DRIVER USER MANUAL

xRSUNI, xRSUNI-LP, xRSPCI
PCI MULTIPORT SERIAL ADAPTERS

ACKSYS
COMMUNICATIONS & SYSTEMS

DRIVER

DRIVER USER MANUAL

TRADE MARKS ®

- *ACKSYS* is a registered trademark of *ACKSYS*.
- Windows Seven, Windows Vista, Windows XP, Windows 2000, Windows NT, Windows 95, Windows 98 and Windows ME are registered trademarks of Microsoft.
- Linux is a registered trademark of Linus Torvalds.

NOTICE

ACKSYS ® in no ways guarantees the contents of the present document, and accepts no responsibility as to the equipment's value or suitability for the user's needs.

ACKSYS ® can in no case be held responsible for any errors that may exist in this document, nor for any damage, of any size, resulting from the supply, the functioning or the use of the equipment.

ACKSYS ® reserves the right to revise this document from time to time or to change its contents, with no obligation to inform anyone.

TABLE OF CONTENTS

1. INTRODUCTION

This manual describes Windows & Linux driver installation for the following boards :

xRSUNI-232 : PCI serial adapters, RS232, 2, 4 & 8 ports, PCI bus 5V & 3V.

xRSUNI-400 : PCI serial adapters, RS422/485 2, 4 & 8 ports, PCI bus 5V & 3V.

xRSUNILP-232 : PCI serial adapters, RS232 4 & 8 ports, PCI bus 5V & 3V, Low profile

xRSPCI-232 : PCI serial adapters, RS232 2, 4 & 8 ports, PCI bus 5V.

xRSPCI-400 : PCI serial adapters, RS422/485 2, 4 & 8 ports, PCI bus 5V.

2. INSTALLING THE ACKSYS DRIVER FOR WINDOWS 98 & ME

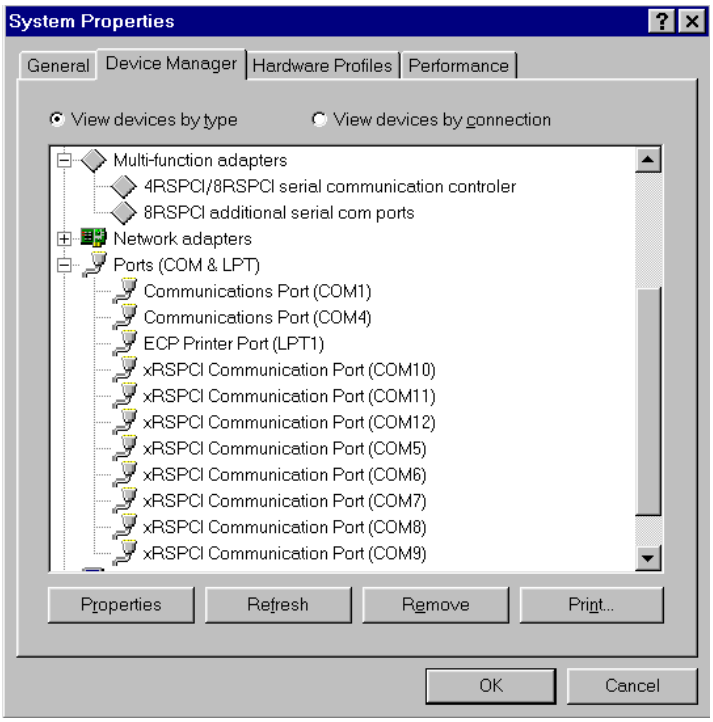
Install the card in the PC and start up Windows.
As this is a PCI card, the configuration will be automatically updated whenever the card is installed or removed.

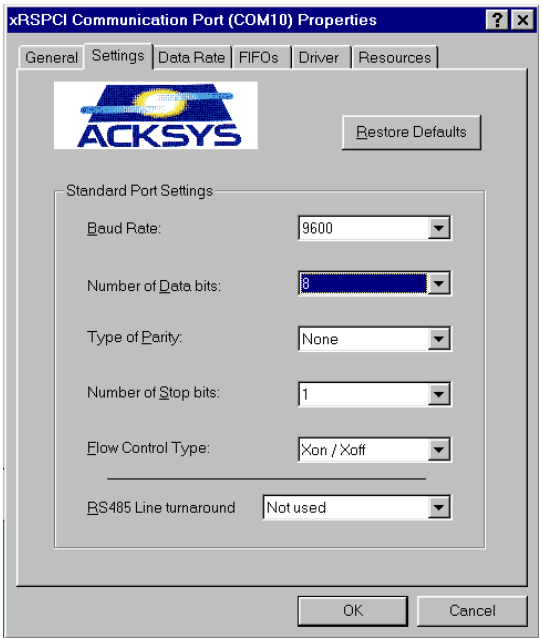
Under Windows 98, 98 SE and ME, the *Add new hardware* wizard will automatically run the first time the card is installed, as soon as the card is detected. Follow the wizard instructions .

The driver has now been successfully installed. If you wish to add a card after the driver has been installed, Windows 98 will skip the *Add new hardware* wizard and will install the PCI communications ports automatically.

Properties of the communications ports

The new communications ports are available as soon as installation is complete, and are visible in the device manager (Control panel/System). To edit the properties of a port, double-click the corresponding line.





