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

MODEL PCI-422/485-2

USER MANUAL

FILE: MPCI-422-485-2.A1a

Notice

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

List of Figures

List of Tables

Chapter 1: Introduction

The PCI-422/485-2 Serial Communications Card was designed for use in PCI-Bus computers and provides two serial ports for effective communication in the most common protocols. The card is 6.15 inches long (156 mm) and may be installed in any 5-volt PCI slot in IBM or compatible computers.

Multi-Protocol Operation

These are dual-protocol serial cards that support RS422 and RS485 communications. RS422 uses differential (or balanced) line drivers to improve noise immunity and increase the maximum distance to 4000 feet. RS485 improves on RS422 with switchable transceivers and the ability to support multiple devices on a single "party line". The RS485 specification defines a maximum of 32 devices on a single line. The number of devices served on a single line can be expanded by use of "repeaters".

RS485 and RS422 Balanced Mode Operation

The cards support RS422 and RS485 modes which use differential balanced drivers for increased range and noise immunity. The card also has capability to add bias voltages and to add load resistors to terminate the communications lines. RS485 communications requires that one transmitter supply a bias voltage to ensure a known "zero" state when all transmitters are off. Also, receiver inputs at each end of the network should be terminated to eliminate "ringing". These cards support these options by means of jumpers on the card. See the "Option Selection" section of this manual for more details.

COM Port Compatibility

Type 16550 UARTs are used as the Asynchronous Communication Element (ACE). These include a 16-byte transmit/receive FIFO buffer to protect against lost data in multitasking operating systems, while maintaining 100% compatibility with the original IBM serial port. PCI bus architecture allows addresses between 0000 and FFF8 hex to be assigned to the cards. PCIFind.EXE is a utility provided to determine the base addresses assigned by the system.

A crystal oscillator is located on the card. This oscillator permits selection of baud rates up to 115,200. Rates up to 460,800 baud (4X UART clock) are available by jumper selection.

The driver/receiver used, the 75176, is capable of driving extremely long communication lines at high baud rates. It can drive up to ±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 Modes

The cards support Half-Duplex, and Full-Duplex communications in 2- and 4-wire cable connections. Half-Duplex allows traffic to travel in both directions, but only one way at a time. In Full-Duplex operation, data travels in both directions at the same time. RS485 communications use the half-duplex mode because only a single pair of wires need to be used and installation costs are dramatically reduced.

RTS and Auto Transceiver Control

In RS485 communications the driver must be enabled and disabled as needed, allowing all cards to share a two wire cable. This card provides automatic control, where the driver is enabled when data is ready to be transmitted. The driver remains enabled for the transmission time of one additional character after data transfer is complete and then is disabled. The cards automatically adjust their timing to the baud rate of the data.

The receiver is also normally disabled during RS485 transmissions.

CE Mark

If your card is CE-marked, it meets the requirements of EN50081-1:1992 (Emissions), EN50082-1:1992 (Immunity), and EN60950:1992 (Safety).

Specification

Communications Interface

• Serial Ports:	Shielded male D-sub 9-pin IBM AT style connectors compatible with RS422 and RS485 specifications.
• 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 baud, is achieved by jumper selection on the card. Type 16550 buffered UART.

Caution

The OUT2 bit of the UART must be set low for proper interrupt driven communication. This bit enables or disables the interrupts and the card will not communicate if the bit is pulled high.

Differential Communication Mode

• Multipoint:	Compatible with RS422 and RS485 specifications. Up to 32 drivers and receivers allowed on line. Serial communications ACE used is type 16550. Driver/Receivers used are type 75176.
• Receiver Input Sensitivity:	±200 mV, differential input.
• Common Mode Rejection:	+12V to -7V
• Transmitter Output Drive Capability:	60 mA.

Environmental

• Operating Temperature Range:	0 °C. to +60 °C.
• Storage Temperature Range:	-50 °C. to +120 °C.
• Humidity:	5% to 95%, non-condensing.
• Power Required:	+5VDC at 125 mA typical, +12VDC at 5 mA typical, 685 mW total power consumption.
• Size:	6 1/2" long (165 mm) by 3 7/8" (98 mm).

