xRSPCI

HARDWARE & SOFTWARE
USER MANUAL

xRSPCI

HARDWARE & SOFTWARE USER MANUAL

TRADE MARKS ®

- *ACKSYS* is a registered trade mark of *ACKSYS*.
- Windows XP, Windows 2000, Windows NT, Windows 95, Windows 98 and Windows ME are registered trade marks of Microsoft.

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

2, 4 OR 8 CHANNEL PCI PLUG & PLAY SERIAL COMMUNICATIONS CARD.

The XRSPCI card offers a reliable and powerful solution for communications applications in industrial environments requiring:

- high-speed serial communications links,
- RS232, RS422, or RS485 type interfaces

The XRSPCI card complies with the PCI 2.1 and higher specifications, making its installation very easy (no jumpers or switches for the interrupt level or base address). All the card's resources are automatically allocated by the PCI BIOS when the PC is switched on.

The card is supplied with drivers for Windows 98, Windows ME, Windows NT, Windows 2000 and Linux (kernel 2.2 and 2.4). Other drivers may be developed on request.

Several XRSPCI cards may be used at once within the same machine, enabling 8, 16, 24 or 32 channel configurations.

2. CHARACTERISTICS OF XRSPCI RANGE CARDS

2.1 Characteristics common to the different versions

- 2, 4 or 8 RS232, RS422 or RS285 asynchronous serial communications ports
- The UARTS register set is compatible with 16C550.
- Maximum communication speed:
  - Using RS232, guaranteed up to **250 Kbits/s**, tested up to **460 Kbits/s**
  - Using RS422/485 guaranteed and tested up to **1.8 Mbits/s**
- PCI slave 2.1 or higher interface.
- Polling registers (image of the UARTs' eight interrupt lines)
- Identification of particular card configurations by reading a register
- 4 base addresses, BAR0 to BAR3
  - BAR0: UART base address in I/O
  - BAR1: UART base address in memory
  - BAR2: Configuration registers base address in I/O
  - BAR3: Configuration registers base address in memory

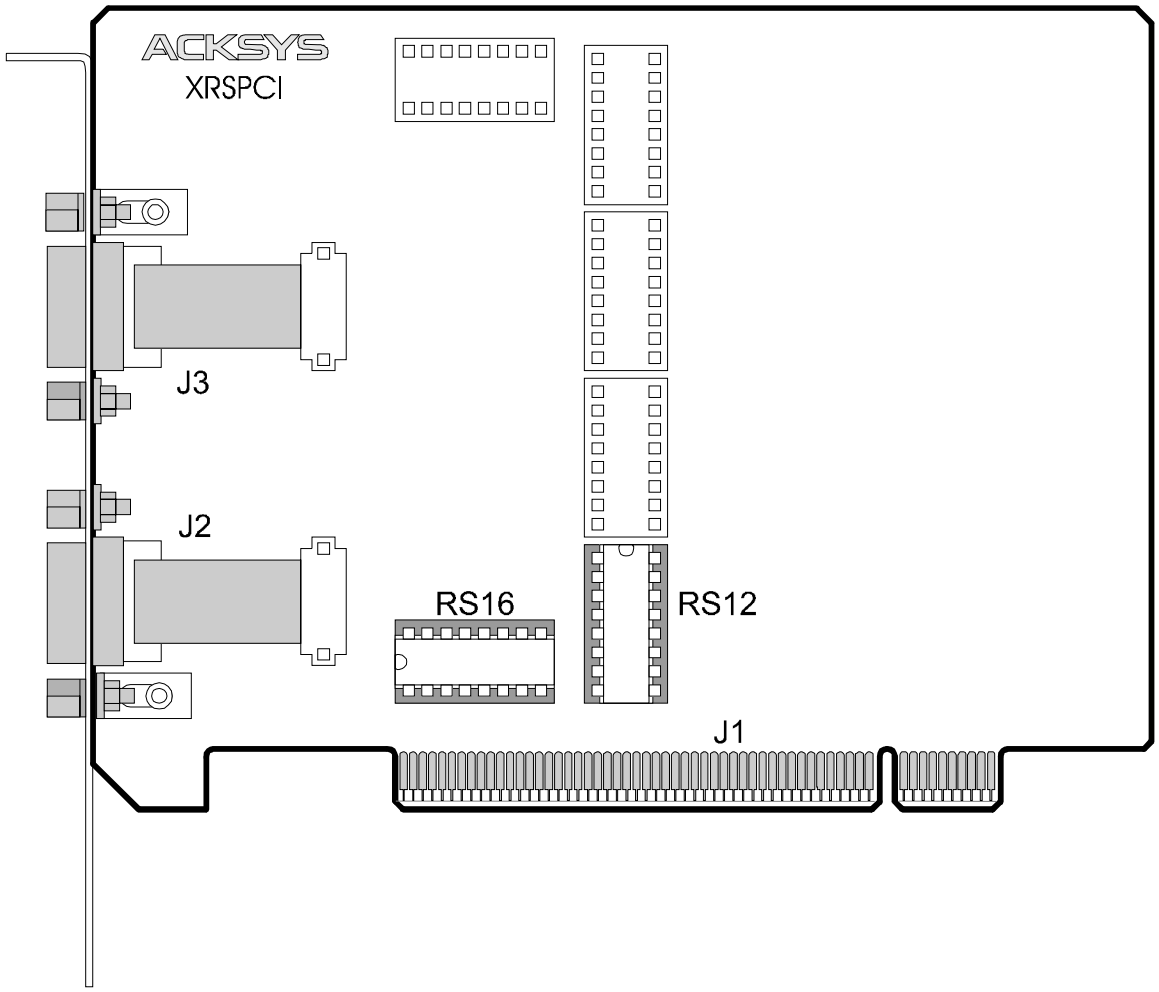
  A detailed description of the configuration registers is available on request for customers who wish to develop drivers.

2.2 Advanced communications functions

- Up to 128-byte FIFO per port, for sending and receiving.
- PCI interface integrated into the UARTs, optimising data transfer.
- RS485 mode with automatic turnaround of line drivers.
- Automatic control of equipment/software flow.
- Advanced baud generator for non-standard speeds.
- Transmission speed of up to 1.8 Mbps (RS422 interface) with standard 29.4912 MHz oscillator.
- ± 15kV ESD protection on each port.

2.3 2RSPCI connectors/relays layout

Relays can only be mounted on RS422/485 versions of the card.