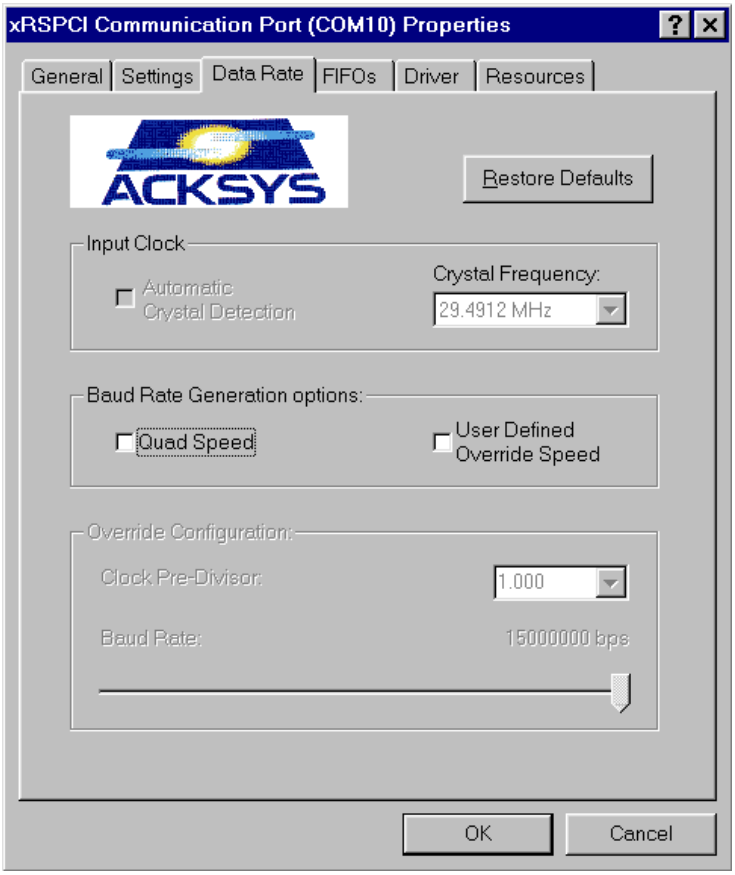
The "**Port settings**" tab enables you to define the initial communications parameters, as for a standard COM port.

RS485 line turnaround
See appendix A for further informations.

If the line turnaround is required, (RS485, RS422 multidrop for the slave nodes), RS485 Line turnaroud must be set to: **'Driven by application'** or **'automatic'**.
If not (RS232, RS422 point to point, RS422 multidrop for the master node). RS485 Line turnaroud must be set to: '**Not used**'.

When the "**Driven by application**" option is selected, the Windows software is in charge of controlling the DTR signal. Don't use this method if DTR cannot be handled in timely manner. DTR must be disabled before beginning the transmission and enabled after the completion of the transmission. Since only a single driver can be enabled on a network at one time it is important that the driver is disabled as quickly as possible after transmission to avoid two drivers trying to control the lines simultaneously, a condition called line contention.

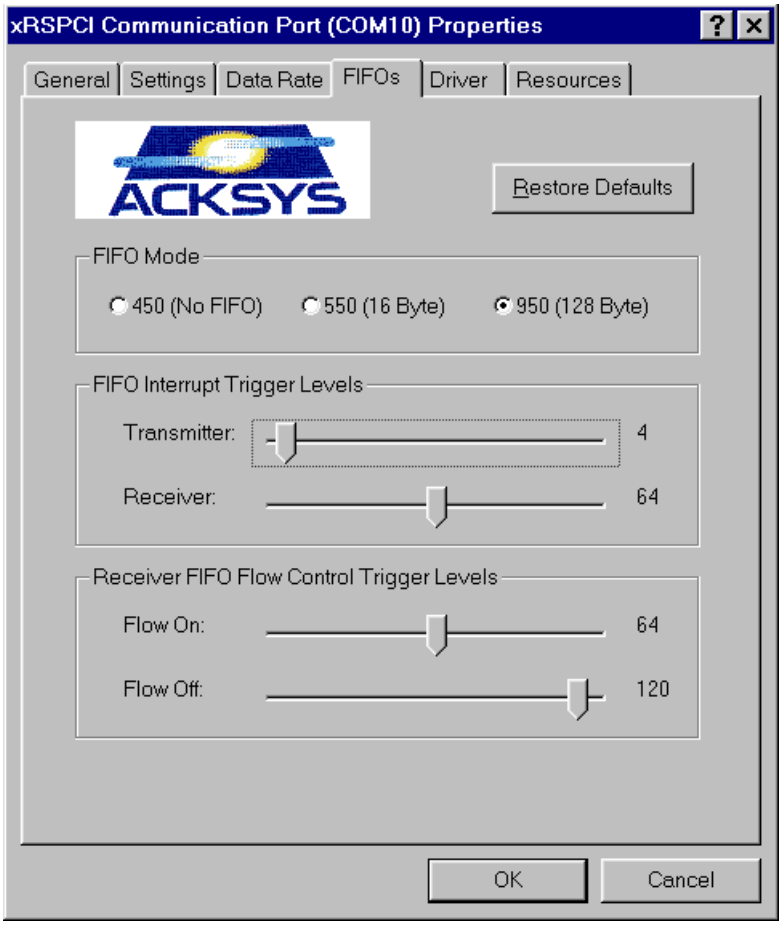
For more efficient control, or if the application cannot manage the DTR signal, select the "**Automatic**" option. The DTR signal will now be driven automatically by the UART whenever a character is transmitted, guaranteeing an optimal timing.



The "**Data Rate**" tab enables you to configure the card's specific operating modes and to visualise the frequency of the oscillator, which, for standard cards, should be 24.4912 MHz.

- '**Quad Speed**' option: this functionality should not be used for conventional applications – the tick box should be left clear.

- '**User Defined Override Speed**' option: this option enables you to enforce the communications speed given in the '**Override Configuration'** box. In this case, the driver ignores the speed parameter passed by the Windows API. The '**Clock Pre-Divisor**' option enables you to apply a division factor to the clock to obtain better precision when a high non-standard speed is required.



The **FIFOs** tab enables you to set the interrupt trigger levels for transmission and reception according to the number of characters in the respective buffers, as well as the flow control trigger levels. The default values are satisfactory for most traditional applications.

Configuring the interrupt trigger levels:

The value defined using the "Transmitter" cursor gives the level from which a transmitter interrupt will be generated. For example, the default value, 4, indicates that an interrupt will be generated as soon as the number of characters in the transmission buffer drops from 5 to 4. This value should remain low, but it may be advisable to increase it for higher speeds or with slow or overloaded CPUs.

The value defined using the "Receiver" cursor gives the level from which a receive interrupt will be generated. In the case of the default value, the interrupt occurs when the number of characters in the receive buffer rises from 63 to 64. If the number of characters received is less than the threshold, and does not change during a period corresponding to the time to transmit 4 characters, a time-out interrupt is generated to warn the peripheral's driver.

