



ACCES I/O PRODUCTS INC  
10623 Roselle Street, San Diego, CA 92121  
TEL (858)550-9559 FAX (858)550-7322

---

# **MODEL PCI-COM422/4**

## **USER MANUAL**



## **Notice**

The information in this document is provided for reference only. ACCES does not assume any liability arising out of the application or use of the information or products described herein. This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of ACCES, nor the rights of others.

IBM PC, PC/XT, and PC/AT are registered trademarks of the International Business Machines Corporation.

Printed in USA. Copyright 1995 by ACCES I/O Products Inc, 10623 Roselle Street, San Diego, CA 92121. All rights reserved.

## **Warranty**

Prior to shipment, ACCES equipment is thoroughly inspected and tested to applicable specifications. However, should equipment failure occur, ACCES assures its customers that prompt service and support will be available. All equipment originally manufactured by ACCES which is found to be defective will be repaired or replaced subject to the following considerations.

## **Terms and Conditions**

If a unit is suspected of failure, contact ACCES' Customer Service department. Be prepared to give the unit model number, serial number, and a description of the failure symptom(s). We may suggest some simple tests to confirm the failure. We will assign a Return Material Authorization (RMA) number which must appear on the outer label of the return package. All units/components should be properly packed for handling and returned with freight prepaid to the ACCES designated Service Center, and will be returned to the customer's/user's site freight prepaid and invoiced.

## **Coverage**

First Three Years: Returned unit/part will be repaired and/or replaced at ACCES option with no charge for labor or parts not excluded by warranty. Warranty commences with equipment shipment.

Following Years: Throughout your equipment's lifetime, ACCES stands ready to provide on-site or in-plant service at reasonable rates similar to those of other manufacturers in the industry.

## **Equipment Not Manufactured by ACCES**

Equipment provided but not manufactured by ACCES is warranted and will be repaired according to the terms and conditions of the respective equipment manufacturer's warranty.

## **General**

Under this Warranty, liability of ACCES is limited to replacing, repairing or issuing credit (at ACCES discretion) for any products which are proved to be defective during the warranty period. In no case is ACCES liable for consequential or special damage arriving from use or misuse of our product. The customer is responsible for all charges caused by modifications or additions to ACCES equipment not approved in writing by ACCES or, if in ACCES opinion the equipment has been subjected to abnormal use. "Abnormal use" for purposes of this warranty is defined as any use to which the equipment is exposed other than that use specified or intended as evidenced by purchase or sales representation. Other than the above, no other warranty, expressed or implied, shall apply to any and all such equipment furnished or sold by ACCES.

## Table of Contents

<b>Chapter 1: Introduction</b> .....	<b><a href="#">1-1</a></b>
RS422 Balanced Mode Operation .....	<a href="#">1-1</a>
COM Port Compatibility .....	<a href="#">1-1</a>
Communication Mode .....	<a href="#">1-1</a>
Baud Rate Ranges .....	<a href="#">1-2</a>
Specifications .....	<a href="#">1-2</a>
<b>Chapter 2: Installation</b> .....	<b><a href="#">2-1</a></b>
CD Installation .....	<a href="#">2-1</a>
3.5-Inch Diskette Installation .....	<a href="#">2-1</a>
Directories Created on the Hard Disk .....	<a href="#">2-2</a>
Installing the Card .....	<a href="#">2-4</a>
Input/Output Connections .....	<a href="#">2-5</a>
<b>Chapter 3: Option Selection</b> .....	<b><a href="#">3-1</a></b>
<b>Chapter 4: Address Selection</b> .....	<b><a href="#">4-1</a></b>
<b>Chapter 5: Programming</b> .....	<b><a href="#">5-1</a></b>
Sample Programs .....	<a href="#">5-1</a>
Windows Programming .....	<a href="#">5-1</a>
Initialization .....	<a href="#">5-1</a>
Reception .....	<a href="#">5-3</a>
Transmission .....	<a href="#">5-4</a>
<b>Chapter 6: Connector Pin Assignments</b> .....	<b><a href="#">6-1</a></b>

## List of Figures

Figure 1-1: PCI-COM422/4 Block Diagram .....	<a href="#">Page 1-3</a>
Figure 3-1: Simplified Termination Schematic .....	<a href="#">Page 3-1</a>
Figure 3-2: PCI-COM422/4 Option Selection Map .....	<a href="#">Page 3-3</a>

## List of Tables

Table 5-1: Baud Rate Divisor Values .....	<a href="#">Page 5-2</a>
Table 6-1: Connector Pin Assignments .....	<a href="#">Page 6-1</a>

# Chapter 1: Introduction

The PCI-COM422/4 Serial Interface Card was designed for effective transmission in RS422 (EIA422) protocol. The card is 6.1 inches long and may be installed in PCI-bus slots of IBM PC or compatible computers. The card features four independent, asynchronous RS422 serial ports, type 16550 buffered UARTs, and, for Windows compatibility, automatic control to transparently enable/disable the transmission drivers.