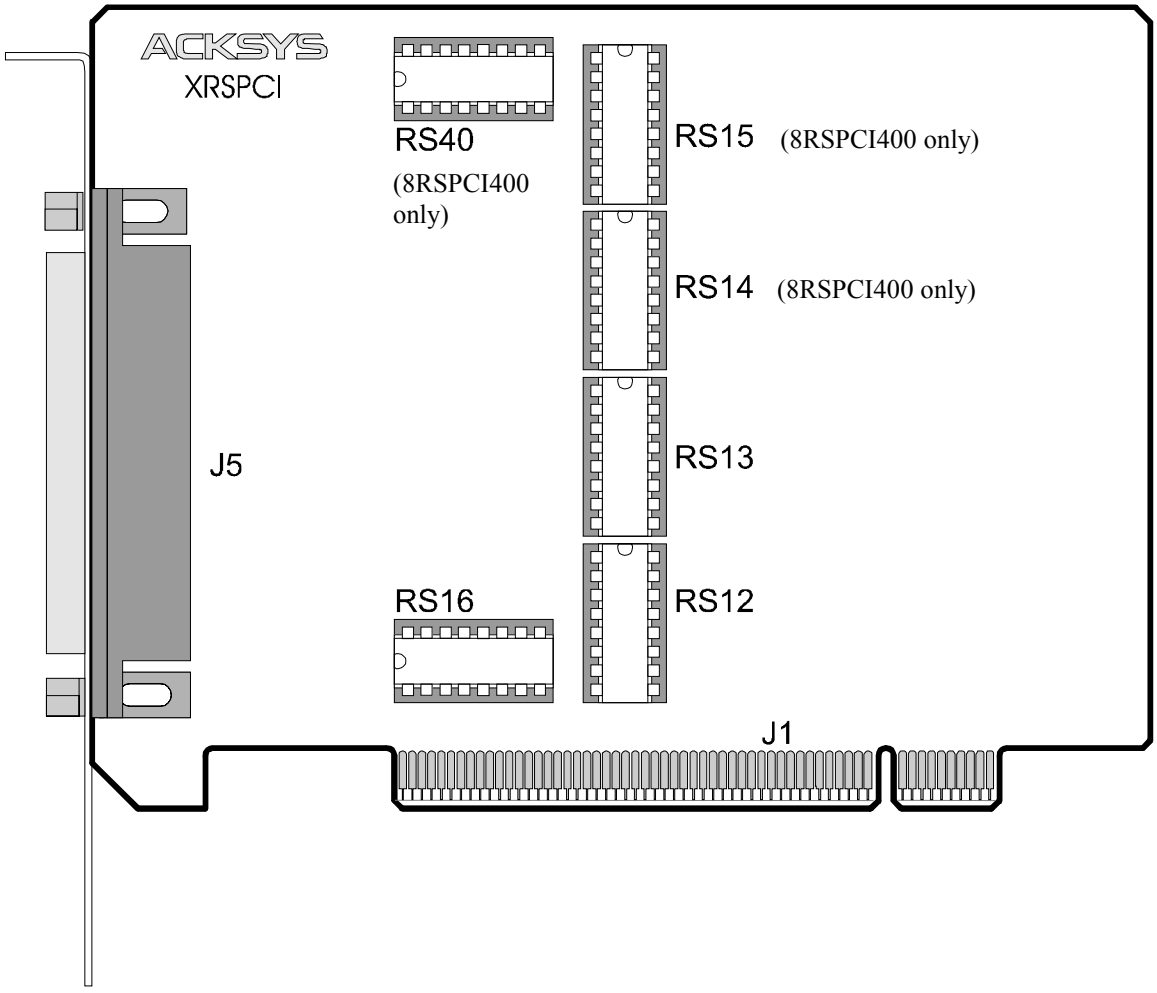


J1: PCI connector (5V environment only)
J2: SUB D 9-pin connector in port 1
J3: SUB D 9-pin connector in port 2

2RSPCI232 connectors : male
2RSPCI400 connectors : female

2.4 4RSPCI & 8RSPCI connectors/relays layout

Relays can only be mounted on RS422/485 versions of the card.

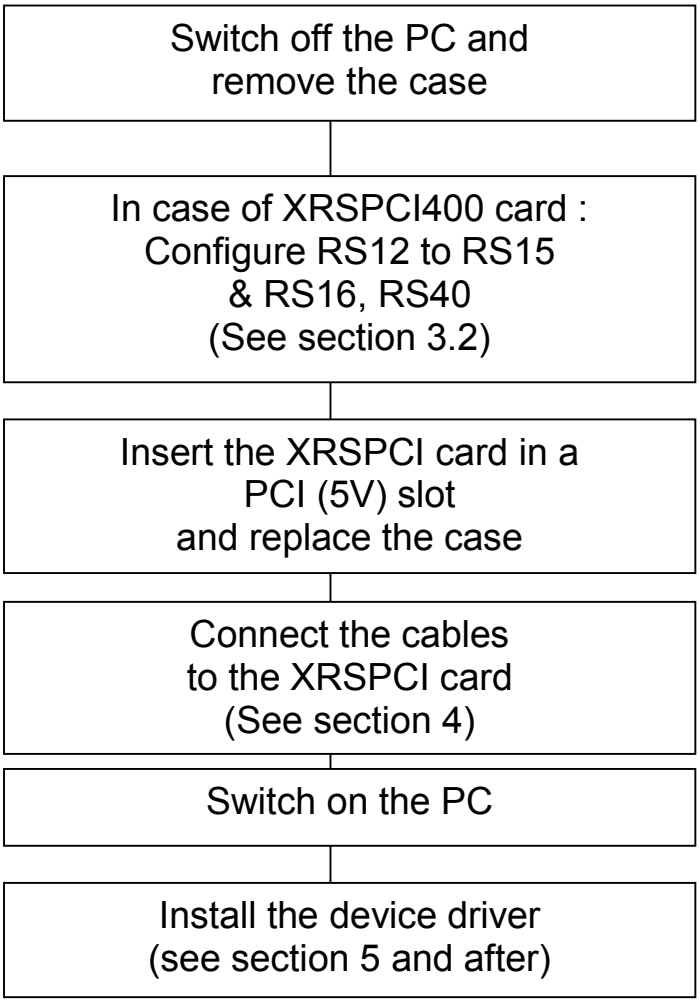


J1: PCI connector (5V environment only)
J5: 62-pin female connector for ACKSYS octopus cable or customer cable

Octopus cable supplied with 4RSPCI232 card : ref 4P232
4 SUB D 9-pin male connectors
Octopus cable supplied with 4RSPCI400 card : ref 4P422
4 SUB D 25-pin male connectors
Octopus cable supplied with 8RSPCI232 card : ref 8P232
8 SUB D 9-pin male connectors
Octopus cable supplied with 8RSPCI400 card : ref 8P422
8 SUB D 25-pin male connectors

3. INSTALLING THE CARD

3.1 General synopsis



In the event of problems, refer to the **TROUBLESHOOTING** section.

3.2 Relays configuration

This section concerns only xRSPCI400 cards.
The use of the RS422 or RS485 mode, termination and polarisation resistors are determined by the presence or absence of relays mounted on the card, as shown in tables 3.1, 3.2 and 3.3 below.

Selecting RS422 or RS485 Mode

Thanks to RS12 to RS15 relays, each peer of channels is configurable in RS422 or RS485.

		2RSPCI400		4RSPCI400		8RSPCI400	
Relay	**Ports**	**RS422**	**RS485**	**RS422**	**RS485**	**RS422**	**RS485**
RS12	**1 & 2**	○	●	○	●	○	●
RS13	**3 & 4**	-	-	○	●	○	●
RS14	**5 & 6**	-	-	-	-	○	●
RS15	**7 & 8**	-	-	-	-	○	●

Table 3.1 : RS12 to RS15 relays mounting by card type

Selecting Termination & polarisation resistances

		2RSPCI400 / 4RSPCI400 /8RSPCI400	
Relay	**Ports**	**Without polarisation & Termination**	**With polarisation & Termination**
RS16	**1, 2, 3 & 4**	○	●

Table 3.2 : RS16 relay mounting

		8RSPCI400	
Relay	**Ports**	**Without polarisation & Termination**	**With polarisation & Termination**
RS40	**5, 6, 7 & 8**	○	●

Table 3.3 : RS40 relay mounting

○ Relay absent ● Relay present - Not applicable

4. CABLES

4.1 RS232 mode

Table 4.1 below gives the signal distribution for the 9-pins male connector in RS232 mode.

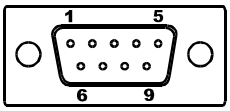
2RSPCI232 card **4RSPCI232 card with 4P232 octopus cable** **8RSPCI232 card with 8P232 octopus cable**			
SUB D 9 male	**Pin**	**Signal**	**Function**
 EIA/TIA 574 DTE	1	DCD (109)	Data Carrier Detect
	2	RXD (104)	Received Data
	3	TXD (103)	Transmitted data
	4	DTR (108)	Data Terminal Ready
	5	GND (102)	Ground
	6	DSR (107)	Data Set Ready
	7	RTS (105)	Request To Send
	8	CTS (106)	Clear To send
	9	RI* (125)	Ring Indicator

Table 4.1 : RS232 signals distribution for 9-pin male connector

*** NOTE**: THE RI (125) SIGNAL ONLY EXISTS ON 2RSPCI232 CARDS (2 PORTS). PIN 9 OF THE OCTOPUS 4P232 and 8P232 CONNECTOR IS NOT CONNECTED.

4.2 RS422 mode

Table 4.2 below gives the signal distribution for the 25-pins male connector of the 4P422 or 8P422 octopus cable in RS422 mode.
Table 4.3 below gives the signal distribution for the 9-pins female connector of the 2RSPCI400 in RS422 mode.
Pins not shown in the table are not used.

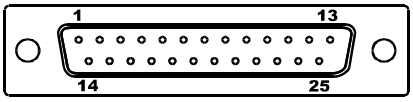
4RSPCI400 card in RS422 mode with 4P422 octopus cable **4RSPCI400 card in RS422 mode with 8P422 octopus cable**			
SUBD 25 male	**Pin**	**Signal**	**Function**
	2	A	Transmit (+Tx)
	3	A’	Receive (+Rx)
	7	GND	Ground
	9	-POL	Polarisation
	14	B	Transmit (-Tx)
	16	B’	Receive (-Rx)
	19	+POL	Polarisation

Table 4.2 : RS422 signals distribution for 25-pins male connector

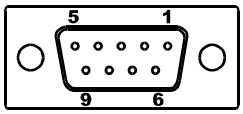
2RSPCI400 card in RS422 mode			
SUBD 9 female	**Pin**	**Signal**	**Function**
	1	-POL	Polarisation
	2	A’	Receive (+Rx)
	3	A	Transmit (+Tx)
	4	+POL	Polarisation
	5	GND	Ground
	7	B	Transmit (-Tx)
	8	B’	Receive (-Rx)

Table 4.3 : RS422 signals distribution for 9-pins female connector

4.3 RS485 mode

Table 4.4 below gives the signal distribution for the 25-pins male connector of the 4P422 or 8P422 octopus cable in RS485 mode.
Table 4.5 below gives the signal distribution for the 9-pins female connector of the 2RSPCI400 in RS485 mode.
Pins not shown in the table are not used.