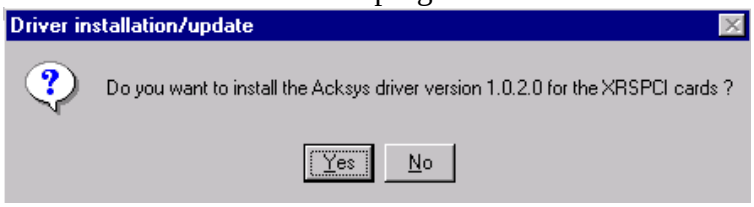
In the case of an application transferring large blocks of data, it is advisable to choose high thresholds in order to reduce the number of interrupts and thus the load on the CPU. However, it is not recommended that the maximum values be used, especially when communicating at high speed, in order to avoid reception overwrites and transmission interruptions.

3. INSTALLING THE ACKSYS DRIVER FOR WINDOWS NT 4.0

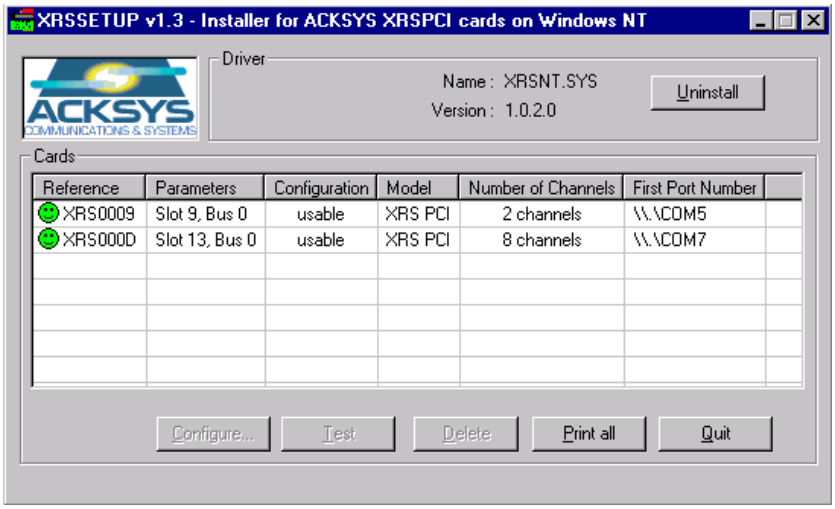
Install the card and power up the PC.

To install the driver:

Start Windows NT and identify yourself as *Administrator*
Insert the xRSPCI driver disk[1]
Run the XRSSETUP program from WINNT subdirectory



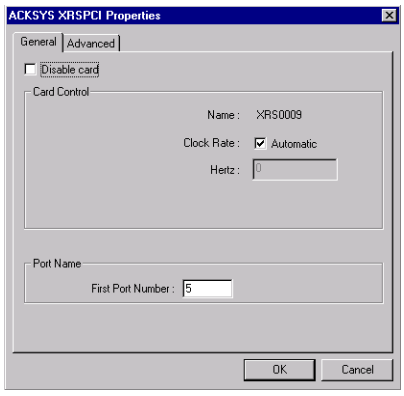
Click the ***Yes*** button

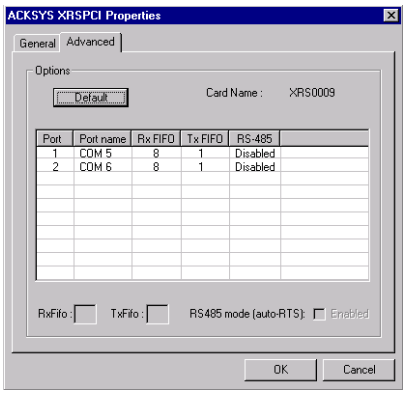


In this window, you will see a list of cards detected by the driver.
You can either click **Exit** to finish the installation or select a card and click **Configure** to open the *Properties of ACKSYS xRSPCI* window.

In the *Properties of ACKSYS XRSPCI* window:

You can deactivate a card by clicking on the *Disable the driver of this card* box.
For each port, you can specify the RxFIFO interrupt trigger level, the size of the TxFIFO buffer and the RS485 mode.





[1] The drivers may be supplied on CD. In this case, you will need to select the folder containing the xRSPCI drivers.

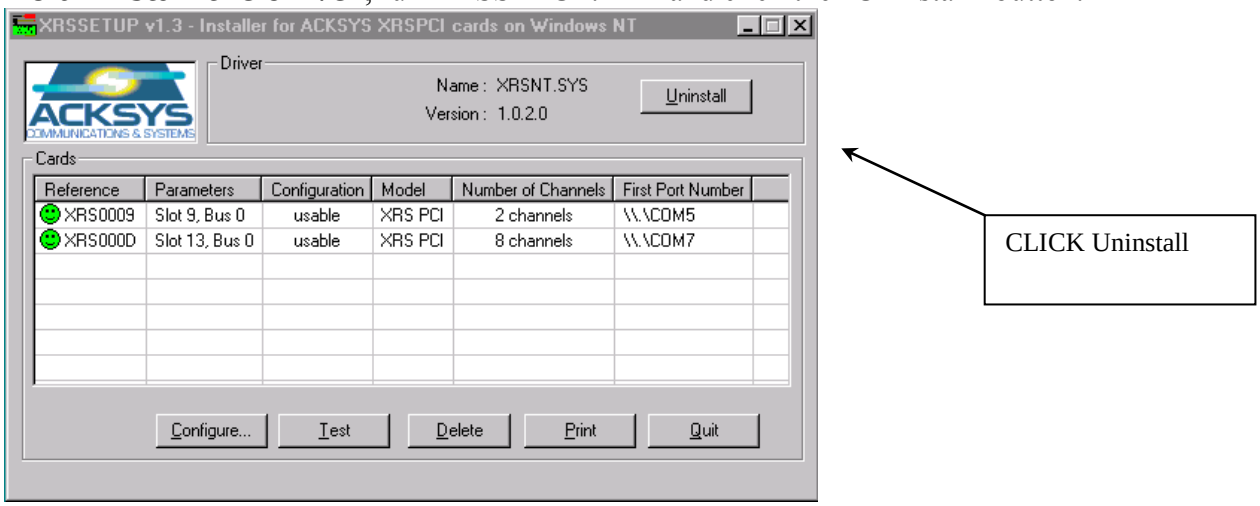
Mode RS485 AutoRTS (understand Line Turnaround)

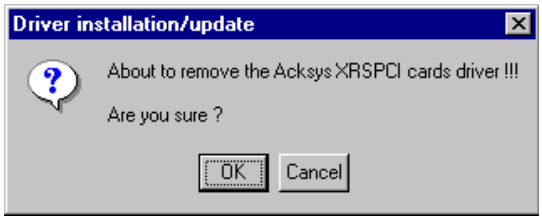
See appendix A for further informations.

