onglet permet de fixer les seuils de déclenchement des interruptions d'émission et de réception en fonction du nombre de caractères présents dans les tampons respectifs, ainsi que les seuils pour le contrôle de flux. Les valeurs par défaut sont satisfaisantes pour la plupart des applications classiques.



Réglages des seuils d'interruption :

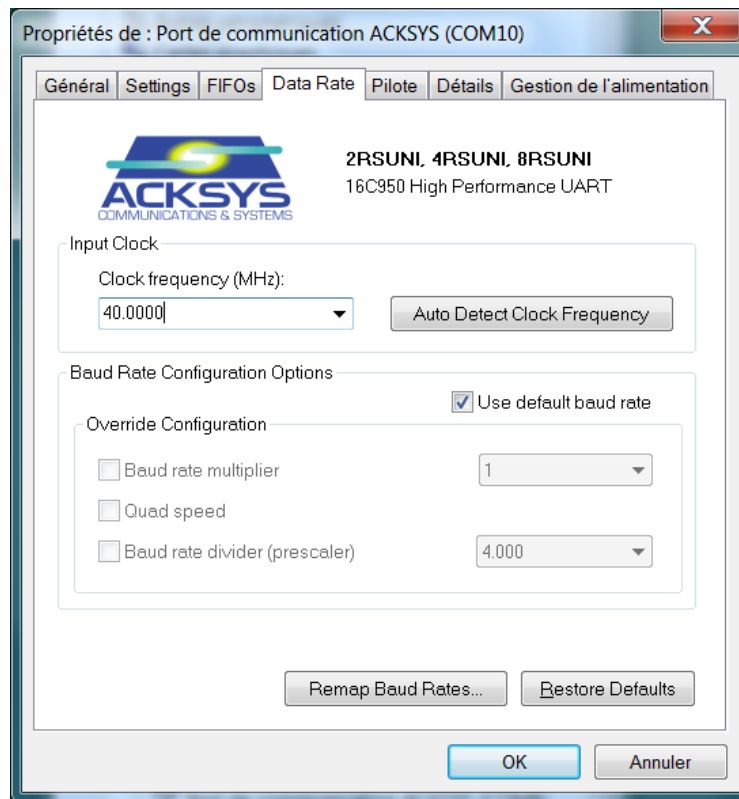
La valeur définie à l'aide du curseur 'Transmitter' indique le seuil à partir duquel une interruption d'émission sera générée. Par exemple, la valeur par défaut, 4, indique qu'une interruption sera générée dès que le nombre de caractères restants dans la FIFO de transmission passera de 5 à 4. Cette valeur devrait rester petite, mais il peut être souhaitable de l'augmenter aux vitesses élevées ou avec des unités centrales peu puissantes ou chargées.

La valeur définie à l'aide du curseur 'Receiver' indique le seuil à partir duquel une interruption de réception sera générée. Dans le cas de la valeur par défaut, l'interruption se produit lorsque le nombre de caractères présent dans la FIFO de réception passe de 63 à 64. Si le nombre de caractères reçus est inférieur au seuil et n'évolue pas pendant un temps correspondant au temps d'émission de 4 caractères, une interruption 'time out' est générée de façon à avertir le pilote de périphérique.

Dans le cas d'une application effectuant des transferts de blocs de données importants, on choisira de préférence des seuils élevés de façon à réduire le nombre d'interruptions et par conséquent le taux d'occupation CPU. Il n'est pas conseillé cependant d'utiliser les valeurs maximales, notamment lorsque la vitesse de communication est élevée, afin d'éviter les écrasements en réception et les interruptions de l'émission.

Onglet « Data Rate »

Cet onglet permet de sélectionner la fréquence de l'oscillateur pour certains modèles spécifiques de cartes. La valeur standard par défaut est 29,4912 MHz.