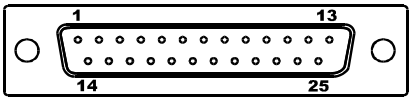
4RSPCI400 card in RS485 mode with 4P422 octopus cable **4RSPCI400 card in RS485 mode with 8P422 octopus cable**			
SUBD 25 male	**Pin**	**Signal**	**Function**
	2	Reserved	
	3	AA’	Transmit/Receive (+Tx/Rx)
	7	GND	Ground
	9	-POL	Polarisation
	14	Reserved	
	16	BB’	Transmit/Receive (-Tx/Rx)
	19	+POL	Polarisation

Table 4.4 : RS485 signals distribution for 25-pins male connector

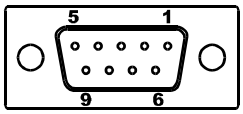
2RSPCI400 card in RS485 mode			
SUBD 9 female	**Pin**	**Signal**	**Function**
	1	-POL	Polarisation
	2	AA’	Transmit/Receive (+Tx/Rx)
	3	Reserved	
	4	+POL	Polarisation
	5	GND	Masse
	7	Reserved	
	8	BB’	Transmit/Receive (-Tx/Rx)

Table 4.5 : RS485 signals distribution for 9-pins female connector

4.4 Cabling recommendations in RS232 mode

In RS232 mode, the 9-pins male connector identifies a standard (EIA/TIA 574) Data Terminal Equipment (DTE) (except RI signal for 4P232 and 8P232 octopus cable).

- To connect them to another DTE peripheral, use a crossover (null-modem) cable
- To connect them to a DCE cable, use a straight cable

4.5 Cabling recommendations in RS422/RS485 mode

Cabling recommendations

To avoid any cable inversion caused by using the non-standard + and – convention (one manufacturer will describe as + what another will describe as -), make up your cable as follows:

In RS422 mode :

Identify A, A', B & B' signals equipment side
The points A, B, A' and B' are as defined in the EIA-422 and V11 recommendations, such that: $V_A < V_B$ and $V_{A'} < V_{B'}$ when idle, state also called MARK or OFF (Transmission/Reception of stop bits).
signal A of the xRSPCI to signal A' of the equipment
signal B of the xRSPCI to signal B' of the equipment
signal A' of the xRSPCI to signal A of the equipment
signal B' of the xRSPCI to signal B of the equipment

In RS485 mode :

Identify AA'& BB' signals equipment side
The points AA' and BB' are as defined in the EIA-485 and V11 recommendations, such that: $V_{AA'} < V_{BB'}$ when idle, state also called MARK or OFF (Transmission/Reception of stop bits).
signal AA' of the xRSPCI to signal AA' of the equipment
signal BB' of the xRSPCI to signal BB' of the equipment

Polarisation & termination resistances :

Polarisation and termination resistances are integrated into all RS422/485 xRSPCI cards and connected by on-board mounted relays

Line polarisation may be necessary in RS422 mode in a multipoint master/slave set-up (also called RS485 4 wire). A single polarisation must be present on the bus.

Line polarisation may be necessary in RS485 mode. A single polarisation must be present on the bus.

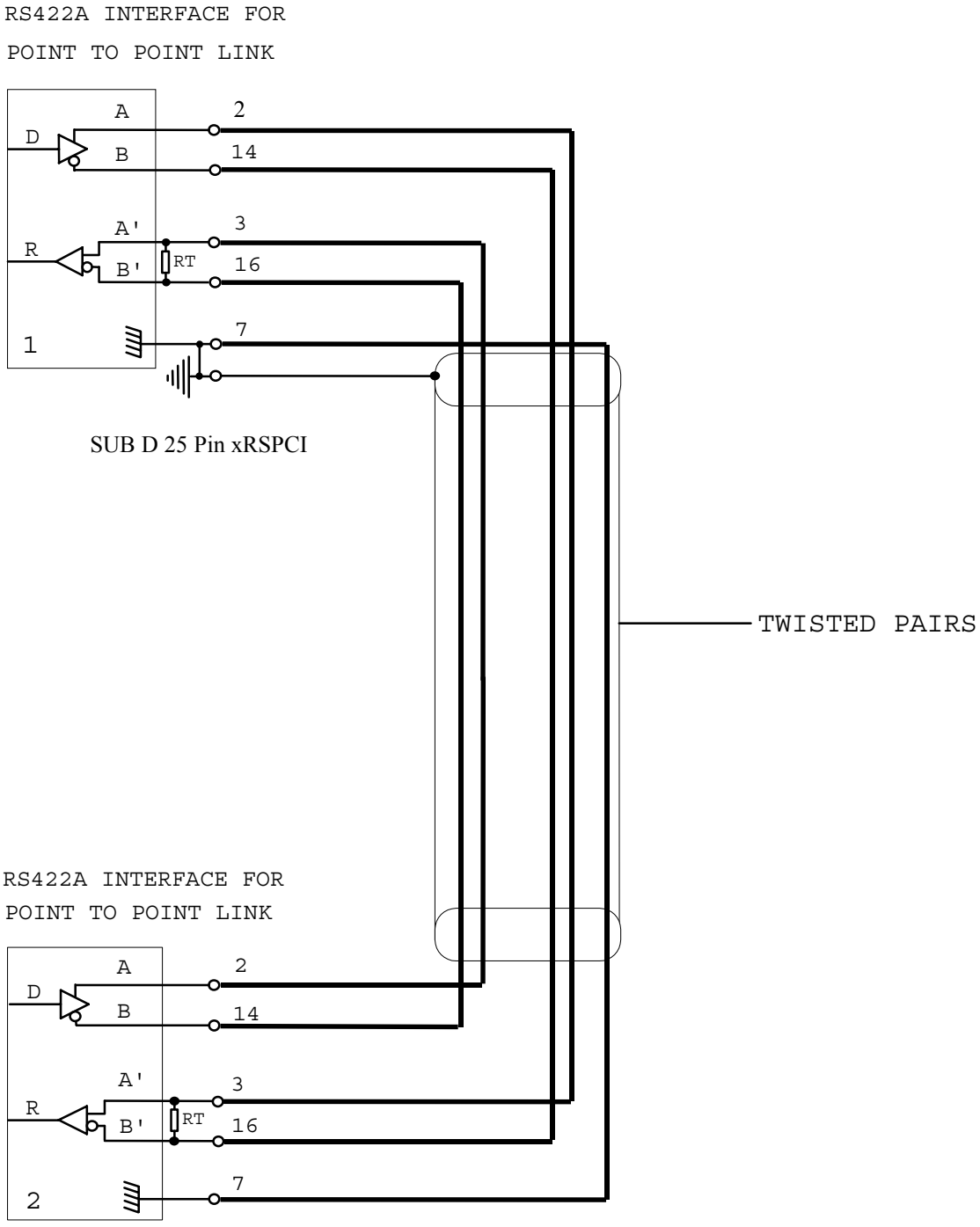
Termination resistances enable reflection on the line to be limited when the distances between equipment are significant.

If polarisation is only required on certain ports, or if the presence of termination resistances poses a problem, it is possible to polarise lines individually by looping back in the connector. In RS422, to polarise the reception line of a port, link the **B'** signal to the **+POL** signal and the **A'** signal to the **–POL** signal in the connector. The relay corresponding to these ports must be removed.