There are two I/O connector options. The standard model PCI-COM422/4 includes a single DB25 male connector on the mounting bracket plus a breakout cable that terminates in four 9-pin female connectors. Optional model PCI-COM422/4S1 has two 9-pin male connectors on the card mounting bracket plus a second mounting bracket with two more 9-pin male connectors with ribbon cables to connect them to headers on the card.

## RS422 Balanced Mode Operation

---

The PCI-COM422/4 supports RS422 communications and uses differential balanced drivers for long range and noise immunity. PCI-COM422/4 also has the capability to add load resistors to terminate the communications lines. RS422 communications requires that a transmitter supply a bias voltage to ensure a known "zero" state. Also, receiver inputs at each end of the network should be terminated to eliminate "ringing". The PCI-COM422/4 supports biasing by default and supports termination by jumpers on the card. If your application requires the transmitter to be un-biased, please contact the factory.

## COM Port Compatibility

---

Type 16550 UARTs are used as the Asynchronous Communication Element (ACE). These include a 16-byte transmit/receive buffer to protect against lost data in multitasking operating systems, while maintaining 100 percent compatibility with the original IBM serial port. The system assigns the address(es).

A crystal oscillator is located on the card. This oscillator ensures precise selection of baud rates up to 115,200 or, by changing a jumper, up to 460,800 with the standard crystal oscillator.

The driver/receiver used, the SN75176B, is capable of driving extremely long communication lines at high baud rates. It can drive up to  $\pm 60$  mA on balanced lines and receive inputs as low as 200 mV differential signal superimposed on common mode noise of +12 V or -7 V. In case of communication conflict, the driver/receivers feature thermal shutdown.

## Communication Mode

---

PCI-COM422/4 supports Full-Duplex and Half-Duplex communications with a 4-wire cable connection. Half-Duplex allows traffic to travel in both directions, but only one way at a time.

## Baud Rate Ranges

---

The card has capability for two baud rate ranges and you can select which you wish to use on a port-by-port basis. One range is for up to 115,200 baud applications and the other is up to 460,800 baud applications.