- La liste '**Clock Frequency (MHz)**' permet d'indiquer directement, si vous la connaissez, la valeur de l'oscillateur monté sur la carte.
- Le bouton '**Auto detect Clock Frequency**' recalcule la valeur la plus probable de l'oscillateur monté sur la carte.
- La case '**Use default baud rate**' devrait rester cochée. Décochée, elle donne accès à des multiplicateurs de vitesse. Les vitesses de transmission réelles différeront alors par un facteur multiplicatif des vitesses spécifiées par l'application.
- Option '**Baud rate divider (prescaler)**' : réduit la vitesse de communication par un facteur de division décimal permettant d'obtenir une meilleure précision pour les vitesses non standards.
- Option '**Baud rate multiplier**' : applique le facteur de multiplication choisi à la vitesse demandée par l'application. Par exemple avec un facteur de 16, lorsque l'application demande une vitesse de 115200 bauds, la vitesse effective sera de 1,8432MHz.
- Option '**Quad Speed**' : ne doit pas être utilisée pour les applications conventionnelles, laissez la case non cochée.
- Le bouton '**Remap Baud Rates**' : permet de substituer une vitesse à une autre. Permet par exemple de faire en sorte que la vitesse 4800 bauds corresponde en réalité à 230400 bauds. Cela est particulièrement utile si vous utilisez des programmes qui ne proposent que des vitesses lentes alors que vos équipements série permettent l'utilisation de vitesses plus rapides.

5. INSTALLATION DU DRIVER ACKSYS POUR LINUX (NOYAU 2.2.X)

La procédure d'installation a été testée initialement sur **Linux Mandrake** version **7.2** et une version du noyau **2.2.17-21**. Pour tout problème de compatibilité avec d'autres versions de Linux, merci de contacter ACKSYS. Tous les programmes décrits ci-dessous se trouvent dans le sous répertoire `linux/kernel22` du media fourni.

Installation du driver

Ce type d'installation n'oblige pas une recompilation du noyau Linux. De plus, elle permet de charger et décharger le driver de façon dynamique.

Copier le fichier `srllinux.o` dans le répertoire

```
/lib/modules/$(shell uname -r)/misc/
```

Démarrage du driver

Pour démarrer le driver il suffit de taper la commande `insmod srllinux`

Pour vérifier que le driver est bien démarré, utilisez la commande `lsmod`.

Si l'oscillateur de la carte n'est pas à 29,4912 MHz, lorsque vous installez le driver vous devez ajouter l'option :

```
input_clock=[Freq. en Hz].
```

Par exemple, pour un oscillateur à 16 MHz, la commande est la suivante :

```
insmod srllinux input_clock=16000000
```

Au prochain redémarrage de Linux, le driver ne se lancera pas automatiquement, pour cela ajoutez le fichier `rc.ack` dans le fichier `rc.serial`.

Si le fichier `rc.serial` n'existe pas créez le, ajoutez la ligne :

```
/etc/rc.d/rc.ack
```

Puis éditez le fichier `rc.local`, ajoutez la ligne suivante :

```
if [ -f /etc/rc.d/rc.serial ]; then
    sh /etc/rc.d/rc.serial
fi
```

Création des noeuds

Il faut créer les noeuds dans le système. Pour cela, éditez le script `msmknod` fournit par ACKSYS, ajustez la constante `MAXPORT` suivant votre carte (si vous avez une carte deux voies mettre 2, si vous avez une carte 4 voies mettre 4 ...) puis exécutez ce script. Il va créer les fichiers de périphérique de la façon suivante :

<i>Carte n°</i>	<i>Numéro de périphérique</i>
1	<code>ttyM0-ttyM[MAXPORT-1]</code>
2	<code>ttyM[MAXPORT]-ttyM[2*MAXPORT-1]</code>

Arrêt du driver

Pour arrêter le driver il suffit de taper la commande `rmmmod srllinux`

6. INSTALLATION DU DRIVER ACKSYS POUR LINUX (NOYAU 2.4.X)

La procédure d'installation a été testée sur **Linux Mandrake** version **9.0** avec une version du noyau 2.4.19, ainsi que sur **Linux redhat** 7.2 avec un noyau 2.4.7-10. Pour tout problème de compatibilité avec d'autres versions de Linux, merci de contacter ACKSYS.

Ce driver est basé sur le Driver Linux Serial 5.05.

Tous les programmes décrits ci-dessous se trouvent sous /linux/kernel24 sur le support de distribution.