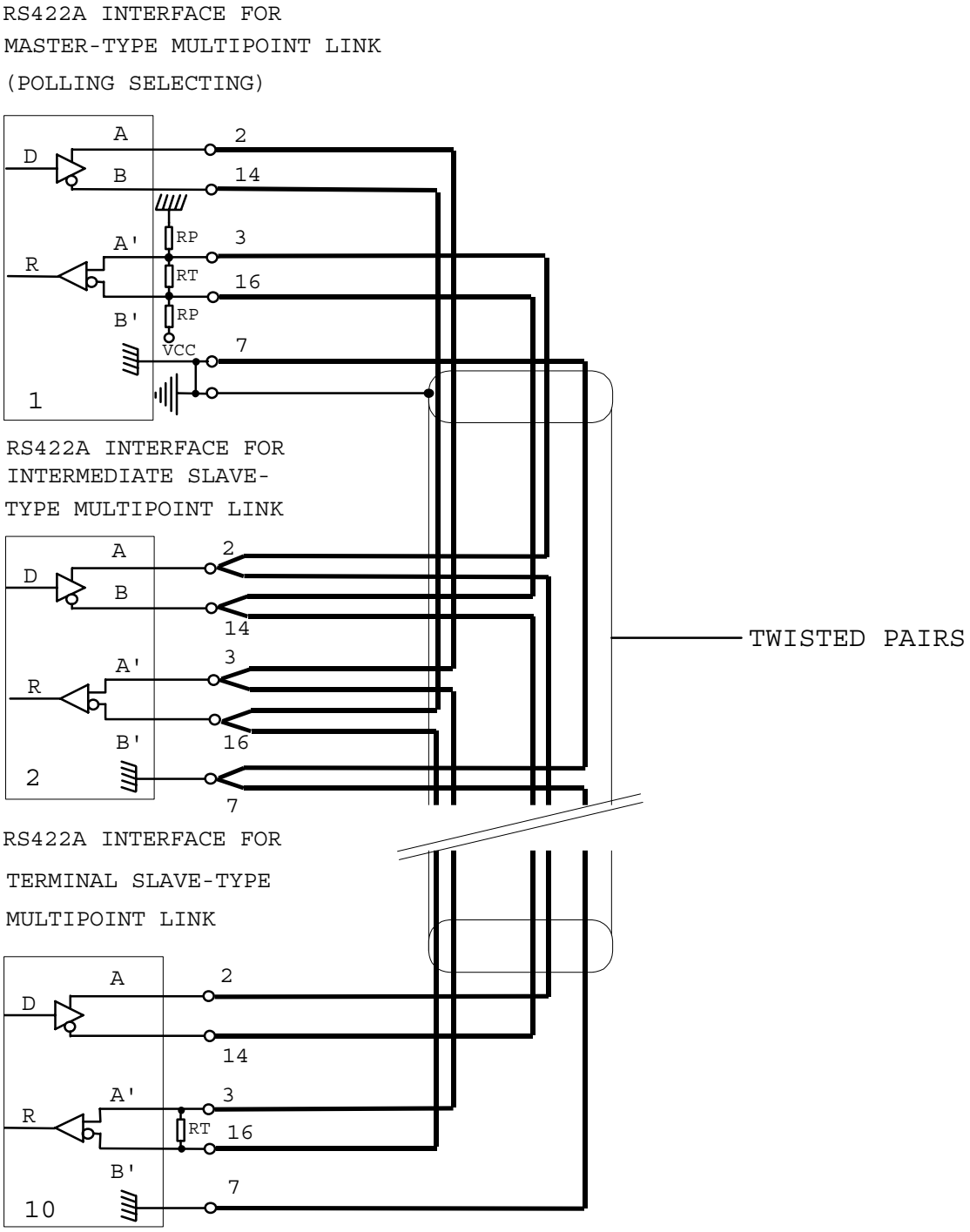
In RS485, to polarise the reception line of a port, link the **BB'** signal to the **+POL** signal and the **AA'** signal to the **–POL** signal in the connector. The relay corresponding to these ports must be removed.

4.5.1 Cabling example in RS422

RS422A FULL-DUPLEX POINT TO POINT CABLING

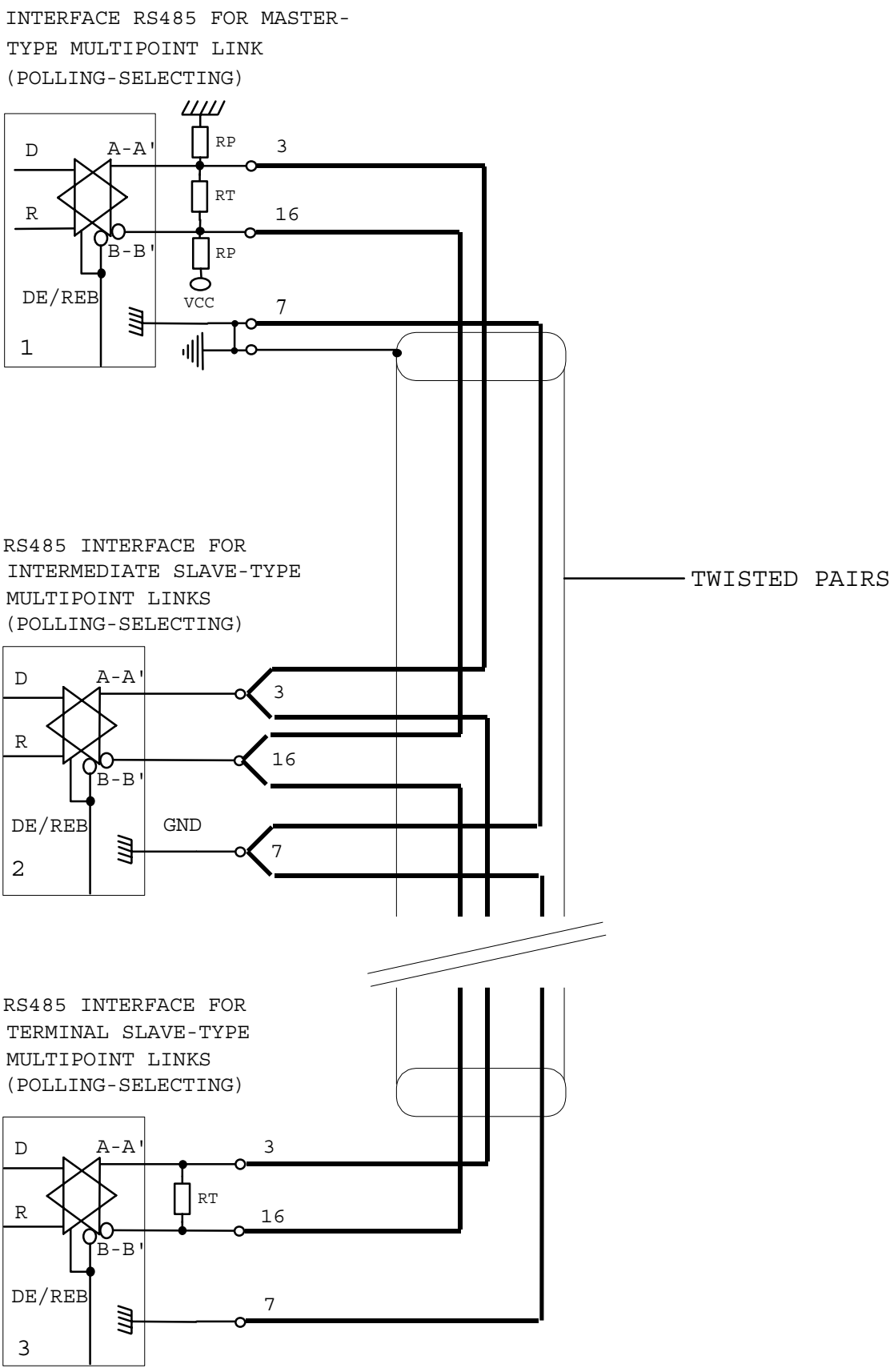


RS422A FULL-DUPLEX MULTIPOINT CABLING



4.5.2 Cabling example in RS485

RS485 HALF-DUPLEX MULTIPOINT CABLING



5. INSTALLING THE ACKSYS DRIVER FOR WINDOWS 98 & ME

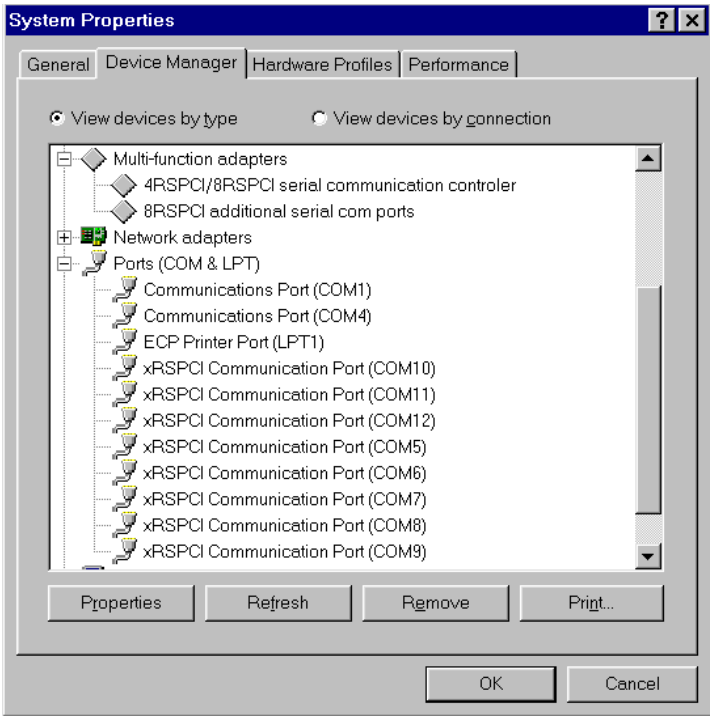
Install the XRSPCI card in the PC (see previous section) and start up Windows.
As this is a PCI card, the configuration will be automatically updated whenever the card is installed or removed.