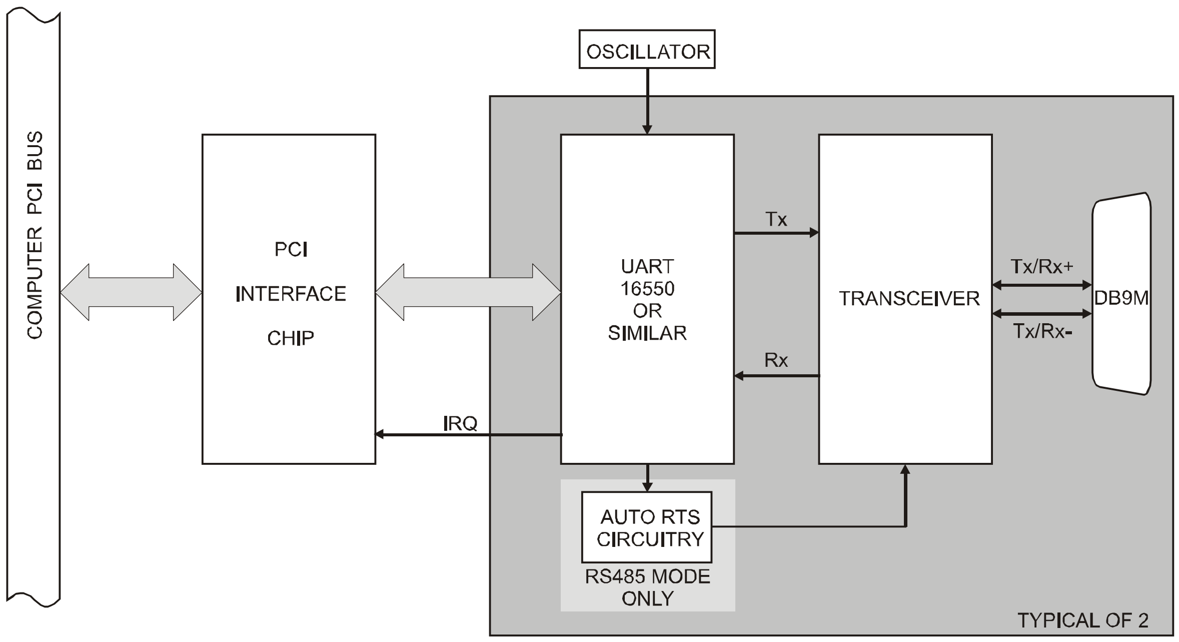

Figure 1-1: PCI-422/485-2 Block Diagram

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/2000

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 ??? .
3. Type ???????? 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-422/485-2 card can be installed in a five-volt PCI slot of an IBM or compatible computer. Before installing the card, carefully read the Option Selection section of this manual and configure the card according to your requirements. Finally, our SETUP.EXE program will lead you through the process of setting the options on the card. The setup program does not set the options. These must be set manually by jumpers on the card.

To Install the Card

1. Install jumpers from either the Option Selection section of this manual or the suggestions of our SETUP.EXE software program.
2. Turn OFF computer power.
3. Remove the computer cover.
4. Install the card in an available PCI-bus slot. You may need to remove a backplate first.
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.
8. Install the card as typical for your OS. A PNP-compliant OS(like Windows NT 2000, Windows NT XP, or any Windows 9x) should automatically detect the card; PNP-compliant Windows OSes use the .INF file on our CD to complete installation. Windows NT4 and lower can use our PCIFind.EXE program to automatically install the card.

Windows COM Numbers

Under Windows 9x, the PCI-422/485-2 will typically install as COM5 and COM6. Using Device Manager, these assignments can be changed to any available COM number greater than COM4; limitations in the Microsoft COM drivers prevent PCI COM ports from appearing as COM1, COM2, COM3, or COM4.

Under Windows NT 2000 and NT XP, the PCI-422/485-2 will typically install as COM3 and COM4. Using Device Manager, these assignments can be changed to any available COM number.