Installation du driver

Ce type d'installation n'oblige pas une recompilation du noyau Linux. De plus, elle permet de charger et décharger le driver de façon dynamique.

Copier le fichier **srlxrspci.o** dans le répertoire

```
/lib/modules/$(shell uname -r)/misc/
```

Les fichiers de périphérique peuvent être créés avec les noms standard (ttyS) ou avec un nom choisi au moment de l'installation (ex : ttyA).

Démarrage du driver

Pour démarrer le driver, il faut taper la commande :

```
insmod srlxrspci.o
```

Pour vérifier que le module est bien installé, utilisez la commande **lsmod**.

Cela vient créer les tty dans le système. Par défaut, le driver va créer les terminaux portant le nom ttyA04 jusqu'à ttyAn (n dépendant de la carte).

Si vous souhaitez changer le nom du terminal, il suffit de passer en argument

```
tty_name_p=<nom du terminal>
```

Par exemple si vous voulez créer un terminal avec le nom ttyS, tapez :

```
insmod srlxrspci.o tty_name_p=ttyS
```

Le driver va créer les terminaux ttyS04 jusqu'à ttySn (n dépendant de votre carte).

Pour vérifier le nom du terminal qui est créé, regarder le fichier /var/log/messages.

- ✓ Si l'oscillateur de la carte n'est pas à 29,4912 MHz, lorsque vous installez le module vous devez ajouter l'option :

```
input_clock=[Freq. en Hz].
```

Par exemple, pour un oscillateur à 16 MHz, la commande est la suivante :

```
insmod srlxrspci.o input_clock=16000000
```

- ✓ Si vous voulez utiliser une vitesse non standard, lorsque vous installez le module vous devez ajouter l'option :

```
speed_custom=[vitesse en bauds]
```

Par exemple, si vous souhaitez une vitesse de 76800 bauds, la commande est la suivante :

```
insmod srlxrspci.o speed_custom=76800
```

Pour configurer la vitesse d'un port série, les fonctions à utiliser sont :

```
int cfsetospeed(struct termios *termios_p, speed_t speed); // vitesse
de sortie
int cfsetispeed(struct termios *termios_p, speed_t speed); // vitesse
d'entrée
```

Lorsque vous voulez prendre en compte le paramètre `speed_custom`, vous devez utiliser la constante `EXTA`, comme le montre l'exemple ci-dessous :

```
cfsetospeed(&ma_struct_termios, EXTA);
```

Cette constante `EXTA` est définie par défaut dans le fichier `/usr/include/bits/termios.h` de la façon suivante :

```
#define EXTA B19200
```

Si vous souhaitez utiliser la vitesse 19200 bauds (B19200), vous devez modifier la constante `EXTA` dans `/usr/include/bits/termios.h` et il faudra alors recompiler le driver Linux.

Au prochain redémarrage de Linux, le module ne se lancera pas automatiquement, Pour cela ajoutez le fichier `rc.ack` dans le fichier `rc.serial`.

Si le fichier `rc.serial` n'existe pas créez le, ajouter la ligne :

```
/bin/sh /etc/rc.d/rc.ack
```

Puis éditez le fichier `rc.local`, ajoutez la ligne suivante :

```
if [ -f /etc/rc.d/rc.serial ]; then
    /bin/sh /etc/rc.d/rc.serial
fi
```

La procédure décrite ci-dessus peut être différente sur une autre version de Linux. Vérifiez que dans le fichier `rc.ack` le driver est bien lancé avec les bonnes options.

Création des nœuds

Une fois le driver correctement démarré, Il faut créer les nœuds dans le système. Pour cela taper la commande :

```
mknod <nom terminal> c <major> <minor>
```

Ce qui donne, pour un démarrage avec les options par défaut :

```
mknod ttyA04 c 40 68
```

Pour vérifier le numéro major et minor du terminal créé par le driver, regarder le fichier `/var/log/messages`.