## Specifications

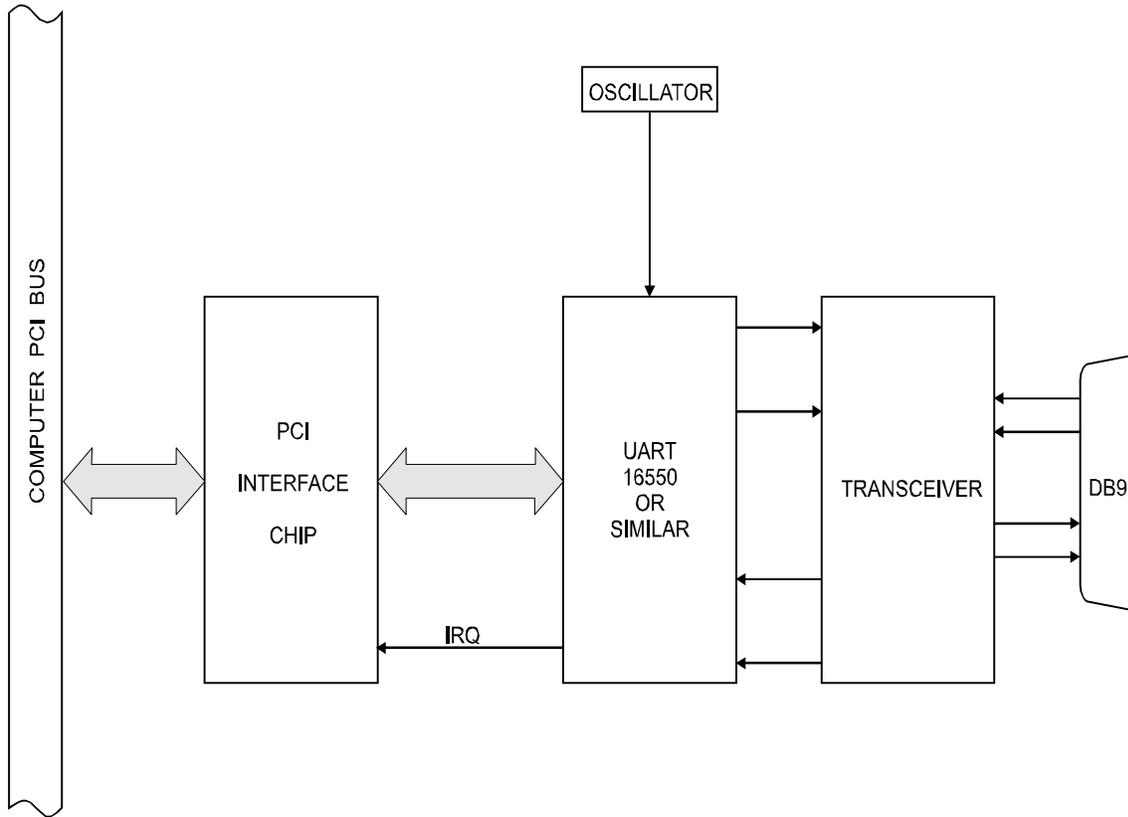
---

### Communications Interface

- I/O Connection: 9-pin D-sub connectors
- Serial Ports: Four shielded male D-sub 9-pin IBM AT style connectors compatible with RS422 specifications. (Note: On model an external "spider" or breakout cable terminates with four female D-sub 9-pin connectors.)
- Character length: 5, 6, 7, or 8 bits.
- Parity: Even, odd or none.
- Stop Interval: 1, 1.5, or 2 bits.
- Serial Data Rates: Up to 115,200 baud, Asynchronous, A faster range of rates up to 460,800, is achieved by jumper selection on the card. Type 16550 buffered UART.
- Address: Continuously mappable within 0000 to FFFF (hex) range of PCI bus addresses (Conforms to PCI Specification 2.1).
- Receiver Input Sensitivity:  $\pm 200$  mV, differential input.
- Common Mode Rejection: +12V to -7V
- Transmitter Output Drive Capability: 60 mA, with thermal shutdown.

### Environmental

- Operating Temperature Range: 0 °C. to +60 °C.
- Storage temperature Range: -50 °C. to +120 °C.
- Humidity: 5% to 95%, non-condensing.
- Size: 6.1" long (155 mm) by 3.9" (99 mm).
- Power Required: +5VDC at 125 mA typical, -12VDC at 5 mA typical, +12VDC at 5 mA typical, 750 mW total power consumption.



**Figure 1-1: PCI-COM422/4 Block Diagram**  
 (Only One Serial Channel Shown)



# Chapter 2: Installation

The software provided with this card is contained on either one CD or multiple diskettes and must be installed onto your hard disk prior to use. To do this, perform the following steps as appropriate for your software format and operating system. Substitute the appropriate drive letter for your CD-ROM or disk drive where you see d: or a: respectively in the examples below.

## CD Installation

---

### DOS/WIN3.x

1. Place the CD into your CD-ROM drive.
2. Type `[?][?][?]` to change the active drive to the CD-ROM drive.
3. Type `[?][?][?][?][?][?][?]` to run the install program.
4. Follow the on-screen prompts to install the software for this card.

### WIN95/98/NT

1. Place the CD into your CD-ROM drive.
2. The CD should automatically run the install program after 30 seconds. If the install program does not run, click START | RUN and type d:install, click OK or press `[?]`.
3. Follow the on-screen prompts to install the software for this card.

## 3.5-Inch Diskette Installation

---

As with any software package, you should make backup copies for everyday use and store your original master diskettes in a safe location. The easiest way to make a backup copy is to use the DOS DISKCOPY utility.

In a single-drive system, the command is:

`[?][?][?][?][?][?][?][?][?][?][?][?][?][?]`

You will need to swap disks as requested by the system.

In a two-disk system, the command is:

`[?][?][?][?][?][?][?][?][?][?][?][?][?][?]`

This will copy the contents of the master disk in drive A to the backup disk in drive B.

To copy the files on the master diskette to your hard disk, perform the following steps.