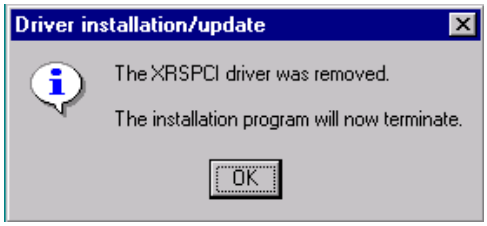
If the line turnaround is required, (RS485, RS422 multidrop for the slave nodes), *mode RS485 AutoRTS* check box can be cheked or not.
If not (RS232, RS422 point to point, RS422 multidrop for the master node), *mode RS485 AutoRTS* check box must not be checked.

If checked, the DTR signal will now be driven automatically by the UART whenever a character is transmitted, guaranteeing an optimal timing.
If not checked, the Windows software is in charge of controlling the DTR signal. Don't use this method if DTR cannot be handled in timely manner. DTR must be disabled before beginning the transmission and enabled after the completion of the transmission.. Since only a single driver can be enabled on a network at one time it is important that the driver is disabled as quickly as possible after transmission to avoid two drivers trying to control the lines simultaneously, a condition called line contention.

To uninstall the driver, run XRSSETUP.EXE and click the "Uninstall" button.







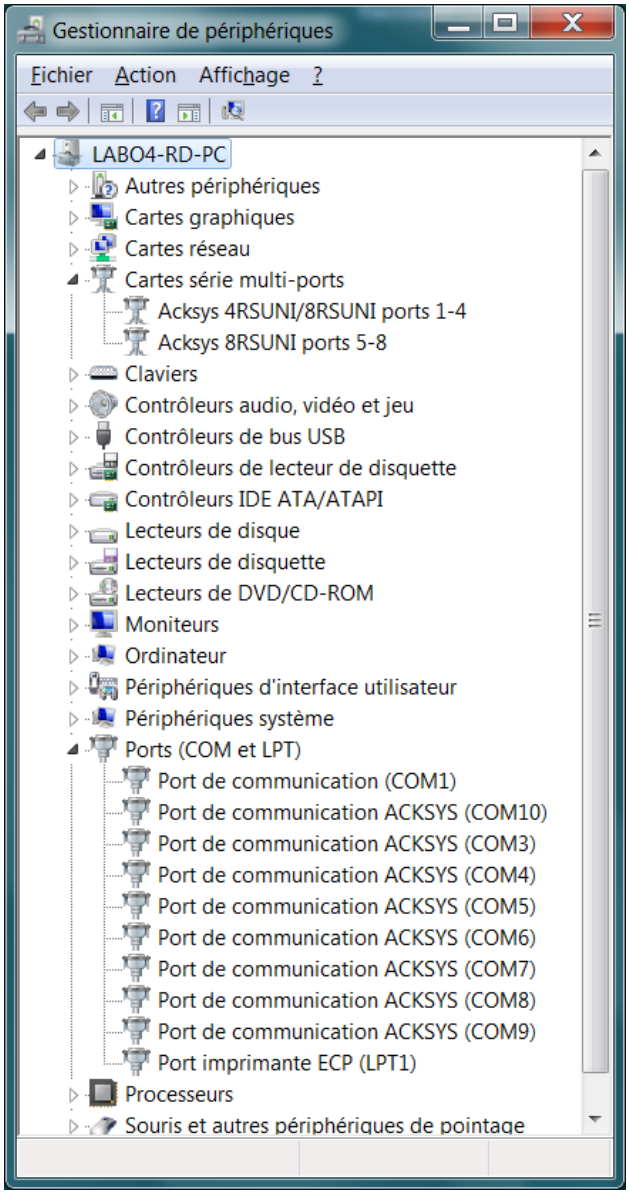
4. INSTALLING THE ACKSYS DRIVER FOR WINDOWS 2000/XP/VISTA/SEVEN

These operating systems automatically detect the new card. A hardware installation wizard is automatically run when the system starts up, as soon as the card is detected. Follow the wizard instructions. Drivers for Windows are located on the ACKSYS CD-ROM.

NOTE FOR WINDOWS XP and WINDOWS VISTA : The wizard may announce you that the driver is not certified. Ignore this message in order to continue the installation of the card.

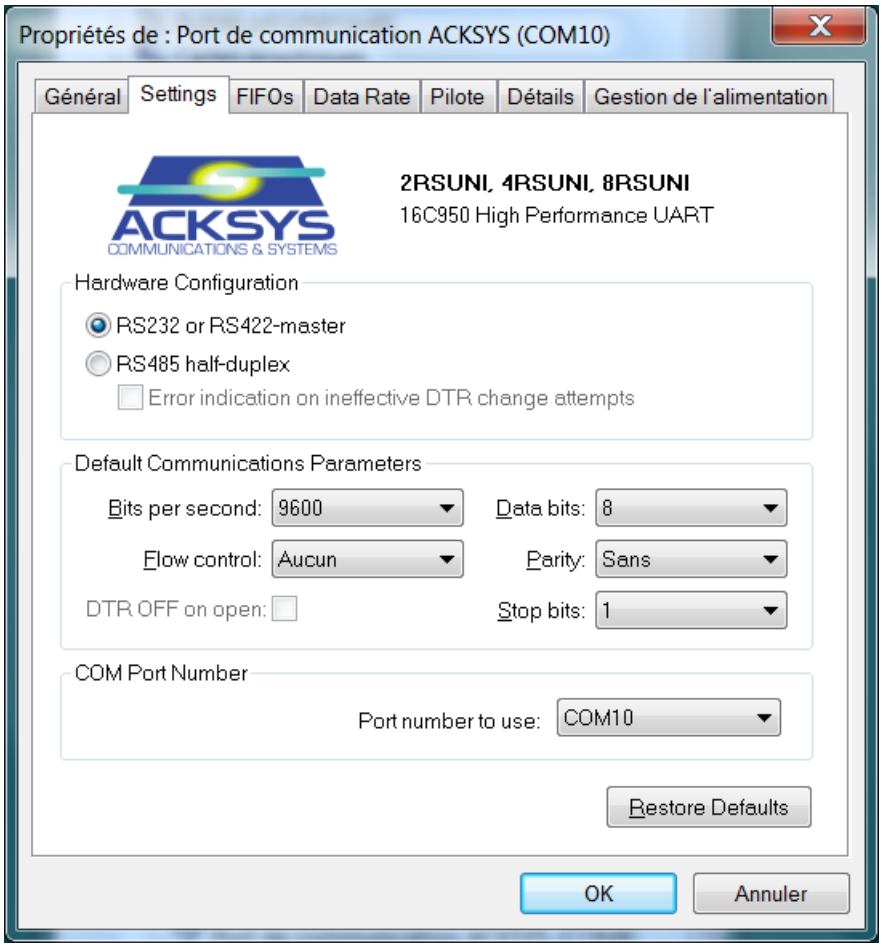
Properties of all the communications ports