Arrêt du driver

Pour arrêter le driver il suffit de taper la commande `rmmmod srlxrspci`

Références

Fichier d'aide linux relatif aux ports série :

<http://en.tldp.org/HOWTO/Serial-HOWTO.html>

Fichier d'aide linux relatif à la programmation des ports série :

<http://en.tldp.org/HOWTO/Serial-Programming-HOWTO/>

Liste des sites contenant les Linux Howtos :

<http://metalab.unc.edu/LDP/mirrors.html>

Fichier source du driver série linux 5.05 :

<http://sourceforge.net/projects/serial>

http://sourceforge.net/project/showfiles.php?group_id=310

Utilisation du turnaround automatique sous linux 2.4

Lorsque l'on utilise une ligne en mode RS485 ou RS422 esclave, il est possible de gérer le retournement de façon automatique. Pour cela il faut utiliser le code ioctl suivant :

- **ACKSYS_ENABLE_485_MODE** : Valide la fonction de retournement automatique.

Le retournement automatique sera géré par l'UART.

Cet IOCTL ne prend pas de paramètres. La fonction de l'API linux permettant d'envoyer l'iocontrol au driver est ioctl(...). Cette fonction est à envoyer une fois le port de communication ouvert.

La constante associée à cet IOCTL est définie dans le fichier iocontrol.h.

ATTENTION : La valeur de la constante n'est pas identique sur les deux noyaux linux 2.2 et 2.4. Il vous faudra obligatoirement recompiler votre application avec le bon fichier include.

7. INSTALLATION DU DRIVER ACKSYS POUR NOYAUX LINUX 2.6.X

Préparatifs

Vous devez disposer des éléments suivants pour installer le driver:

- Les en-têtes du noyau.
- L'utilitaire linux "setserial" si vous avez besoin d'initialisations particulières.
- Un accès utilisateur privilégié "root".
- Au moins une carte XRSPCI, XRSUNI ou UNXPCI.
- Pour les noyaux antérieurs à 2.6.25, vous devez aussi installer les sources du noyau.

Décompression de l'archive

Le fichier archive est au format tar compressé. Quand vous extrayez son contenu, il se crée un sous-répertoire du répertoire courant.

Ce sous-répertoire contient lui-même deux répertoires:

- utils/ contient des outils pour initialiser les ports,
- driver/ contient, pour chaque sous-version du noyau, les sources du driver.

Compilation pour les noyaux 2.6.10 à 2.6.18

La procédure d'installation a été testée initialement sur une distribution **Debian Sarge** et un noyau **Linux 2.6.10**. Ce driver est basé sur le driver série standard de Linux, qui a été restructuré plusieurs fois dans le noyau 2.6. Des notes spécifiques à la distribution **Fedora** figurent à la fin de ce chapitre. **L'installation sur Fedora Core 6 est similaire.**

Un module driver Linux doit être compilé en y incluant la connaissance de la configuration du noyau dans lequel il sera chargé. Aussi, nous le fournissons seulement en code source, et vous devez le compiler pour l'utiliser. Une fois compilé, vous pouvez charger le module objet dans tout noyau linux correspondant à la configuration utilisée lors de la compilation.

Pour compiler le driver, vous devez:

- Installer les sources et les headers du noyau
- Copier les sources Acksys dans le répertoire drivers/serial des sources du noyau
- Modifier "drivers/serial/Makefile" pour ajouter la ligne en gras à l'endroit indiqué:

```
...  
obj-$(CONFIG_SERIAL_8250) += 8250.o $(serial-8250-y)  
obj-m += acksys8250.o acksys8250_pci.o  
obj-$(CONFIG_SERIAL_8250_CS) += serial_cs.o  
...
```

- Suivre les instructions données par le fournisseur de la distribution Linux, ou les instructions figurant dans le README du noyau. Une génération typique du noyau commence par les étapes suivantes:

```
make mrproper  
make menuconfig (ou make gconfig, ou make xconfig)
```

L'étape de configuration crée un fichier nommé ".config" contenant tous les paramètres de configuration. Vous devez vous assurer que ce fichier correspond à la configuration du noyau dans lequel le module sera chargé. Si vous avez le fichier de configuration du noyau courant, vous pouvez le copier dans ".config" au lieu d'exécuter "make menuconfig". Voir page suivante.