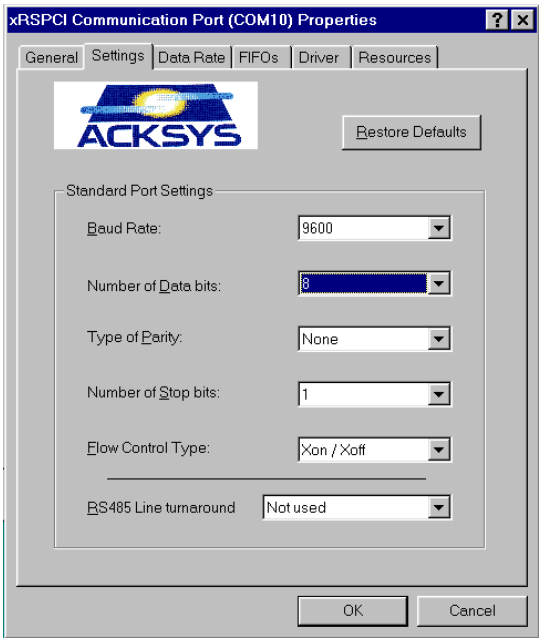
Under Windows 98, 98 SE and ME, the *Add new hardware* wizard will automatically run the first time the card is installed, as soon as the card is detected. Follow the wizard instructions .

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.

5.1 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

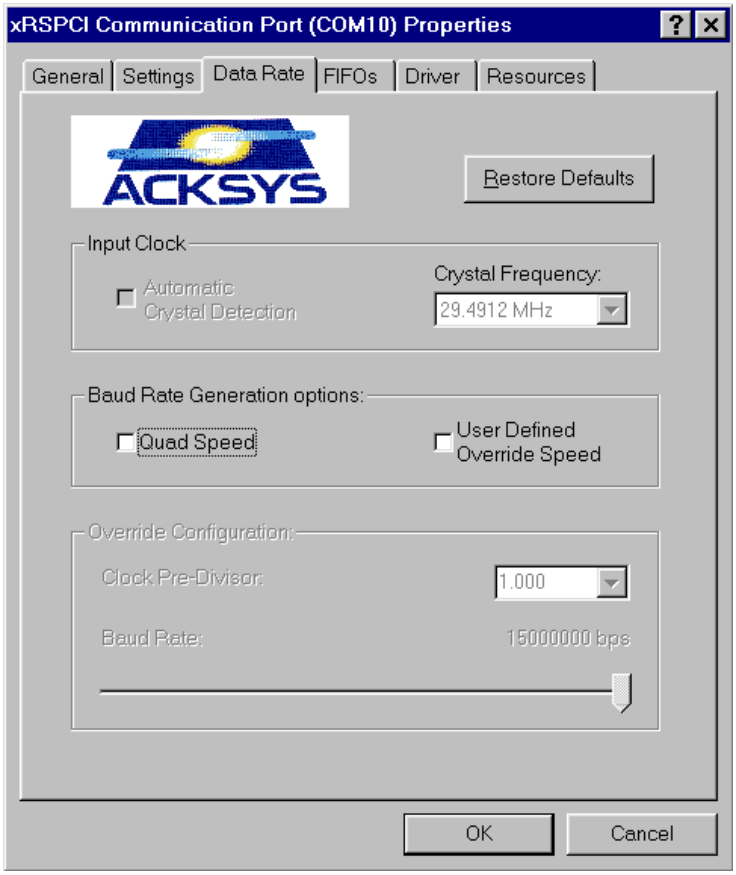
This refers to driving the direction of communication in RS485. If the port is configured as RS422 or RS232 (pin to pin links), simultaneous transmission and reception ("full duplex") are enabled, and so the communication direction does not need to be driven. The selected option must be: "**Not used**" – transmission is therefore always enabled.

In RS485 mode, transmission and reception alternate ("half duplex") and it is therefore necessary to drive a signal indicating whether the line should be transmitting or receiving. When idle (no transactions) the line is receiving; as soon as a character or group of characters needs to be transmitted, the line must be switched into transmission mode: the xRSPCI card uses the DTR signal to carry out this function.

When the "**Driven by application**" option is selected, the application is in charge of controlling the DTR signal. The signal must be activated before transmission and deactivated when the last character has been transmitted. This method does not allow precise control of the switching time after a transmission is complete.

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 switching time for the communication direction.

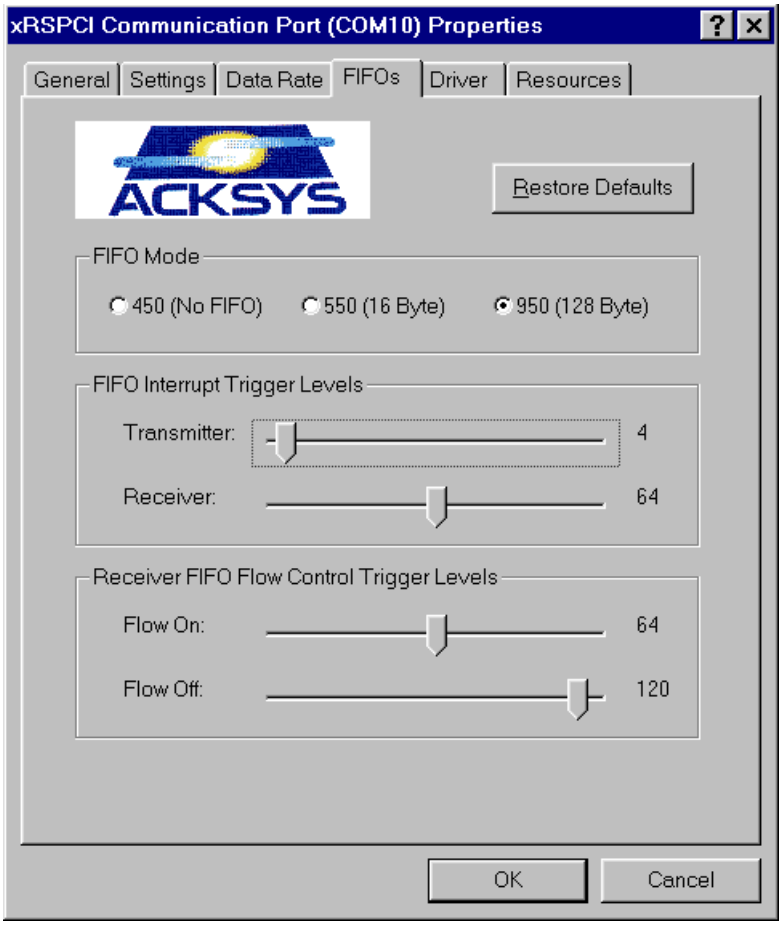
Note: *Driving the communication direction can be necessary in* ***RS422*** *mode, when several pieces of equipment are connected to a bus (****master/slave RS422 or RS485 4 wire****). If the port of the xRSPCI card is a "slave" peripheral, it shares its transmission line with the other "slave" peripherals in the network, and must therefore leave it in a state of "high impedance" when it is not transmitting.*



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 xRSPCI 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 xRSPCI 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.