The new communications ports are visible in the device manager (control panel/hardware). Double-click one of the ports to edit its properties.



The “Settings” tab

This tab enables the default communications parameters to be defined, as with a standard COM port. It can also be used to define the interface type and automatic line turnaround for RS485.



« Hardware configuration »

RS232 or RS422-master :

Simultaneous transmission and reception (‘full duplex’) is allowed. The software application can handle the RTS and DTR signals.

If a software application is designed to drive the turnaround by itself (when using a MR400ISO or an ACKSYS RS485 external converter), this mode allows the application to drive the RTS signal. This signal must be activated before transmission and disabled once the last character is sent. This method does not allow a precise control of the commutation time after a complete frame transmission.

RS485 or RS422-slave :

In this mode, the UART automatically generates a turnaround signal which is available on the DTR and RTS output signals.

Use this mode when the TxD (AB) signals are connected to a bus in slave mode, transmission and reception being alternated (‘half duplex’).
MR400ISO interface cards and ACKSYS RS485 external converters use the RTS signal to free the bus when the card is not transmitting. In idle state (no transmission), the line is in receive mode. As soon as a character or a group of characters must be sent, RTS is activated and the line switches to the transmission mode.

Error indication on ineffective DTR changes attempts

When the automatic turnaround is enabled, driving RTS signal has no effect anymore and the DTR is reserved in the driver. If the application tries to modify DTR, there won’t be any real effect. This checkbox indicates if the application must receive an error notification or not.

Default configuration parameters

WARNING : as well as for basic COM ports, these parameters are only used by some Windows programs. Always check the configuration parameters of your application.

DTR OFF on open

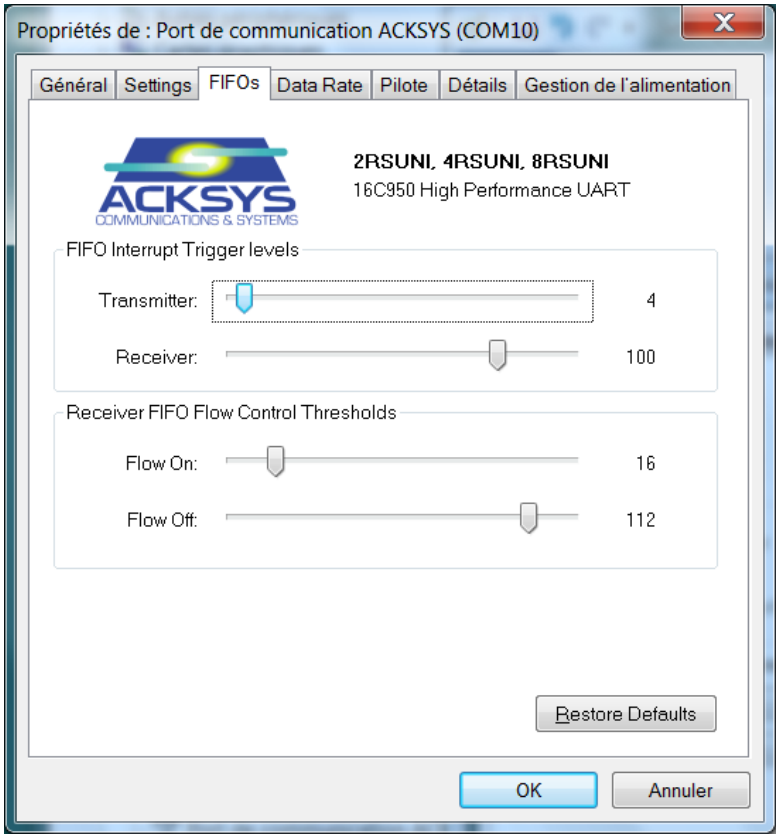
The default behavior for Windows serial drivers is to activate the DTR at the port’s opening. This might be annoying if the DTR is used to activate some functionality in the serial equipment, such as starting a radio MODEM. Checking this box leaves the DTR disabled until the application takes a deliberate action.

COM port number

The name of the COM port may be changed here.

The « FIFOs » tab

The **FIFOs** tab enables you to set the interrupt trigger levels for transmission and reception according to the number of characters in the respective buffers, as well as the flow control trigger levels. The default values are satisfactory for most traditional applications.