Sur certaines distributions la configuration courante est dans /boot/config (Debian, Mandrake). Sur d'autres elle sera dans /lib/modules/`uname -r`/include, ou dans /lib/modules/`uname -r`/build/include (RedHat...), ou dans /usr/src/linux/include, ou dans /proc/config.gz (SuSe)... Vous devez vérifier cela vous-même.

- Si vous utilisez le .config exact qui correspond à votre noyau, vous pouvez sauter la compilation et l'installation. Construisez seulement les modules nécessaires :
 - ✓ Compilez les modules, par exemple:
`make M=drivers/serial`
 - ✓ Déplacez les modules générés dans le répertoire système des modules :
`libmod=/lib/modules/`uname -r`/kernel/drivers/serial`
`cp acksys8250.ko acksys8250_pci.ko $libmod`
`depmod`
- Si vous utilisez un nouveau .config que vous avez adapté à vos besoins, vous devez maintenant créer un nouveau noyau complet. Les modules Acksys seront ajoutés automatiquement.
 - ✓ Compilez complètement le noyau :
`make`
 - ✓ Installez le noyau et ses modules :
`make install modules_install`
 - ✓ Configurez GRUB ou LILO. Suivez les indications fournies par la commande ci-dessus.
 - ✓ Si nécessaire pour votre distribution, créez un fichier initrd (**xxx** est la version du noyau) :
`mkinitrd -o /boot/initrd.img-xxx xxx`

Compilation pour les noyaux 2.6.26 et plus

La compilation du driver génère un module externe. La procédure d'installation a été testée initialement sur une distribution **Debian Lenny** et un noyau **Linux 2.6.26**. Ce driver est basé sur le driver série standard de Linux, qui a été restructuré plusieurs fois dans le noyau 2.6. Pour tout problème de compatibilité avec d'autres versions de Linux, vérifiez en premier lieu le fonctionnement sur cette version.

ATTENTION : un module driver Linux doit être compilé en y incluant la connaissance de la configuration exacte du noyau dans lequel il sera chargé. Aussi, nous le fournissons seulement en code source, et vous devez le compiler pour l'utiliser. Une fois compilé, vous pouvez charger le module objet dans tout noyau linux correspondant à la configuration utilisée lors de la compilation.

Pour compiler le driver :

- Installez les headers et les outils de compilation du noyau.
- Utilisez l'invite de commande (terminal).
- Allez dans le répertoire des sources du driver approprié pour votre version de noyau.
- Lancez la compilation en tapant :
`make`
- Installez le module généré dans le répertoire « /lib/modules » en tapant :
`make install`

Installation du driver

Le driver peut être chargé et déchargé dynamiquement. Chargez-le avec cette commande :

```
modprobe acksys8250_pci
```

Les ports sont nommés /dev/ttyS<nombre>. <nombre> dépend du modèle de carte Acksys, du nombre de ports sur chaque carte, et de l'ordre des cartes sur le bus PCI. Vous pouvez contrôler les ports reconnus avec :

```
acksys8250_install.sh -p
```

Si vous avez besoin d'arrêter le driver, tapez cette commande :

```
modprobe -r acksys8250_pci
```

Configuration du driver

- ✓ Si vous utilisez une carte avec un oscillateur non standard (autre que 29,4912 MHz), vous devez exécuter la commande suivante pour chaque port de cette carte :

```
setserial /dev/ttySxxx baud_base <osc divided by 16>
```

Voici des exemples pour gérer certaines valeurs d'oscillateur sur une 2RSPCI ou 2RSUNI:

Oscillateur installé	Commande à passer avant l'utilisation
29,4916 MHz	Rien - c'est la valeur par défaut
48 MHz	setserial /dev/ttyS100 baud_base 3000000 setserial /dev/ttyS101 baud_base 3000000
60 MHz	setserial /dev/ttyS100 baud_base 3750000 setserial /dev/ttyS101 baud_base 3750000

- ✓ Si vous avez besoin d'une vitesse de transmission non standard, vous devez exécuter les commandes suivantes sur le port concerné :

```
setserial /dev/ttySxxx divisor <uartdivisor>  
setserial /dev/ttySxxx spd_cust
```