1. Place the master diskette into a floppy drive.
2. Change the active drive to the drive that has the diskette installed. For example, if the diskette is in drive A, type `A:` .
3. Type `XCOPY *.* /S` and follow the on-screen prompts.

## Directories Created on the Hard Disk

---

The installation process will create several directories on your hard disk. If you accept the installation defaults, the following structure will exist.

### **[CARDNAME]**

Root or base directory containing the SETUP.EXE setup program used to help you configure jumpers and calibrate the card.

**DOS\PSAMPLES:** A subdirectory of [CARDNAME] that contains Pascal samples.

**DOS\CSAMPLES:** A subdirectory of [CARDNAME] that contains "C" samples.

**Win32\language:** Subdirectories containing samples for Win95/98 and NT.

### **WinRisc.exe**

A Windows dumb-terminal type communication program designed for RS422/485 operation. Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line. Can be used to say hello to an installed modem.

### **ACCES32**

This directory contains the Windows 95/98/NT driver used to provide access to the hardware registers when writing 32-bit Windows software. Several samples are provided in a variety of languages to demonstrate how to use this driver. The DLL provides four functions (InPortB, OutPortB, InPort, and OutPort) to access the hardware.

This directory also contains the device driver for Windows NT, ACCESNT.SYS. This device driver provides register-level hardware access in Windows NT. Two methods of using the driver are available, through ACCES32.DLL (recommended) and through the DeviceIOControl handles provided by ACCESNT.SYS (slightly faster).

## **SAMPLES**

Samples for using ACCES32.DLL are provided in this directory. Using this DLL not only makes the hardware programming easier (MUCH easier), but also one source file can be used for both Windows 95/98 and WindowsNT. One executable can run under both operating systems and still have full access to the hardware registers. The DLL is used exactly like any other DLL, so it is compatible with any language capable of using 32-bit DLLs. Consult the manuals provided with your language's compiler for information on using DLLs in your specific environment.

## **VBACCES**

This directory contains sixteen-bit DLL drivers for use with VisualBASIC 3.0 and Windows 3.1 only. These drivers provide four functions, similar to the ACCES32.DLL. However, this DLL is only compatible with 16-bit executables. Migration from 16-bit to 32-bit is simplified because of the similarity between VBACCES and ACCES32.

## **PCI**

This directory contains PCI-bus specific programs and information. If you are not using a PCI card, this directory will not be installed.

## **SOURCE**

A utility program is provided with source code you can use to determine allocated resources at run-time from your own programs in DOS.

## **PCIFind.exe**

A utility for DOS and Windows to determine what base addresses and IRQs are allocated to installed PCI cards. This program runs two versions, depending on the operating system. Windows 95/98/NT displays a GUI interface, and modifies the registry. When run from DOS or Windows3.x, a text interface is used. For information about the format of the registry key, consult the card-specific samples provided with the hardware. In Windows NT, NTioPCI.SYS runs each time the computer is booted, thereby refreshing the registry as PCI hardware is added or removed. In Windows 95/98/NT PCIFind.EXE places itself in the boot-sequence of the OS to refresh the registry on each power-up.

This program also provides some COM configuration when used with PCI COM ports. Specifically, it will configure compatible COM cards for IRQ sharing and multiple port issues.

## **WIN32IRQ**

This directory provides a generic interface for IRQ handling in Windows 95/98/NT. Source code is provided for the driver, greatly simplifying the creation of custom drivers for specific needs. Samples are provided to demonstrate the use of the generic driver. Note that the use of IRQs in near-real-time data acquisition programs requires multi-threaded application programming techniques and must be considered an intermediate to advanced programming topic. Delphi, C++ Builder, and Visual C++ samples are provided.

### **Findbase.exe**

DOS utility to determine an available base address for ISA bus , non-Plug-n-Play cards. Run this program once, before the hardware is installed in the computer, to determine an available address to give the card. Once the address has been determined, run the setup program provided with the hardware to see instructions on setting the address switch and various option selections.

### **Poly.exe**

A generic utility to convert a table of data into an nth order polynomial. Useful for calculating linearization polynomial coefficients for thermocouples and other non-linear sensors.

### **Risc.bat**

A batch file demonstrating the command line parameters of RISCTerm.exe.

### **RISCTerm.exe**

A dumb-terminal type communication program designed for RS422/485 operation. Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line. Can be used to say hello to an installed modem. RISCTerm stands for Really Incredibly Simple Communications TERMinal.