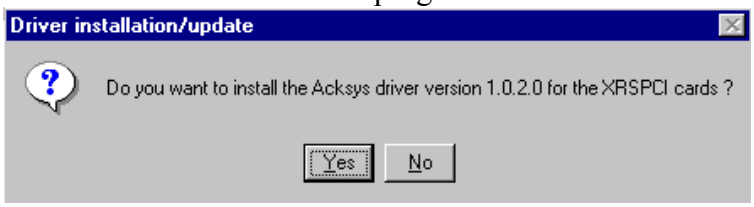
In the case of an application transferring large blocks of data, it is advisable to choose high thresholds in order to reduce the number of interrupts and thus the load on the CPU. However, it is not recommended that the maximum values be used, especially when communicating at high speed, in order to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

6. INSTALLING THE ACKSYS DRIVER FOR WINDOWS NT 4.0

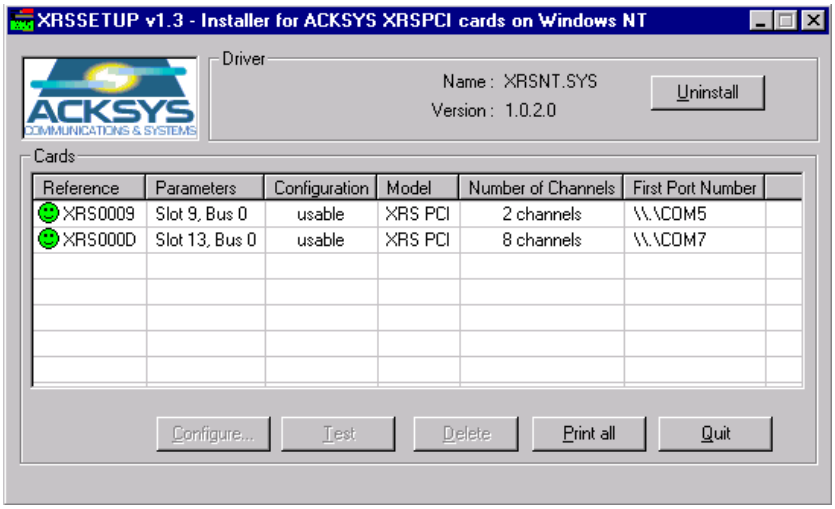
Install the xRSPCI card (see previous section) and start up Windows NT.

To install the driver:

Start your PC and identify yourself as *Administrator*
Insert the xRSPCI driver disk[1]
Run the XRSSETUP program from A:\WINNT



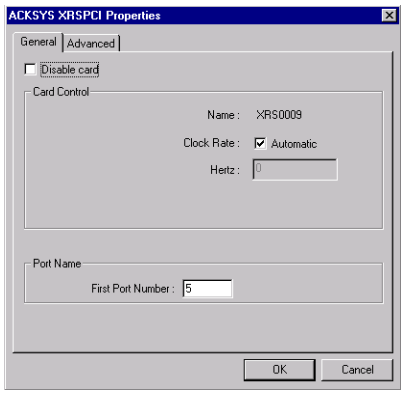
Click the ***Yes*** button

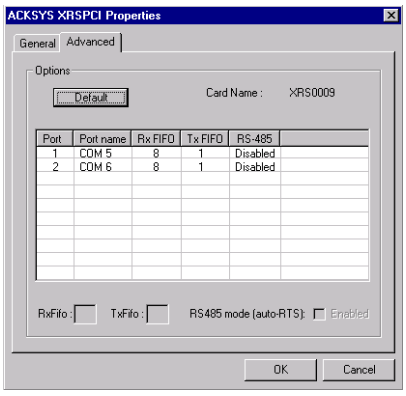


In this window, you will see a list of xRSPCI cards detected by the driver.
You can either click **Exit** to finish the installation or select a card and click **Configure** to open the *Properties of ACKSYS xRSPCI* window.

In the *Properties of ACKSYS XRSPCI* window:

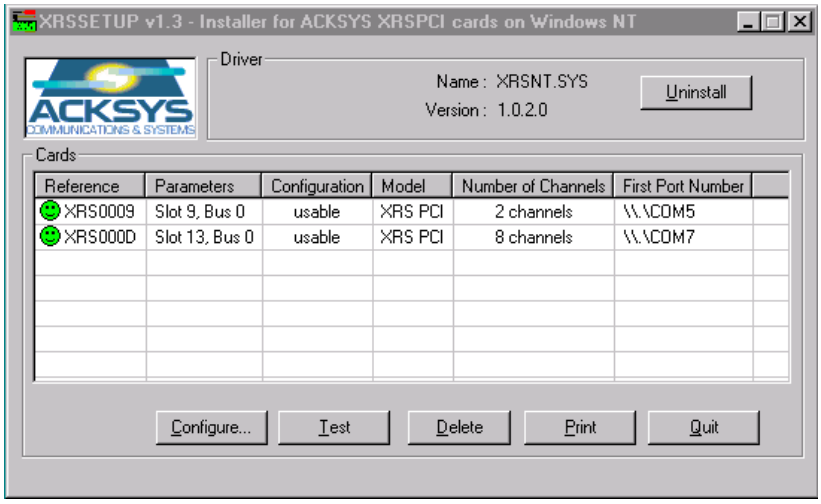
You can deactivate a card by clicking on the *Disable the driver of this card* box.
For each port, you can specify the RxFIFO interrupt trigger level, the size of the TxFIFO buffer and the RS485 mode.



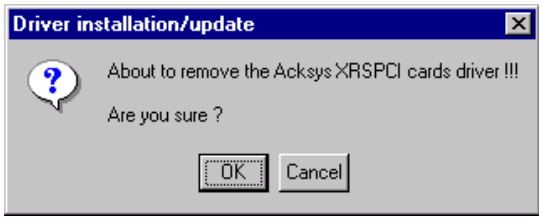


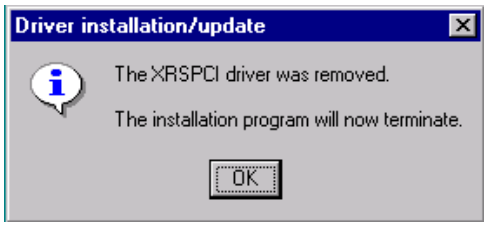
[1] The drivers may be supplied on CD. In this case, you will need to select the folder containing the xRSPCI drivers.