Under Windows NT4 and lower, PCIFind.EXE will typically install the PCI-422/485-2 as COM5 and COM6. Because this is not a PNP-compliant OS, PCIFind.EXE will place itself in the system registry to update the installation should the card move(for example, if another PCI card is installed) or be removed. In order to change the COM number assignment, PCIFind.EXE must be removed from the registry(which will also sacrifice its pseudo-PNP behavior); please contact the factory for further assistance if this is desired.

Input/Output Connections

Two DB9 connectors are provided on the card mounting bracket for input/output connections. 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 positive chassis ground. Also, proper EMI cabling techniques (cable connect to chassis ground at the aperture, shielded twisted-pair wiring, etc) be used for the input/output wiring.

Chapter 3: Option Selection

Operation of the serial communications section is determined by jumper installation as described in the following paragraphs. Locations of the jumpers are shown in Figure 3-1, Option Selection at the end of this section of the manual.

422/485

Jumpers must be installed in these blocks for each COM port. The function of the 422 and 485 jumpers is to configure the port for RS-422 or RS-485 mode communications. Either an RS-422 or RS-485 jumper must be used for each channel.

Terminations and Bias

A transmission line should be terminated at the receiving end in its characteristic impedance. Installing a jumper at the location labeled LDO applies a 120? load across the output for RS-422 mode and across the transmit/receive input/output for RS-485 operation.

Installing a jumper at the location labeled LDI applies a 120? load across the input for RS-422 operation.

The PCI-422/485-2 provides biasing for the RS-485 mode.

CLK X1 and CLK X4

Placement of this jumper controls baud rate. CLK X1 permits baud rates up to 115.2K Baud and CLK X4 permits baud rates up to 460.8K Baud.

Data Cable Wiring

When two numbers are joined together with a "&" the two pins are jumpered together.

MODE		JUMPERS	CABLE CARD1	CABLE CARD2
Simplex	2-wire Receive Only	422	Rx+ 9	2
			Rx- 1	3
Simplex	2-wire Transmit Only	422	Tx+ 2	9
			Tx- 3	1
Half Duplex	2-wire	485	TRx+ 2	2
			TRx- 3	3
Full Duplex	4-wire	422	Tx+ 2	9
			Tx- 3	1
			Rx+ 9	2
			Rx- 1	3

Interrupts

Use PCIFind.EXE to determine the IRQ that was assigned to the PCI-422/485-2 by the BIOS or Operating System. Alternatively, in Windows 9x or NT2000, the Device Manager can be used. The PCI-422/485-2's ports are listed under the Ports class. Selecting the port, clicking Properties, then selecting the Resources tab will show the base addresses and IRQ assigned to the port.

In Windows NT 4 and Lower Without PCIFind.EXE

In Windows NT, changes must be made to the system registry to support IRQ sharing. *PCIFind.EXE normally detects the ACCES COM boards in the system and makes these changes,* but if PCIFind.EXE will not be run or is working improperly for some reason, these changes must be made manually. 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 COM card configured to use address 0xFFF0 with an interrupt of 0xFFF8, 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 0xFFF0
Interrupt = REG_WORD 5
DosDevices = REG_SZ COM3
InterruptStatus = REG_DWORD 0xFFF8
PortIndex = REG_DWORD 1

Serial3 Subkey:
PortAddress = REG_DWORD 0xFFE0
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM4
InterruptStatus = REG_DWORD 0xFFF8
PortIndex = REG_DWORD 2

Serial4 Subkey:
PortAddress = REG_DWORD 0xFF90
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM5
InterruptStatus = REG_DWORD 0xFFF8
PortIndex - REG_DWORD 3

Serial5 Subkey:
PortAddress = REG_DWORD 0xFF80
Interrupt = REG_DWORD 5
Dos Devices = REG_SZ COM6
InterruptStatus = REG_DWORD 0xFFF8
PortIndex = REG_DWORD 4