Configuring the interrupt trigger levels:

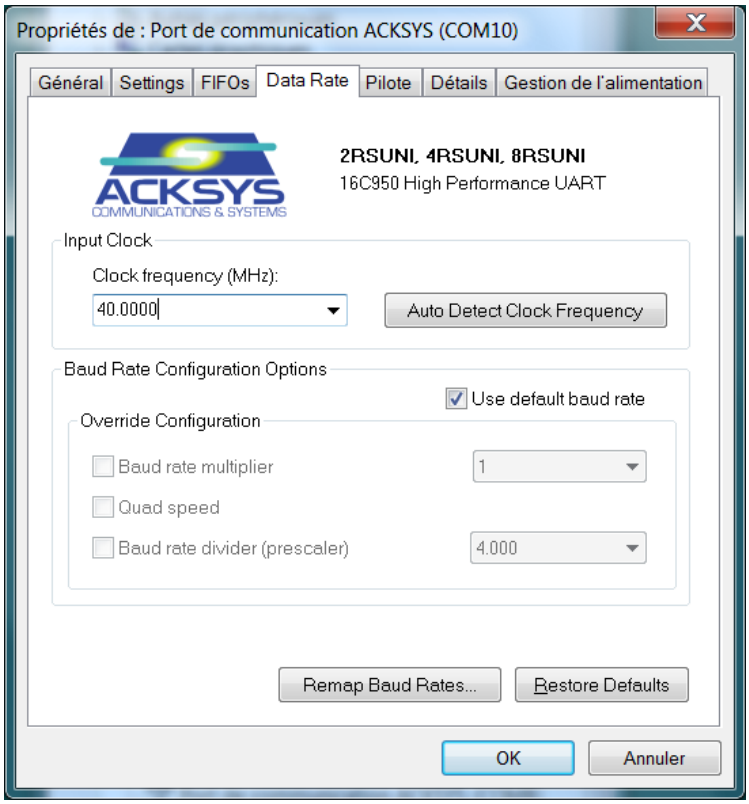
The value defined using the "Transmitter" cursor gives the level from which a transmitter interrupt will be generated. For example, the default value, 4, indicates that an interrupt will be generated as soon as the number of characters in the transmission buffer drops from 5 to 4. This value should remain low, but it may be advisable to increase it for higher speeds or with slow or overloaded CPUs.

The value defined using the "Receiver" cursor gives the level from which a receive interrupt will be generated. In the case of the default value, the interrupt occurs when the number of characters in the receive buffer rises from 63 to 64. If the number of characters received is less than the threshold, and does not change during a period corresponding to the time to transmit 4 characters, a time-out interrupt is generated to warn the peripheral's driver.

In the case of an application transferring large blocks of data, it is advisable to choose high thresholds in order to reduce the number of interrupts and thus the load on the CPU. However, it is not recommended that the maximum values be used, especially when communicating at high speed, in order to avoid reception overwrites and transmission interruptions.

The “Data rate” tab

This tab enables you to select the frequency of the oscillator for some specific cards. The default value is 24.4912 MHz.



- The ‘**Clock Frequency (MHz)**’ list allows you, if you know it, to directly specify the frequency of the embedded oscillator.
- The ‘**Auto detect Clock Frequency**’ button computes the most likely oscillator frequency.
- The ‘**Use default rate**’ checkbox should stay checked. Unchecked, it allows the user to specify rate multipliers. In this case, the real transmission speeds will be a multiple of the specified one.
- The ‘**Baud rate multiplier**’ option applies the multiplication factor to the speed chosen by the application. For example : With a factor of 16, when the application specifies a speed of 115200 bauds, the real speed will be 1,8435MHz.
- The ‘**Quad Speed**’ is reserved for specific applications and shall be left unchecked.
- The ‘**Baud rate divider (prescaler)**’ option divides the oscillator speed by a decimal factor which allows to reach a better precision for non standard speeds.
- The ‘**Remap Baud Rate**’ button allows to substitute a speed for another. For example, it is possible to make the 4800 bauds speed to really communicate at 230400 bauds. This is useful if your program only provides slow speeds although your serial devices allow higher ones.

5. INSTALLING THE ACKSYS DRIVER FOR LINUX KERNEL 2.2.X

The installation procedure has been tested initially using **Linux Mandrake** version **7.2** and version **2.2.17-21** of the kernel. If you have any compatibility problems with other versions of Linux, please contact ACKSYS.
All the programs described below can be found in linux/kernel22 subdirectory on the media supplied.

Installing the driver

This type of installation does not require Linux to be recompiled. In addition, it enables the driver to be loaded and unloaded dynamically.

Copy the file **srllinux.o** into the directory

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

Starting the driver

To install the driver, just enter **insmod srllinux**
To check that the driver has been correctly installed, use the **lsmod** command.
If the card's oscillator is not set to 29.4912 MHz, you must add the following option when you install the driver :

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

For example, for a 16 MHz oscillator, the command is as follows:

```
insmod srllinux input_clock=16000000
```

The next time Linux is started up, the driver will not run automatically – for this to happen, add the file **rc.ack** in the file **rc.serial**.

If the **rc.serial** file does not exist, create it, adding the line:

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

Then edit the file **rc.local**, adding the following line:

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

Creating the nodes

The nodes must be created in the system. To do this, edit the **msmknod** script supplied by ACKSYS, adjust the **MAXPORT** constant according to your card (if you have a 2-port card enter 2, if you have a 4-port card enter 4, etc.) and then run the script. It will create the peripheral files as follows:

Card number	*Peripheral number*
1	ttyM0-ttyM[MAXPORT-1]
2	ttyM[MAXPORT]-ttyM[2*MAXPORT-1]

Stopping the driver

To stop the driver, enter **rmmod srllinux**

6. INSTALLING THE ACKSYS DRIVER FOR LINUX KERNEL 2.4.X

The installation procedure has been tested initially using **Linux Mandrake** version **8.0** with kernel version **2.4.3-20mdk** and **Linux Redhat** with kernel version 2.4.7-10. If you have any compatibility problems with other versions of Linux, please contact ACKSYS. This driver is based on the Serial Linux Driver 5.05. All the programs described below can be found in linux/V3.4 subdirectory on the media supplied.

Installing the driver

This type of installation does not require Linux to be recompiled. In addition, it enables the driver to be loaded and unloaded dynamically.

Copy the file **`srlxrspci.o`** into the directory

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

The peripheral's files may be created with standard names (ttyS) or with a name chosen at installation time (e.g.: ttyA).

Starting the driver

To start the driver, type the following command :

```
insmod srlxrspci.o
```

To check that the driver has been correctly installed, use command **`lsmod`**.

This will create the tty devices in the system. By default, the driver create ttyA0 to ttyAn (n depends on your board).

If you want to change the tty name, you must pass as argument

```
tty_name_p=<tty name>
```

For example if you want to create the tty with name "ttys", start the driver with command :

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

To check the terminal name, see file "`/var/log/messages`".