## **Installing the Card**

---

The PCI-COM422/4 card can be installed in a five-volt PCI-bus slot. Before installing the card, carefully read Chapter 3, Option Selection and configure the card according to your requirements. Our SETUP.EXE program will lead you through this process. The setup program does not set the options. You set them manually by jumpers on the card.

### **To Install the Card**

1. Turn OFF computer power.
2. Remove the computer cover.
3. Install jumpers from either Chapter 3, Option Selection of this manual or the suggestions of our SETUP.EXE software program.
4. Install the card in an available PCI-bus slot. You may need to remove a backplate first. Note: The standard model PCI-COM422/4 includes one 25-pin connector on the card mounting bracket plus a breakout cable that terminates in four DB9 female connectors. Thus this unit requires one slot position.
5. Inspect for proper fit of the card and tighten screws. Make sure that the card mounting bracket is properly screwed into place and that there is a positive chassis ground.
6. Replace the computer cover and turn the computer ON.
7. Enter the CMOS setup program of your system and verify that the PCI Plug-and-Play option is set appropriately for your system. Systems running Windows95 (or any other PNP-compliant Operating System) should set the CMOS option to OS. Systems running under DOS, WindowsNT 3.51, Windows 3.1, or any other non-PNP-compliant Operating System should set the PNP CMOS option to BIOS or Motherboard. Save the option and continue booting the system.

Optional model PCI-COM422/4S1 has connectors for two of the four ports and includes a second mounting bracket which provides connections for two more ports. That companion mounting bracket must be mounted in an adjacent position because ribbon cables are used to mate those two connectors via headers on the card.

If you are using Windows95/98, your operating system should detect the new hardware and prompt you for the installation CD. The simplest option is to insert the CD provided into the CD ROM and allow the operating system to look for the files that it wants.

Alternatively, if you have already installed the software to your computer, you can specify the installation directory instead of allowing Windows95/98 to look around. Either option will work acceptably. Please note that even though Windows95/98 looks for and finds software to install, no driver is actually installed, used, or needed. This step merely tells Windows95 where to place the card in the registry of devices. You can find the installed card in Device Manager, under the class Data Acquisition. Use this method to determine the card's base address as PCIFind.EXE may or may not work under your specific installation of Windows95.

If you are not using Windows95, or if your installation of Windows95 does not detect the new hardware you installed, run PCIFind.EXE to determine the base address that the computer assigned. PCIFind uses the Vendor ID and Device ID to search for your card, then reads the base address and IRQ assigned. If you want to determine the base address and IRQ assignment yourself, The Vendor ID will be 494F and the Device ID will be 1058.

The base address assigned by BIOS or the operating system can change each time new hardware is installed into or removed from the computer. Please re-check PCIFind or Device Manager if the hardware configuration is changed,

## **Input/Output Connections**

---

There are two I/O connector configurations. Model PCI-COM422/4 has a male DB25 connector on the card mounting bracket and breakout cables that terminate in four female DB9 connectors. Optional model PCI-COM422/4S1 has two DB9 connectors on the card mounting bracket (Ports A and B) and two more DB9 connectors on a separate mounting bracket (for Ports C and D).

To ensure that there is minimum susceptibility to EMI and minimum radiation, it is important that the card mounting bracket be properly screwed into place and that there be a good connection to the chassis. Also, proper EMI cabling techniques (cable connect to chassis at the aperture, shielded twisted-pair wiring, etc) be used for the input/output wiring.



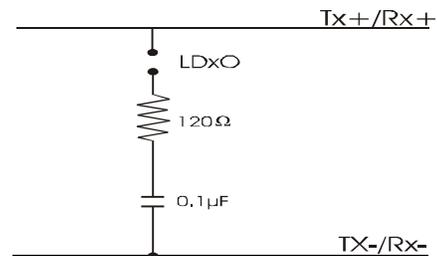
## Chapter 3: Option Selection

To help you locate the jumpers described in this section, refer to Figure 3-2, Option Selection Map at the end of this section. Operation of the serial communications section is determined by jumper installation as described in the following paragraphs.

### Terminations

A transmission line should be terminated at the receiving end in its characteristic impedance. Installing a jumper at the locations labeled LDxO and LDxI apply a 120 $\Omega$  load across the transmit/receive input/output for RS422 operation.