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



CLICK Uninstall



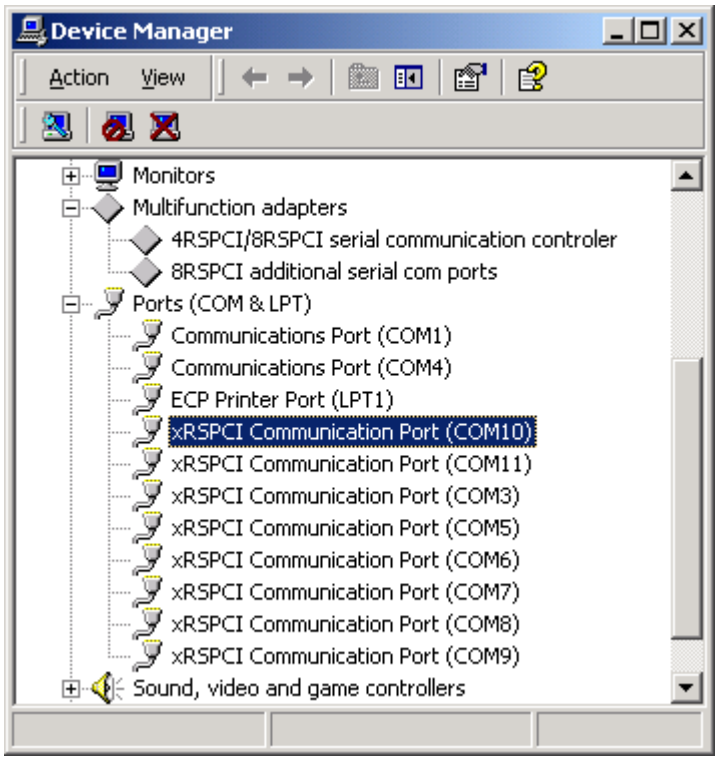


7. INSTALLING THE ACKSYS DRIVER FOR WINDOWS 2000/XP

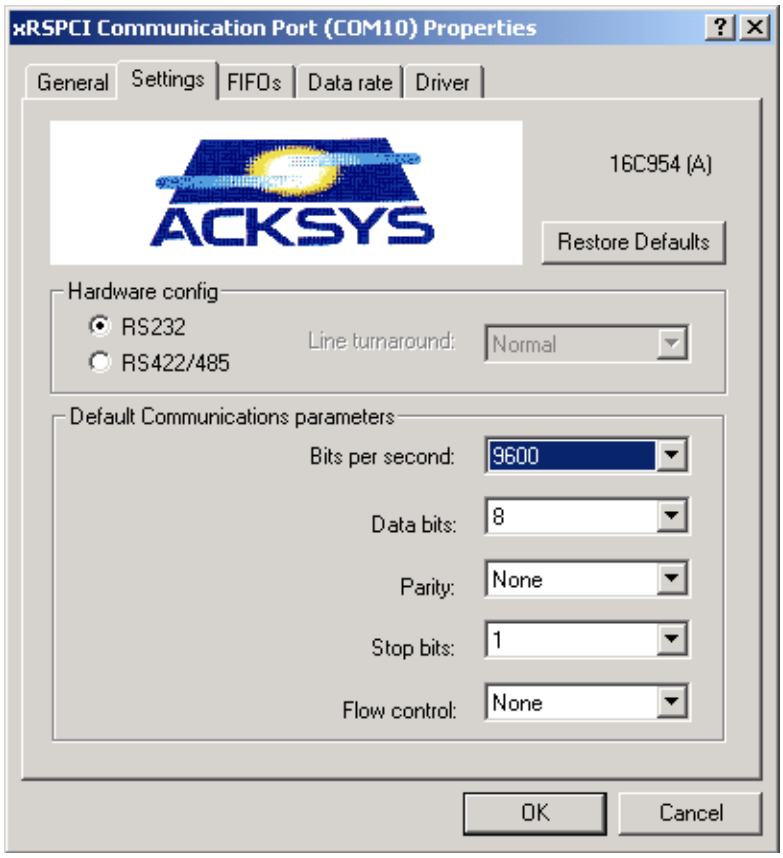
As with Windows 9x or ME, Windows 2000 automatically detects the XRSPCI card. A hardware installation wizard is automatically run when the system starts up, as soon as the card is detected. Follow the wizard instructions .
NOTE FOR WINDOWS XP : The wizard will announce you that the driver is not certified. Ignore this message in order to continue the installation of the card.

7.1 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.



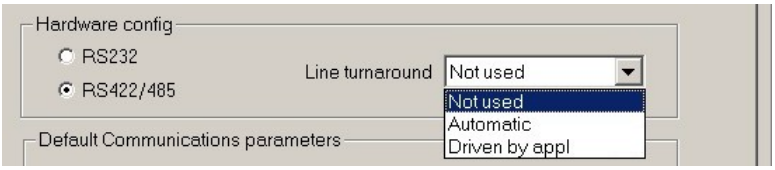
The "**Settings**" tab enables the default communications parameters to be defined, as with a standard COM port. It can also be used to define the interface type and automatic line turnaround for RS485.

Hardware config

RS232: In this mode, the control of hardware flow and the positioning of control signals are enabled by the peripheral driver. Automatic line turnaround is forbidden.

RS422/485: In this mode, the control of hardware flow is forbidden. Automatic line turnaround can be programmed.

Line turnaround



This refers to driving the direction of communication in RS485. If the port is configured as RS422 (pin to pin links), simultaneous transmission and reception ("full duplex") are enabled,

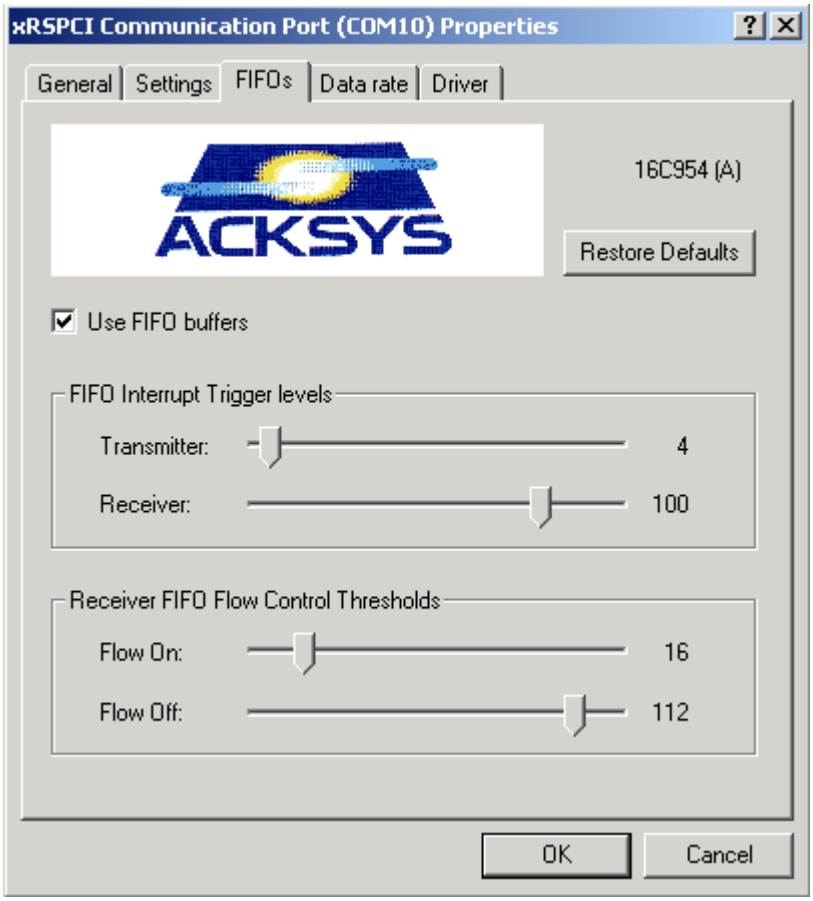
and so the communication direction does not need to be driven. The selected option must be: "**Not used**" – transmission is therefore always enabled.

In RS485 mode, transmission and reception alternate ("half duplex") and it is therefore necessary to drive a signal indicating whether the line should be transmitting or receiving. When idle (no transactions) the line is receiving; as soon as a character or group of characters needs to be transmitted, the line must be switched into transmission mode: the xRSPCI card uses the DTR signal to carry out this function.

When the "**Driven by application**" option is selected, the application is in charge of controlling the DTR signal. The signal must be activated before transmission and deactivated when the last character has been transmitted. This method does not allow precise control of the switching time after a transmission is complete.

For more efficient control, or if the application cannot manage the DTR signal, select the "**Automatic**" option: the DTR signal will now be activated automatically by the UART whenever a character is transmitted, guaranteeing an optimal switching time for the communication direction.

Note: *Driving the communication direction can be necessary in* ***RS422*** *mode, when several pieces of equipment are connected to a bus (****master/slave RS422 or RS485 4 wire****). If the port of the xRSPCI card is a "slave" peripheral, it shares its transmission line with the other "slave" peripherals in the network, and must therefore leave it in a state of "high impedance" when it is not transmitting.*



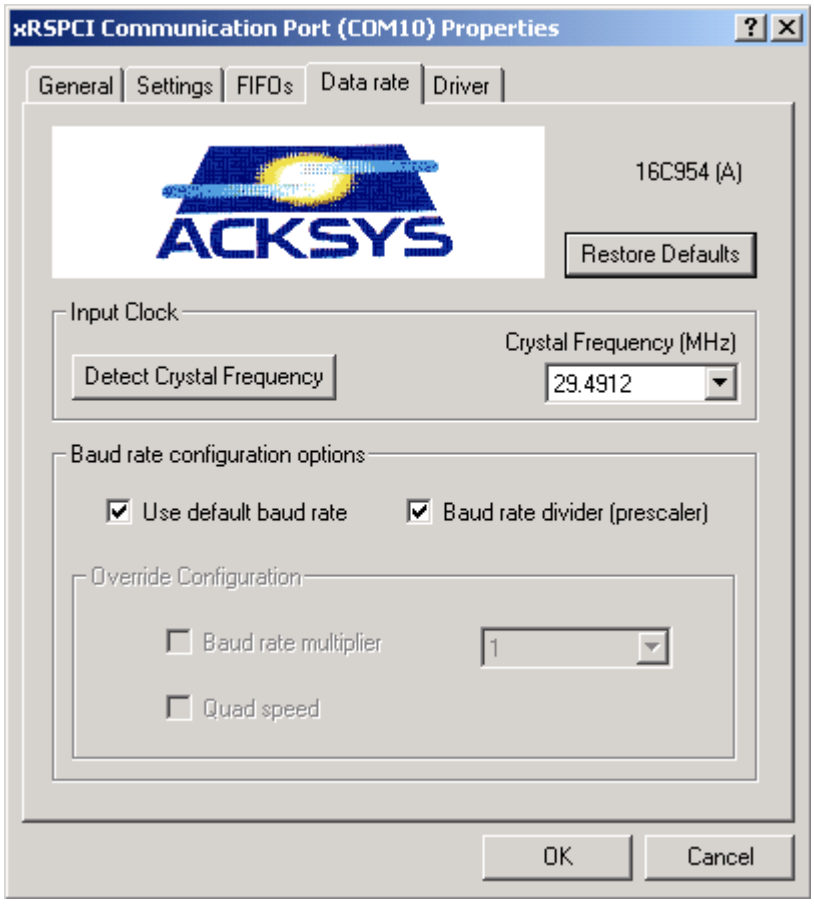
The **FIFOs** tab enables you to set the interrupt trigger levels for transmission and reception according to the number of characters in the respective buffers, as well as the flow control trigger levels. The default values are satisfactory for most traditional applications.