Ceci remplace la vitesse de 38400 bauds avec celle spécifiée. Par exemple, si vous voulez obtenir 76800 bauds avec l'oscillateur à 29,4912 MHz, le diviseur est $29491200 / 16 / 76800 = 24$, et donc la commande est la suivante :

```
setserial /dev/ttyS100 divisor 24 spd_cust
```

Après cela, toute tentative de paramétrer la vitesse 38400 bauds installera 76800 bauds à la place. Il sera impossible d'utiliser 38400 bauds, sauf à exécuter la commande inverse :

```
setserial /dev/ttyS100 spd_normal
```

- ✓ Si vous prévoyez de connecter l'un des ports sur un bus RS485, vous devez passer la commande suivante :

```
acksys8250_install.sh -a /dev/ttySxxx
```

Utilisation du driver

Le driver est entièrement compatible avec le driver série Linux de base fourni dans le noyau version 2.6. Pour plus d'informations, référez-vous aux pages suivantes du manuel : `termios(3)`, `ttyS(4)`, `tty_ioctl(4)`, `setserial(8)`.

Pour basculer un port en mode retournement RS485 automatique, vous pouvez :

- Soit utiliser l'utilitaire fourni à partir d'un shell, interactif ou non, ou depuis un programme compilé utilisant l'appel "system(3)" :

```
acksys8250_install.sh -a /dev/ttySxxx
```

- Soit utiliser l'ioctl suivant : `TIOCSERSETRS485`

Activer le retournement RS485 automatique :

```
#ifndef TIOCSERSETRS485
#define TIOCSERSETRS485 0x5461 /* enable rs-485 */
#endif
... fd = file descriptor of the opened XRS port ...
i = ioctl(fd, TIOCSERSETRS485, 1);
if(i < 0) {
    perror("TIOCSERSETRS485");
}
printf("port set to RS485 auto turnaround using DTR.\n");
```

Désactiver le retournement RS485 automatique :

```
#ifndef TIOCSERSETRS485
#define TIOCSERSETRS485 0x5461 /* enable rs-485 */
#endif
... fd = file descriptor of the opened XRS port ...
i = ioctl(fd, TIOCSERSETRS485, 0);
if(i < 0) {
    perror("TIOCSERSETRS485");
}
printf("port set to normal full-duplex mode.\n");
```

Complément: Compilation du driver sous Fedora Core 4

La procédure suivante est publiée avec l'aimable autorisation de la société RT2I.

Installation du pilote xRSPCI - xRSUNI pour Linux 2.6

Installer les sources du système (kernel-2.6.11.1369_FC4.src.rpm)

```
# rpm -ivh kernel-2.6.11.1369_FC4.src.rpm
# rpmbuild -bp --target=noarch /usr/redhat/SPECS/kernel-2.6.spec
Les sources se trouvent dans /usr/src/redhat/BUILD/kernel-2.6.11/linux-2.6.11
```

dans le répertoire drivers/serial, copier les sources des drivers acksys

ajouter la ligne dans le Makefile (voir doc acksys)

```
obj-$ (CONFIG_SERIAL_8250) += 8250.O $(serial-8250-y)      ligne existante
obj-m += acksys8250.o acksys8250_pci.o                    ligne à rajouter
obj-$ (CONFIG_SERIAL_8250_CS) += serial_cs.o              ligne existante
```

modifier le source "acksys8250.c", ajouter après la ligne 50: : *(note Acksys: ceci semble inutile en kernel 2.6.18)*

```
#define uart_match_port ack_uart_match_port
```

compiler les modules

```
# make mrproper
# make oldconfig
# make M=drivers/serial
```

si il y a des erreurs

```
# cd /usr/src/redhat/BUILD/kernel-2.6.11/linux-2.6.11/include/linux
# mkdir linux
# cp autoconf.h ./linux
# cd ..
# cd ..
# make M=drivers/serial
```

installer les modules

```
# cd drivers/serial
# libmod=/lib/modules/2.6.11-1.1369_FC4smp/kernel/drivers/serial
# cp acksys8250.ko acksys8250_pci.ko $libmod
# depmod
```