In RS422 operations where there are multiple terminals, only the RS422 ports at each end of the network should have terminating impedance as described above. To so terminate the COM A port, place a jumper at the location labeled LDAO. To terminate the COM B, COM C, and COM D ports, place jumpers at locations labeled LDBO, LDCO, and LDDO respectively. Similarly, inputs are terminated at jumpers labeled LDAI, LDB1, etc.



**Figure 3-1:** Simplified Termination Schematic

Also, for RS422 operation, there must be a bias on the Tx+ and Tx- lines. If the COM422/4 card is not to provide that bias, contact the factory technical support.

### Data Cable Wiring

Signal	Pin Connection
TX+	2
TX-	3
RX+	9
RX-	1
Ground	5

### Baud Rate Ranges

The jumpers labeled CLK X1 and CLK X4 provide means to select baud rates in either of two ranges. When in the "X1" position, the baud rate range is up to 115,200 baud. When in the CLK X4 position, the baud rate range is 200 to 460,800 baud.

## Interrupts

Please note that, in Windows NT, changes must be made to the system registry to support IRQ sharing. The following is excerpted from "Controlling Multiport Serial I/O Cards" provided by Microsoft in the MSDN library, documentid:mk:@ivt:nt40res/D15/S55FC.HTM, also available in the WindowsNT Resource Kit. The text enclosed in brackets ("[" ]") denotes a comment by ACCES.

The Microsoft serial driver can be used to control many dumb multiport serial cards. Dumb indicates that the control includes no on-board processor. Each port of a multiport card has a separate subkey under the CurrentControlSet\Services\Serial subkey in the registry. In each of these subkeys, you must add values for **DosDevices**, **Interrupt**, **InterruptStatus**, **PortAddress**, and **PortIndex** because these are not detected by the Hardware Recognizer. (For descriptions and ranges for these values, see Regentry.hlp, the Registry help file on the WindowsNT Workstation Resource Kit CD.)

For example, if you have a four-port PCI-COM422/4 card configured to use address 0xFFFF0 with an interrupt of 0xFFFF8, the values in the Registry are: [assuming that every port is configured to use the same IRQ and that the addresses are configured to be consecutive and contiguous]

### Serial2 Subkey:

PortAddress = REG\_DWORD 0xFFFF0  
Interrupt = REG\_WORD 5  
DosDevices = REG\_SZ COM3  
InterruptStatus = REG\_DWORD 0xFFFF8  
PortIndex = REG\_DWORD 1

### Serial3 Subkey:

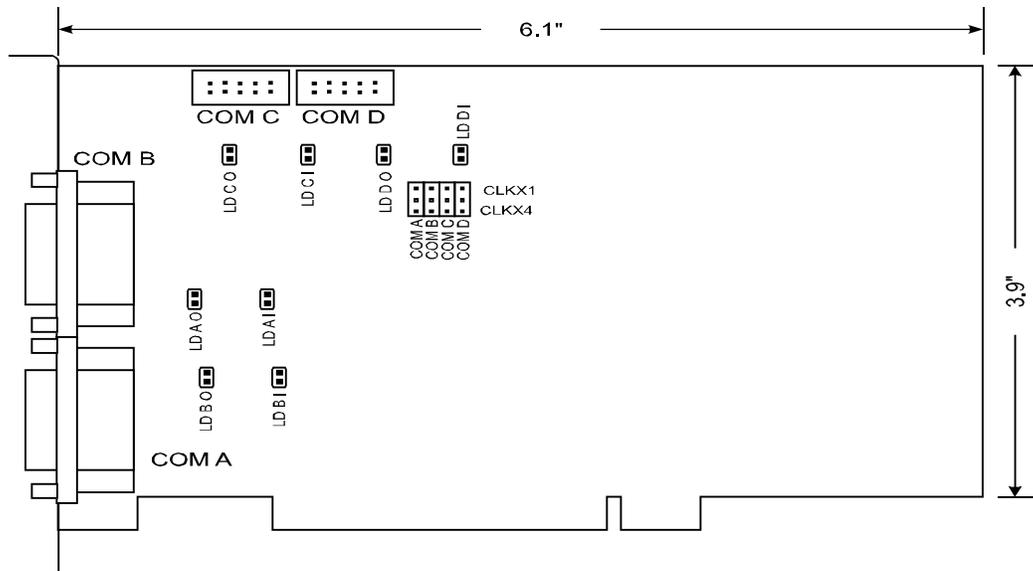
PortAddress = REG\_DWORD 0xFFE0  
Interrupt = REG\_DWORD 5  
DosDevices = REG\_SZ COM4  
InterruptStatus = REG\_DWORD 0xFFFF8  
PortIndex = REG\_DWORD 2