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 enables you to select the frequency of the oscillator for certain specific models of the xRSPCI cards. The default value is 24.4912 MHz.

- '**Use default baud rate**' option: This should remain selected. It indicates that the communications speed is fixed, as normal, by the Win32 API.

- '**Baud rate divider (prescaler)**' option: this option is active by default, and enables the communications speed to be calculated using a decimal prescaler, enabling better precision for non-standard speeds to be obtained. In this case, the xRSPCI driver calculates parameters that give the best precision in terms of the requested speed. When this option is not selected, the rounding of non-standard speeds is compatible with older cards.

- '**Baud rate multiplier**' option: this functionality cannot be selected when the "Use default baud rate" option is ticked. It enables the multiplication factor chosen in the selection box to be applied to the speed requested by the application. For example, with a factor of 16, when the application requests a speed of 115200 baud, the effective speed will be 1.8432 MHz, thus enabling an application which would otherwise be limited to 115200 baud to work at higher speeds.

- '**Quad Speed**' option: this functionality cannot be selected when the "Use default baud rate" option is ticked. It should not be used for conventional applications, and the tick box should be left clear.

8. INSTALLING THE ACKSYS V2.1 DRIVER FOR LINUX KERNEL 2.2

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/V2.1 on the media supplied.

Installing the module

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/

- 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]

- To install the module, just enter **insmod srllinux**
- To stop the module, enter **rmmod srllinux**
- To check that the module has been correctly installed, use the **lsmod** command.

NOTE:

The next time Linux is started up, the module 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
```

NOTE

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

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

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

```
insmod srllinux input_clock=16000000
```

9. INSTALLING THE ACKSYS V3.4 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 on the media supplied.

9.1 Installing the module

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).

9.2 Starting the driver

To start the driver, type the following command :

```
insmod srlxrspci.o
```

This will create the tty 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
```

note :

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

9.3 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
```

note :

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

9.4 stopping the driver

- To stop the module just enter **`rmmod srlxrspci`**
- To check that the module has been correctly installed, use command **`lsmod`**.

9.5 Note on this 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.

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

```
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 module:

```
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
```

9.6 speed_custom parameter usage

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.

9.7 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

10. USING RS485 MODE UNDER LINUX

In RS485 mode, automatic control of the line direction is managed by the following iocontrol commands :

- ACKSYS_ENABLE_485_MODE: Validate the function of automatic line control.
- ACKSYS_DISABLE_485_MODE: Function disabling the automatic line control.

When enabled, automatic line control is handled directly by the UART. No parameter is required for these iocontrols. They are sent to the driver using linux API function ioctl(...). The communication channel must be opened before calling this function. If you work in RS485 mode, you must not send iocontrol ACKSYS_DISABLE_485_MODE, under risk to disturb the operation of the bus. These constants are defined in the file iocontrol.h.

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

APPENDIX A – THE 62 PIN CONNECTOR (4 AND 8 PORT CARDS)

10.1 RS232-version cards

Pin	RS232
1	DCD1
2	TXD1
3	DSR1
4	DCD2
5	DTR2
6	RTS2
7	RXD3
8	DSR3
9	CTS3
10	TXD4
11	DSR4
12	DCD5
13	DTR5
14	RTS5
15	RXD6
16	DSR6
17	CTS6
18	TXD7
19	RTS7
20	RXD8
21	DSR8

Pin	RS232
22	RXD1
23	DTR1
24	RTS1
25	RXD2
26	DSR2
27	CTS2
28	TXD3
29	RTS3
30	DCD4
31	DTR4
32	RTS4
33	RXD5
34	DSR5
35	CTS5
36	TXD6
37	RTS6
38	DCD7
39	DTR7
40	CTS7
41	TXD8
42	RTS8

Pin	RS232
43	GND
44	CTS1
45	TXD2
46	GND
47	DCD3
48	DTR3
49	GND
50	RXD4
51	GND
52	CTS4
53	TXD5
54	GND
55	DCD6
56	DTR6
57	GND
58	RXD7
59	DSR7
60	DCD8
61	DTR8
62	CTS8

Table 6.1. Allocation of signals on the 62-pin connector in RS232 mode

10.2 RS422-version cards

Pin	RS422
1	-pol1
2	A1
3	n.c.
4	-pol2
5	+pol2
6	B2
7	A'3
8	n.c.
9	B'3
10	A4
11	n.c.
12	-pol5
13	+pol5
14	B5
15	A'6
16	n.c.
17	B'6
18	A7
19	B7
20	A'8
21	n.c.

Pin	RS422
22	A'1
23	+pol1
24	B1
25	A'2
26	n.c.
27	B'2
28	A3
29	B3
30	-pol4
31	+pol4
32	B4
33	A'5
34	n.c.
35	B'5
36	A6
37	B6
38	-pol7
39	+pol7
40	B'7
41	A8
42	B8

Pin	RS422
43	GND
44	B'1
45	A2
46	GND
47	-pol3
48	+pol3
49	GND
50	A'4
51	GND
52	B'4
53	A5
54	GND
55	-pol6
56	+pol6
57	GND
58	A'7
59	n.c.
60	-pol8
61	+pol8
62	B'8

Table 6.2. Allocation of signals on the 62-pin connector in RS422 mode

Polarisation and termination resistances:

All channels have termination and polarization resistances on the reception pairs (A'B'). These resistances can be removed by removing the RS16 (for channels 1 to 4) and RS40 (for channels 5 to 8) relays. In this case, it is still possible to polarize the channels individually by using the polarization outputs available on the 62-point connector (J5): link **+pol$_n$** to **B'$_n$** and **-pol$_n$** to **A'$_n$**.

For example, to polarize channel **3**, link **+pol$_3$ (48)** to **B'$_3$ (9)** and **–pol$_3$ (47)** to **A'$_3$ (7)** .

If the termination resistances are mounted individually, they must be cabled externally.

10.3 Cartes en version RS485

Pin	RS485
1	-pol1
2	reserved
3	n.c.
4	-pol2
5	+pol2
6	reserved
7	AA'3
8	n.c.
9	BB'3
10	reserved
11	n.c.
12	-pol5
13	+pol5
14	reserved
15	AA'6
16	n.c.
17	BB'6
18	reserved
19	reserved
20	AA'8
21	n.c.

Pin	RS485
22	AA'1
23	+pol1
24	reserved
25	AA'2
26	n.c.
27	BB'2
28	reserved
29	reserved
30	-pol4
31	+pol4
32	reserved
33	AA'5
34	n.c.
35	BB'5
36	reserved
37	reserved
38	-pol7
39	+pol7
40	BB'7
41	reserved
42	reserved

Pin	RS485
43	GND
44	BB'1
45	reserved
46	GND
47	-pol3
48	+pol3
49	GND
50	AA'4
51	GND
52	BB'4
53	reserved
54	GND
55	-pol6
56	+pol6
57	GND
58	AA'7
59	n.c.
60	-pol8
61	+pol8
62	BB'8

Table 2. Allocation of signals on the 62-pin connector in RS485 mode

Polarization and termination resistances:

All channels have termination and polarization resistances. These resistances can be removed by removing the RS16 (for channels 1 to 4) and RS40 (for channels 5 to 8) relays. In this case, it is still possible to polarize the channels individually by using the polarization outputs available on the 62-point connector (J5): link **+pol$_n$** to **BB'$_n$** and **-pol$_n$** to **AA'$_n$**.

For example, to polarize channel **3**, link **+pol$_3$ (48)** to **BB'$_3$(9)** and **–pol$_3$(47)** to **AA'$_3$(7)**

If the termination resistances are mounted individually, they must be cabled externally.

11. APPENDIX B – TROUBLESHOOTING

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

Check the first problem.
In the System properties window, check that the XRSPCI 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 XRSPCI card and your equipment do not work

Check the connection between your equipment and the XRSPCI 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. Another method consists of identifying the A, B, A', B' or AA' or BB' signals on your equipment and to make up the cable as described in paragraphs 4.2.1 and 4.2.2. Note that signal A must have a lower voltage than signal B in the MARK state (idle state).

If you think that either the card or the software has a problem, contact ACKSYS

- Via e-mail at support@acksys.fr or by fax on +33 1 39 11 47 96, describing your problem on the "Fault report form" given at the end of the manual.

- By telephone on +33 1 39 11 62 81

12. XRSPCI FAULT REPORT FORM

Name	
Company	
Telephone	
Fax	
E-mail	
XRSPCI serial number	
Operating system	
ACKSYS driver version	
Type of computer	

Description of the problem

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....