- ✓ If the card's oscillator is not set to 29.4912 MHz, you must add the following option when you install the driver:

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

  For example, with a 16 MHz oscillator, the command is as follows :

  ```
  insmod srlxrspci input_clock=16000000
  ```

- ✓ If you need a custom speed, you must add the following option when you install the driver :

  ```
  speed_custom=[speed in bauds]
  ```

  For example, if you want a speed of 76800 bauds, the command is as follows:

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

  To configure the serial port speed, use these functions:

  ```
  int cfsetospeed(struct termios *termios_p, speed_t speed); //output
  speed
  int cfsetispeed(struct termios *termios_p, speed_t speed); // input
  speed
  ```

  If you want to use the parameter **`speed_custom`**, you must use the constant **`EXTA`**, as this example follows:

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

EXTA constant is defined by default in the file **/usr/include/bits/termios.h** in the following way:

```
#define EXTA B19200
```

If you want to use 19200 bauds (B19200), you must change the constant **EXTA** in **/usr/include/bits/termios.h** and compile Linux driver.

The next time Linux is started up, the module will not run automatically unless you add the file **rc.ack** in file **rc.serial**.

If the **rc.serial** file does not exist, create it, adding the line:

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

Then edit the file **rc.local**, adding the following line :

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

The procedure described above may differ for other versions of Linux.

Creating nodes

When the driver is running, use command mknod as follow to create nodes in the system :

```
mknod <tty name> c <major> <minor>
```

If the driver is started with default option :

```
mknod ttyA04 c 40 68
```

To check *major* and *minor* values, please see file "`/val/log/messages`".

Stopping the driver

To stop the module just enter **rmmod srlxrspci**

References

Linux help file relating to serial ports:
http://en.tldp.org/HOWTO/Serial-HOWTO.html

Linux help file relating to programming the serial ports:
http://en.tldp.org/HOWTO/Serial-Programming-HOWTO/

List of sites containing Linux Howtos:
http://metalab.unc.edu/LDP/mirrors.html

Source files for the Serial Linux driver 5.05:
http://sourceforge.net/projects/serial
http://sourceforge.net/project/showfiles.php?group_id=310

Using automatic line turnaround under Linux 2.4

In RS485 or RS422 slave mode, automatic line turnaround can be set. Use the following iocontrol code :

- ACKSYS_ENABLE_485_MODE: Validate the function of automatic line turn around.

Automatic line turnaround is handled directly by the UART.
No parameter is required for this iocontrol. It is sent to the driver using linux API function ioctl(...). The communication channel must be opened before calling this function.
These constants are defined in the file iocontrol.h.

BE CAREFUL: The value of the constant is not identical on the two Linux Kernel 2.2 and 2.4. It is necessary to recompile your application with the correct include file.

7. INSTALLING THE ACKSYS DRIVER FOR LINUX KERNELS 2.6.X

Prerequisites

You must have the following at hand to install the driver:

- The kernel headers.
- The "setserial" linux utility if special port setup is needed.
- Root login and password.
- One or more XRSPCI or XRSUNI cards.
- For kernels older than 2.6.25, you must also install the kernel sources.

Unpacking the archive

The archive is provided in compressed tar format. When unpacked it creates a subdirectory in the current directory.

That subdirectory contains in turn two directories:

- utils/ contains utilities to set up the ports,
- driver/ contains, for each kernel subversion, the driver sources.

Compiling for kernels 2.6.10 to 2.6.18

The installation procedure has been tested initially using the **Linux kernel 2.6.10** installed on a **Debian Sarge** distribution. This driver is based on the Linux serial driver which has been redesigned several times in kernel 2.6. You will find notes specific to Fedora Core 4 at the end of this chapter. Installing on Fedora Core 6 is similar.

A Linux driver module must be compiled with the knowledge of the kernel configuration on which it will be loaded. So, it is provided only as source code, and you must compile it before using. Once compiled, you may load the object module into any Linux kernel matching the configuration used during the compilation.

To compile it, you must:

- Install the kernel sources and headers
- Copy the acksys sources to the drivers/serial subdirectory of the kernel sources
- Edit the file drivers/serial/Makefile to add the line in bold at the indicated place:

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

- Follow the instructions given by the Linux distribution provider, or the instructions found in the kernel README. A typical kernel generation would begin with the following steps:

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

  The configuration step will create a file named ".config" holding all the configuration parameters. You must ensure that this file matches the configuration of the kernel in which the module will be loaded. If you have the configuration file of the current kernel, you can copy it to ".config" instead of running "make menuconfig". See next page.

On some OSes the current configuration is available in /boot/config (Debian, Mandrake). On others it will be in /lib/modules/`name –r`/include, or in /lib/modules/`uname –r`/build/include (RedHat…), or in /usr/src/linux/include, or in /proc/config.gz (SuSe)… You must check this by yourself.

- If you used the exact .config file matching your kernel, you can skip the kernel compilation and installation. Simply build the required modules:
  - ✓ Compile the kernel modules, for instance:
    ```
    make M=drivers/serial
    ```
  - ✓ Move the generated modules to the system modules directory:
    ```
    libmod=/lib/modules/`uname –r`/kernel/drivers/serial
    cp acksys8250.ko acksys8250_pci.ko $libmod
    depmod
    ```
- If you used a new .config file that you tailored to your needs, you must now create a full new kernel. This will automatically add the Acksys modules.
  - ✓ Compile the full kernel:
    ```
    make
    ```
  - ✓ Install the kernel and its modules:
    ```
    make install modules_install
    ```
  - ✓ Configure GRUB or LILO. Follow the indications given by the previous command.
  - ✓ If necessary for your distribution, create a initrd file (**xxx** is the kernel version):
    ```
    mkinitrd –o /boot/initrd.img-xxx xxx
    ```

Compiling for kernels 2.6.26 and later

The compilation will create an external loadable module. The installation procedure has been tested initially using the **Linux kernel 2.6.26** installed on a **Debian Lenny stable** distribution. This driver is based on the Linux serial driver which has been redesigned several times in kernel 2.6. If you have any compatibility problems with other versions of Linux, please first control against this Linux version.