### Serial4 Subkey:

PortAddress = REG\_DWORD 0xFF90  
Interrupt = REG\_DWORD 5  
DosDevices = REG\_SZ COM5  
InterruptStatus = REG\_DWORD 0xFFFF8  
PortIndex = REG\_DWORD 3

### Serial5 Subkey:

PortAddress = REG\_DWORD 0xFF80  
Interrupt = REG\_DWORD 5  
Dos Devices = REG\_SZ COM6  
InterruptStatus = REG\_DWORD 0xFFFF8  
PortIndex = REG\_DWORD 4



**Figure 3-2: PCI-COM422/4 Option Selection Map**

## Chapter 4: Address Selection

The PCI-COM422/4 card uses four separate address spaces. COM A occupies 16 consecutive register locations and COM B, COM C, and COM D each occupy eight consecutive register locations.

PCI architecture is Plug-and-Play. This means that the BIOS or Operating System determines the resources assigned to PCI cards rather than you selecting those resources with switches or jumpers. As a result, you cannot set or change the card's base address. You can only determine what the system has assigned.

To determine the base address that has been assigned, run the PCIFind.EXE utility provided. This utility will display a list of all of the ACCES cards detected on the PCI bus, the addresses assigned to each function on each of the cards, and the respective IRQs (if any) allotted.

Alternatively, some operating systems (Windows95/98/2000) can be queried to determine which resources were assigned. In these operating systems, you can use either PCIFind (Windows95) or PCINT (WindowsNT), or the Device Manager utility from the System Properties Applet of the control panel. The PCI-COM422/4 is installed in the Data Acquisition class of the Device Manager list. Selecting the card, clicking Properties, and then selecting the Resources Tab will display a list of the resources allocated to the card.

PCIFind uses the Vendor ID and Device ID to search for your card, then reads the base address and IRQ. If you want to determine the base address and IRQ yourself, use the following information.

The Vendor ID for this card is 494F. (ASCII for "IO")  
The Device ID for the PCI-COM422/4 is 1058h.

The PCI Bus supports 64K of I/O space, so your card's addresses may be located anywhere in the 0000 to FFFF hex range

# Chapter 5: Programming

## Sample Programs

---

There are sample programs provided with the PCI-COM422/4 card in C, Pascal, QuickBASIC, and several Windows languages. DOS samples are located in the DOS directory and Windows samples are located in the WIN32 directory.

## Windows Programming

---

The PCI-COM422/4 card installs into Windows as COM ports. Thus the Windows standard API functions can be used. In particular:

- ? CreateFile() and CloseHandle() for opening and closing a port.
- ? SetupComm(), SetCommTimeouts(), GetCommState(), and SetCommState() to set and change a port's settings.
- ? ReadFile() and WriteFile() for accessing a port.

See the documentation for your chosen language for details.

Under DOS, the process is very different. The remainder of this chapter describes DOS programming.

## Initialization

---

Initializing the chip requires knowledge of the UART's register set. The first step is to set the baud rate divisor. You do this by first setting the DLAB (Divisor Latch Access Bit) high. This bit is Bit 7 at Base Address +3. In C code, the call would be:

```
outportb(BASEADDR +3,0x80);
```

You then load the divisor into Base Address +0 (low byte) and Base Address +1 (high byte). The following equation defines the relationship between baud rate and divisor:

$$\text{desired baud rate} = (\text{UART clock frequency}) / (32 * \text{divisor})$$

On the PCI-COM422/4 card, the UART clock frequency is 1.8432 MHz. On the next page is a table for the popular divisor frequencies:

Baud Rate	Divisor	Max Diff. Cable Length *
115200	1	2200 ft
57600	2	4000 ft
38400	3	4000 ft
28800	4	4000 ft
19200	6	4000 ft
14400	8	4000 ft
9600	12	4000 ft
4800	24	4000 ft
2400	48	4000 ft
1200	96	4000 ft

**Table 5-1:** Baud Rate Divisor Values

\*These are theoretical maximums based on typical conditions and good quality cables based on the EIA 485 and EIA 422 standard for balanced differential drivers. RS-232 communication lines have a maximum length of 50 feet, regardless of speed.