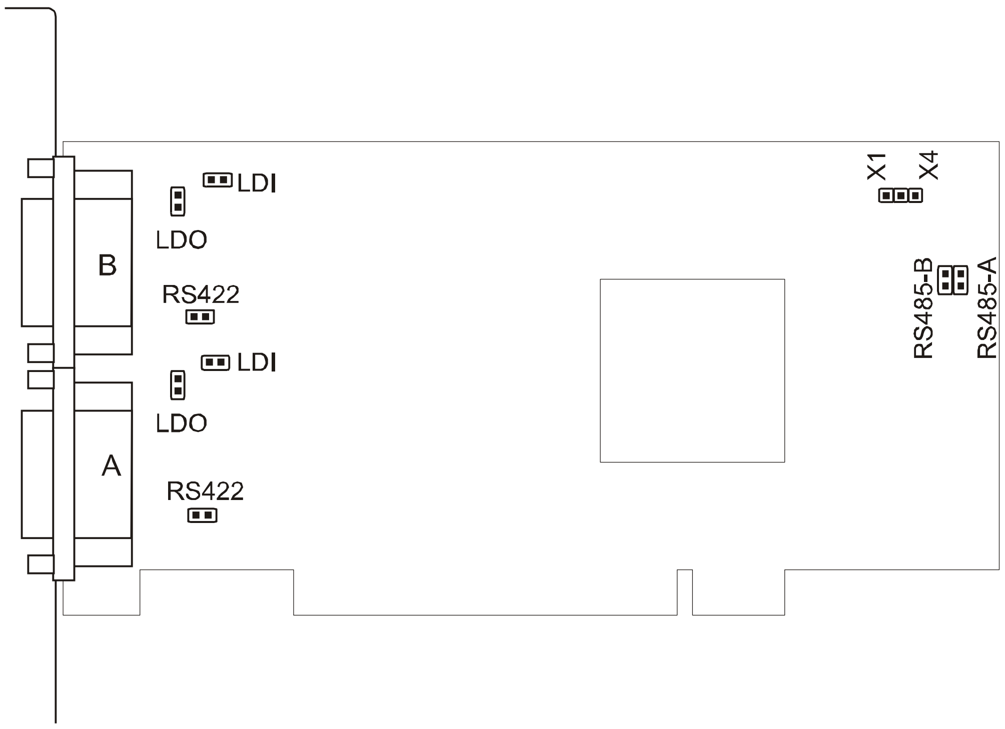

Figure 3-1: PCI-422/485-2 Option Selection Map

Jumpers

RS422:	Select differential(RS-422, four-wire) communications mode
RS485-A, RS485-B:	Select differential(RS-485, two-wire) communications mode
LDO:	Apply load to RS-422 output or RS-485
LDI:	Apply load to RS-422 input
X1/X4:	Select either 115.2 Kbaud or 460.8 Kbaud maximum

Chapter 4: Address Selection

The PCI-422/485-2 uses two separate address spaces, each occupying eight consecutive register locations, for COM A and COM B, respectively. PCI architecture is Plug-and-Play. This means the BIOS or Operating System determines the resources assigned to PCI cards, rather than you selecting these resources with switches or jumpers. As a result, the card's base address cannot be changed, it can only be determined.

To determine the base address that has been assigned to the PCI-422/485-2, run the provided PCIFind.EXE utility program (PCINT.EXE for Windows95 systems.). 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 and DMAs (if any) allotted.

Alternatively, some operating systems (Windows 95/98/2000) can be queried to determine which resources were assigned. In these operating systems, you can use either PCIFind or the Device Manager utility from the System Properties Applet of the control panel. These cards are installed in the Data Acquisition class of the Device Manager list. Selecting the card then clicking Properties, then selecting the Resources Tab will display a list of the resources allocated to the card.

PCIFind uses the Vendor ID and the 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 assigned, use the following information:

The Vendor ID code for this card will be 494F (ASCII for "I/O").
The Device ID code for PCI-422/485-2 will be 1150

PCI supports 64 K of address space, so your card's addresses may be located anywhere in the 0000 to FFF8 hex range.

Chapter 5: Programming

Sample Programs

There are sample programs provided with the PCI-422/485-2 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-422/485-2 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 relaionship between baud rate and divisor:

Desired Baud Rate = (UART clock frequency) ÷ (32   divisor)

When the BAUD jumper is in the X1 position, the UART clock frequency is 1.8432MHz. When the jumper is in the X4 position, the clock frequency is 7.3728 MHz. The following table lists popular divisor frequencies. Note that there are two columns to consider depending on the position of the BAUD jumper.