WARNING: A Linux driver module must be compiled with the knowledge of the kernel configuration on which it will be loaded. So, it is provided only as source code, and you must compile it before using. Once compiled, you may load the object module into any Linux kernel matching the configuration used during the compilation.

To compile the driver:

- Install the kernel headers and compilation tools.
- Use a terminal command prompt.
- Go to the relevant driver sources directory depending on your kernel version.
- Start compiling with the command:
  ```
  make
  ```
- Install the generated module in the "`/lib/modules`" directory with:
  ```
  make install
  ```

Installing the driver

The driver can be loaded and unloaded dynamically. Load it using the following command:

```
modprobe acksys8250_pci
```

The ports are named /dev/ttyS<number>. <number> will depend on the kind of acksys cards, the number of ports on each card, and the order of the cards on the PCI bus. You can check the recognized ports with:

```
acksys8250_install.sh -p
```

If you ever need to stop the driver, enter these commands:

```
modprobe -r acksys8250_pci
```

Configuring the driver

✓ If you are using a card with a nonstandard oscillator (other than 29.4912 MHz), you must execute the following command for each port on this card:

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

Here are sample commands to handle some oscillator values on a 2RSPCI or 2RSUNI :

On-board oscillator	**Command to issue before use**
29.4916 MHz	None – this is the default
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*

✓ If you need to use a nonstandard baud rate, you must execute the following command on the port concerned:

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

This will replace the 38400 baud rate with the specified one. For example, if you want a speed of 76800 bauds with the 29.4912 MHz oscillator, the divisor is 29491200 / 16 / 76800 = 24, hence the command is as follows:

setserial /dev/ttyS100 divisor 24 spd_cust

After that, setting the 38400 baud rate in any way will really set 76800 bauds. There is no way to use 38400 bauds until you use the reverse command:

setserial /dev/ttyS100 spd_normal

✓ If you plan you connect any port on a RS485 bus, you must issue the following command:

acksys8250_install.sh –a /dev/ttySxxx

Using the driver

The driver is fully compatible with the basic linux serial driver provided in kernel version 2.6. For more information, see the following MAN pages: termios(3), ttyS(4), tty_ioctl(4), setserial(8).

To switch a port to RS485 automatic turnaround, you can either :

- Use the provided utility from an interactive or script shell, or from a compiled program using the system(3) call:

  acksys8250_install.sh –a /dev/ttySxxx

- Use the following ioctl: TIOCSERSETRS485

  Turn on RS485 turnaround:

```
#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");
```

  Turn off RS485 turnaround:

```
#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: Compiling the driver under Fedora Core 4

The following procedure is published with the authorization of RT2I. Translation by Acksys. Applies to the driver for Linux 2.6.10.

Installing the xRSPCI - xRSUNI driver for Linux 2.6

Install kernel sources (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
```

The sources are located in /usr/src/redhat/BUILD/kernel-2.6.11/linux-2.6.11

In the drivers/serial directory, copy the Acksys driver sources

Add the line in the Makefile (see Acksys documentation)

obj-$ (CONFIG_SERIAL_8250) += 8250.O $ (serial-8250-y)	existing line
obj-m += acksys8250.o acksys8250_pci.o	line to add
obj-$ (CONFIG_SERIAL_8250_CS) += serial_cs.o	existing line

change the source file "acksys8250.c", add after line 50: *(Acksys note: this seems unneeded for kernel 2.6.18)*

```
#define uart_match_port   ack_uart_match_port
```

compile the modules

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

if there are errors

```
# 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
```

install the modules

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

check if the module can be loaded:

```
# modprobe acksys8250_pci
```

(add this line in the file '/etc/rc.d/rc.serial' in order to load the module at boot time)

```
# lsmod
```

(check that theacksys8250 et acksys8250_pci modules are in the loaded modules list)

run the script 'acksys8250_install.sh', provided with the driver, to list the available ports

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

for a 2RSPCI you would have:

```
0000:03:02 2RSPCI ttyS100
0000:03:02 2RSPCI ttyS101
```

if the devices do not exist, create them with this command:

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

check once more that the ports exist with

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

check that the ports work as expected

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

8. APPENDIX A – LINE TURNAROUND

In multidrop network, RS485 or RS422 slave serial ports must have a TxD driver that can be disconnected from the transmission line when the serial port is not transmitting. In xRS cards range, this is implemented using the DTR control signal. The DTR line is connected to the RS485/RS422 driver enable such that setting the DTR line to a low (logic 0) state enables the RS485/RS422 driver. Setting the DTR line high (logic 1) puts the driver into the tristate condition. This in effect disconnects the driver from the bus, allowing other nodes to transmit over the same wire pair.

This mecanism, called line turnaroud, must be used in

RS485 two wires network for all nodes

RS422 four wires network for slaves nodes.

In RS422 with only two nodes or RS232, turnaround is not used.

9. APPENDIX B – TROUBLESHOOTING

The card is not detected by the BIOS

Check the PCI/PnP option in the motherboard set-up program, and set it to AUTO.
Check that the card is properly inserted into the slot.
Try other slots until you find one that works. If you cannot, try the card in another PC to verify its operation. If necessary, contact your PC manufacturer to obtain an updated BIOS.

The card is not detected by Windows 95/98/2000/ME/Xp

Check the first problem.
In the System properties window, check that the card has not already been recognized as a standard PCI card or a multi-function adapter card. If this is the case, delete the corresponding entry and click the Refresh button until the Add new hardware wizard starts up.
Reinstall Windows.

Communications between the card and your equipment do not work

Check the connection between your equipment and the card.
Check the communications parameters (speed, parity, number of stop bits, flow control) on each side.
In RS422/RS485 mode, the use of the +/- convention can cause problems. This is a faulty standard, with one manufacturer calling + what another will call -. In this case, you could try connecting the + signal to the – signal.
In RS485 or RS422 slave modes, check if the turnaround is correctly handled.

Contact ACKSYS

ACKSYS COMMUNICATIONS & SYSTEMS
ZA Val Joyeux
10, rue des entrepreneurs
78450 Villepreux - FRANCE

Telephone: +33 (0)1 30 56 46 46
Fax: +33 (0)1 30 56 12 95
Web: www.acksys.fr
Hotline: support@acksys.fr
Sales: sales@acksys.fr