In C, the code to set the chip to 9600 baud is:

```
outportb(BASEADDR, 0x0C);
outportb(BASEADDR +1,0);
```

The second initializing step is to set the Line Control Register at Base Address +3. This register defines word length, stop bits, parity, and the DLAB.

Bits 0 and 1 control word length and allow word lengths from 5 to 8 bits. Bit settings are extracted by subtracting 5 from the desired word length.

Bit 2 determines the number of stop bits. There can be either one or two stop bits. If Bit 2 is set to 0, there will be one stop bit. If Bit 2 is set to 1, there will be two stop bits.

Bits 3 through 6 control parity and break enable. They are not commonly used for communications and should be set to zeroes.

Bit 7 is the DLAB discussed earlier. It must be set to zero after the divisor is loaded or else there will be no communications.

The C command to set the UART for an 8-bit word, no parity, and one stop bit is:

```
outportb(BASEADDR +3, 0x03)
```

The final initialization step is to flush the receiver buffers. You do this with two reads from the receiver buffer at Base Address +0. When done, the UART is ready to use.

## Reception

---

Reception can be handled in two ways: polling and interrupt-driven. When polling, reception is accomplished by constantly reading the Line Status Register at Base Address +5. Bit 0 of this register is set high whenever data are ready to be read from the chip. A simple polling loop must continuously check this bit and read in data as it becomes available. The following code fragment implements a polling loop and uses a value of 13, (ASCII Carriage Return) as an end-of-transmission marker:

```
do
{
while (!(inportb(BASEADDR +5) & 1)); /*Wait until data ready*/
data[i++] = inportb(BASEADDR);
}
while (data[i] != 13); /*Reads the line until null character rec'd*/
```

Interrupt-driven communications should be used whenever possible and is required for high data rates. Writing an interrupt-driven receiver is not much more complex than writing a polled receiver but care should be taken when installing or removing your interrupt handler to avoid writing the wrong interrupt, disabling the wrong interrupt, or turning interrupts off for too long a period.

The handler would first read the Interrupt Identification Register at Base Address +2. If the interrupt is for Received Data Available, the handler then reads the data. If no interrupt is pending, control exits the routine. A sample handler, written in C, is as follows:

```
readback = inportb(BASEADDR +2);
if (readback & 4) /*Readback will be set to 4 if data are available*/
data[i++] = inportb(BASEADDR);
outportb(0x20, 0x20); /*Write EOI to 8259 Interrupt Controller*/
return;
```

## Transmission

---

RS422 transmission is simple to implement. The AUTO feature of the PCI-COM422/4 card automatically enables the transmitter when data is ready to send so no software enabling is required. The following software example is for non-AUTO operation.

To transmit a string of data, the transmitter must first check Bit 5 of the Line Status Register at Base Address +5. That bit is the transmitter-holding-register-empty flag. If it is high, the transmitter has sent the data. The process of checking the bit until it goes high followed by a write is repeated until no data remains.

The following C code fragment demonstrates this process:

```
outportb(BASEADDR +4, inportb(BASEADDR +4)|0x02);
/*Set RTS bit without altering states of other bits*/
while(data[i]); /*While there is data to send*/
{
    while(!(inportb(BASEADDR +5)&0x20)); /*Wait until transmitter is empty*/
    outportb(BASEADDR,data[i]);
    i++;
}
outportb(BASEADDR +4, inportb(BASEADDR +4)&0xFD);
/*Reset RTS bit without altering states of other bits*/
```

## Chapter 6: Connector Pin Assignments

The popular 9-pin D subminiature connector is used for interfacing to communication lines. The connector is equipped with 4-40 threaded standoffs (female screw lock) to provide strain relief.

Pin No.	RS422 Signals
1	Rx-
2	Tx+
3	Tx-
4	
5	GND Ground
6	
7	
8	
9	Rx+

**Table 6-1:** Connector Pin Assignments



## Customer Comments

If you experience any problems with this manual or just want to give us some feedback, please email us at: [manuals@accessioproducts.com](mailto:manuals@accessioproducts.com). Please detail any errors you find and include your mailing address so that we can send you any manual updates.



10623 Roselle Street, San Diego CA 92121  
Tel. (619)550-9559 FAX (619)550-7322  
[www.accessioproducts.com](http://www.accessioproducts.com)