Baud Rate	**Divisor x1**	**Divisor x4**	**Max. Diff'l. Cable Length***
460800	N/A	1	500 ft
230400	N/A	2	800 ft
153600	N/A	3	1300 ft
115200	1	4	2200 ft.
57600	2	8	4000 ft.
38400	3	12	4000 ft.
28800	4	16	4000 ft.
19200	6	24	4000 ft.
14400	8	32	4000 ft.
9600	12	48	4000 ft.
4800	24	96	4000 ft.
2400	48	192	4000 ft.
1200	96	384	4000 ft.

*These are theoretical maximums based on typical conditions and good quality cables.

Table 5-1: Baud Rate Divisors

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 third step of the initialization sequence is to set the Modem Control Register at Base Address +4. Bit 1 is the Request to Send (RTS) control bit. This bit should be left low until transmission time. (Note: When operating in the automatic RS485 mode, the state of this bit is not significant.) Bits 2 and 3 are user-designated outputs. Bit 2 may be ignored on this card. Bit 3 is used to enable interrupts and should be set high if an interrupt-driven receiver is to be used.

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

The Auto feature of the PCI-422/485-2 card automatically enables the transmitter when data are ready to send so no software enabling is required. The following software example is for non-Auto operation.
First the RTS line should be set high by writing a 1 to Bit 1 of the Modem Control Register at Base Address +4. The RTS line is used to toggle the transceiver from receive mode to transmit mode and vice versa. It is not carried out on the line in RS485 and is not used for handshaking. Similarly, the CTS line is not used in RS485 and should always be enabled by installing a jumper as described earlier.

After the above is done, the card is ready to send data. 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. After all data has been transmitted, the RTS bit should be reset by writing a 0 to Bit 1 of the Modem Control Register.