tester si le module se charge bien par

```
# modprobe acksys8250_pci
(rajouter cette ligne dans le fichier /etc/rc.d/rc.serial pour que le module soit chargé au boot)
# lsmod
(vérifier que les modules acksys8250 et acksys8250_PCI sont bien dans la liste des modules chargés)
```

lancer le script acksys8250_install.sh fournis avec les drivers pour lister les ports disponibles

```
#!/.acksys8250_install.sh -P
pour une 2RSPCI on doit obtenir:
0000:03:02 2RSPCI ttyS100
0000:03:02 2RSPCI ttyS101
```

si les ports n'existent pas les créer avec la commande

```
#!/.acksys8250_install.sh -d
```

vérifier à nouveau que les ports existent par

```
#!/.acksys8250_install.sh -P
```

vérifier que les ports fonctionnent correctement

```
# stty </dev/ttyS100
# stty </dev/ttyS101
# ls > /dev/ttyS100
# ls > /dev/ttyS101
```

8. ANNEXE A RETOURNEMENT DE LIGNE

Le line turnaround (retournement de ligne) est un mécanisme nécessaire dans le cas où la ligne d'émission et de réception d'un port série sont multiplexées (ligne half duplex) ou lorsque la ligne d'émission de ce port est multiplexée avec la ligne d'émission d'autres ports série.

Il permet de mettre la ligne d'émission en haute impédance lorsque le port série n'a rien à émettre. La carte ACKSYS utilise le signal DTR pour assurer ce mécanisme. Ce signal doit être désactivé (DISABLE) durant la phase d'émission et activé (ENABLE) une fois qu'elle est complètement terminée.

Ce mécanisme doit être utilisé pour les ports série RS485 ou RS422 esclave.

Ce mécanisme n'est pas utilisé pour les ports série RS232, RS422 point à point ou RS422 multipoint maître.

Lorsque le retournement de ligne est activé, la ligne doit être polarisée par des résistances de polarisation, qui vont assurer un niveau électrique stable au repos. Une seule polarisation par ligne est nécessaire.

9. ANNEXE B PROBLEMES RENCONTRES

La carte n'est pas détectée par le BIOS

Vérifiez l'option PCI/PnP dans le programme setup de la carte mère, et mettez-la en AUTO.

Vérifiez que la carte est bien insérée dans le slot.

Essayez d'autres slots jusqu'à ce que vous en trouviez un bon. En cas d'échec, essayez la carte dans un autre PC pour valider son fonctionnement. Le cas échéant, consultez le constructeur du PC pour obtenir une mise à jour du BIOS.

La carte n'est pas détectée par Windows 95/98/2000/Millennium/Xp

Vérifiez le premier problème.

Vérifiez dans la fenêtre de Propriétés système si la carte n'est pas déjà reconnue en tant que carte PCI standard ou carte adaptatrice multifonction. Si c'est le cas, supprimez l'entrée correspondante et cliquez sur le bouton rafraîchir jusqu'à ce que l'assistant d'ajout de matériel soit lancé.

Réinstallez Windows.

La communication entre la carte et votre équipement ne fonctionne pas

Vérifiez la connexion entre votre équipement et la carte.

Vérifiez les paramètres de communication (vitesse, parité, nombre de bits de stops, contrôle de flux) de chaque côté.

En mode RS422/RS485, l'utilisation de l'appellation +/- pour réaliser le câblage peut ne pas fonctionner. Ceci est dû à un défaut de normalisation, un constructeur appellera donc + ce qu'un autre appellera -. Il convient alors dans ce cas de relier le signal + sur le signal -.

En mode RS485 ou RS422 esclave, vérifier que le retournement est bien piloté.

Contacteur ACKSYS

ACKSYS
COMMUNICATIONS & SYSTEMS

ZA Val Joyeux
10, rue des entrepreneurs
78450 Villepreux - FRANCE

Téléphone: +33 (0)1 30 56 46 46

Télécopie: +33 (0)1 30 56 12 95

Web: www.acksys.fr

Support technique: support@acksys.fr

Ventes: sales@acksys.fr