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*/
```

Caution

The OUT2 bit of the UART must be set low for proper interrupt driven communication. This bit enables or disables the interrupts and the card will not communicate if the bit is pulled high.

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 Functions	RS485 Functions
1	Rx- Receive Data	
2	Tx+ Transmit Data	TRx+ Transmit/Receive Data
3	Tx- Transmit Data	TRx- Transmit/Receive Data
4		
5	GND Ground	GND Ground
6		
7		
8		
9	Rx+ Receive Data	

Table 6-1: Connector Pin Assignments

Note

If connections are to be made to a CE-marked version, then CE-certifiable cabling and breakout methodology (Cable shields connected to ground at the mounting bracket shielded twisted-pair wiring, etc) must be used.

Appendix A: Application Considerations

Balanced Differential Signals

The reason that RS422 and RS485 devices can drive long lines with more noise immunity is that a balanced differential drive method is used. In a balanced differential system, the voltage produced by the driver appears across a pair of wires. A balanced line driver will produce a differential voltage from ±2 to ±6 volts across its output terminals. A balanced line driver can also have an input "enable" signal that connects the driver to its output terminals. If the "enable" signal is OFF, the driver is disconnected from the transmission line. This disconnected or disabled condition is usually referred to as the "tristate" condition and represents a high impedance. RS485 drivers must have this control capability. RS422 drivers may have this control but it is not always required.

A balanced differential line receiver senses the voltage state of the transmission line across the two signal input lines. If the differential input voltage is greater than +200 mV, the receiver will provide a specific logic state on its output. If the differential voltage input is less than -200 mV, the receiver will provide the opposite logic state on its output. The maximum operating voltage range is from +6V to -6V allowing for voltage attenuation that can occur on long transmission cables.

A maximum common mode voltage rating of ±7V provides good noise immunity from voltages induced on the twisted pair lines. The signal ground line connection is necessary in order to keep the common mode voltage within that range. The circuit may operate without the ground connection but may not be reliable.

Parameter	Conditions	Min.	Max.
Driver Output Voltage (unloaded)		4V	6V
		-4V	-6V
Driver Output Voltage (loaded)	LDO	2V	
	jumpers in	-2V	
Driver Output Resistance			50?
Driver Output Short-Circuit Current			±150 mA
Driver Output Rise Time			10% unit interval
Receiver Sensitivity			±200 mV
Receiver Common Mode Voltage Range			±7V
Receiver Input Resistance			4K?

Table A-1: RS422 Specification Summary

To prevent signal reflections in the cable and to improve noise rejection in both the RS422 and RS485 mode, the receiver end of the cable should be terminated with a resistance equal to the characteristic impedance of the cable. (The exception is when the line is driven by an RS422 driver that is never "tristated" or disconnected from the line. In this case, the driver provides a low internal impedance that terminates the line at that end.)

Note

You do not have to add a terminator resistor to your cables when you use the PCI-422/485-2 card. Termination resistors for the RX+ and RX- lines are provided on the card and are placed in the circuit when you install the termination jumpers. Moreover, installing the BIAS jumpers properly biases these lines. (See the Option Selection section of this manual.)

RS485 Data Transmission

The RS485 Standard allows a balanced transmission line to be shared in a party-line mode. As many as 32 driver/receiver pairs can share a two-wire party line network. Many characteristics of the drivers and receivers are the same as in the RS422 Standard. One difference is that the common mode voltage limit is extended and is +12V to -7V. Since any driver can be disconnected (or tristated) from the line, it must withstand this common mode voltage range while in the tristate condition.

RS485 Two-Wire Multidrop Network

The following illustration shows a typical multidrop or party line network. Note that the transmission line is terminated on both ends of the line but not at drop points in the middle of the line.

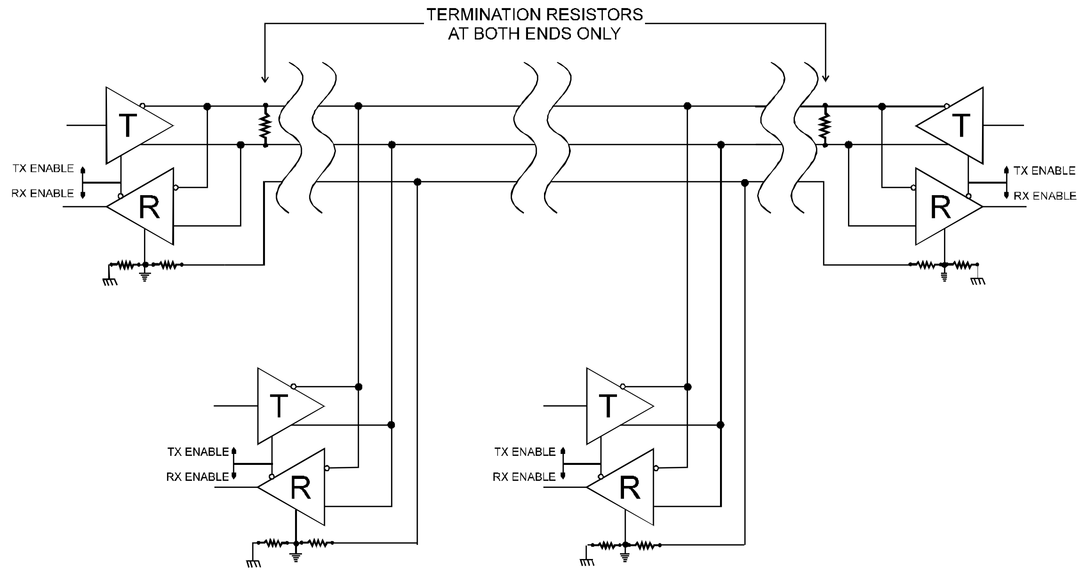

Figure A-1: Typical RS485 Two-Wire Multidrop Network

RS485 Four-Wire Multidrop Network

An RS485 network can also be connected in a four-wire mode. In a four-wire network it's necessary that one node be a master node and all others be slaves. The network is connected so that the master communicates to all slaves and all slaves communicate only with the master. This has advantages in equipment that uses mixed protocol communications. Since the slave nodes never listen to another slave's response to the master, a slave node cannot reply incorrectly.

Customer Comments

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

ACCES I/O PRODUCTS, INC.
10623 Roselle Street, San Diego CA 92121
Tel. (858)550-9559   FAX (858)550-7322
www.accesioproducts.com