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

PCMCIA SERIAL INTERFACE CARD

MODEL PCM485

USER MANUAL

File: MPCM485.E3a

NOTICES

The information in this document is provided for reference only. ACCES I/O Products Inc 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.

U.S. Patent No.s 4,603,320 and 4,972,470.

TABLE OF CONTENTS

INSTALLING THE CARD

The PCM485 card can be installed in any PCMCIA Type II card slot. There are no switches or jumpers to set. Everything on the PCM485 card is programmable including address and interrupt level.

DOS OR WINDOWS 3.X INSTALLATION

You must have installed the Card & Socket Services (CSS) software that was provided with your PCMCIA-capable computer. For complete installation instructions refer to “readme.txt” in your PCM485 software directory.

CAUTION:

The PCM485 must be oriented with the label up, or “side A” in PCMCIA terms. The standard PCMCIA case is “keyed” to prevent improper insertion but is plastic and may be broken off if excessive force is used during installation.

DO NOT FORCE THE PCM485 CARD INTO THE SLOT.

Turn the power off and install the PCM485 card into the PCMCIA slot of your computer. Install the card with the label facing “up” or towards side “A” which is usually towards the keyboard, but you should check with the owners manual. When correctly oriented, the PCM485 card should seat in the slot with little pressure. Force the PCM485 card in upside down and you will probably burn out components as well as destroy the key slot.

Then reboot your computer so the drivers are loaded into memory. You may then run the setup and test software to confirm operation of your PCM485 card.

DOS ‘PLUG N PLAY’ OR SUPERCLIENT INSTALLATION

The SuperClient uses the PCM485 internal CIS (configuration information) to configure the card and prepare system for use. Consult the SuperClient documentation to determine the installation procedure.

WINDOWS 95 INSTALLATION

Windows95 will automatically recognize the installation of the PCMCIA card and provide a list of options to install the card. Select the option “Driver from disk provided by hardware manufacturer” and select the CD provided by ACCES which will install support for the card. Depending on options selected in the Windows95 PCMCIA drivers, you may hear a “rising” two-note tone upon successful installation and a new PCMCIA icon in the taskbar tray.

WINDOWS NT 4.0 INSTALLATION

Windows NT 4.0 is not a Plug-and-Play operating system, so getting the card to work properly requires a few extra steps. After physically installing the PCM485 card into the slot, run the “Setup.exe” program included in the software package. This program will create an entry in the PCMCIA Database, allowing Windows NT 4.0 to recognize the PCM485. After running “Setup.exe,” the system must be rebooted for the changes to take effect. The device should then show up in the “PC Card (PCMCIA)” Control Panel where its resource assignments can be found.

Although manual modification of the Registry should not be necessary, the steps necessary to duplicate the actions performed by “Setup.exe” are documented here for informational purposes. After starting Regedit, navigate to the following key:

```
    HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Pcmcia\
    Database
```

Add a new subkey named “ACCES I/O Products,” then add a new subkey to the “ACCES I/O Products” key you just created named “PCM485.” In the “PCM485" key, add a new string with the name “Driver” and the value “PCM485." This completes the modifications necessary for Windows NT 4.0 to recognize the PCM485 card. The system must then be rebooted for the changes to take effect.

Once the card is inserted, you may then run the SAMPLE1 software to confirm operation of your PCM485 card.

FUNCTIONAL DESCRIPTION

FEATURES

- ! Type II PCMCIA Card
- ! Asynchronous RS-485 or RS-422 Serial Port
- ! Type 16550 16-Byte Buffered UART
- ! Baud Rates to 115Kbaud
- ! Automatic Send-Data Control for Windows Compatibility
- ! Extended AT Interrupts (2-7, 10-12, 14 & 15)
- ! Software setup including FINDBASE and TSR setup program
- ! DB-9 Male Connector via cable adaptors
- ! Cable connector compatible with COM-2S and ICOM-2S ports

MULTIPOINT COMMUNICATIONS

PCMCIA started as a way to add extra memory to laptop and portable computers. As the market grew, the need for a standard expansion slot was recognized in the PCMCIA design and the second release added hardware support for I/O devices.

The PCM485 is a multi-protocol serial card supporting RS422 or RS485 communications in a single 5V PCMCIA Type II card. The PCM485 uses a crystal controlled oscillator which ensure precise selection of baud rates from 50 to 115,200.

The output transceiver used, the MAXIM 485, is capable of driving extremely long communication lines at high baud rates. It can drive up to 60mA on balanced lines with a quiescent current of 300µA. It can receive inputs as low as 200mV differential signal superimposed on common mode noise of maximum 500V. In case of communication conflict, the transceivers feature thermal shutdown.

COM PORT COMPATIBILITY

The COM port hardware used on the original 1981 IBM XT was based on the 8250 UART and was designed to operate with bus speeds of 4.77 MHZ. This was replaced by the 16450 which had better AC characteristics allowing higher speeds and was used in all IBM AT class computers. A few years later, the 16550 was introduced with the addition of a First In-First Out (FIFO) buffer on the transmitter and receiver channels to allow higher data rates and protect against data loss in slower computers and multitasking environments. The PCM485 uses a 16550 UART to provide 100 percent compatibility with DOS COM ports.

COMMUNICATION MODES

Model PCM485 supports Simplex, Half-Duplex, and Full-Duplex communications in a variety of 2- and 4-wire cable connections. Simplex is the simplest form of communications with transmissions occurring only in one direction. Half-Duplex allows traffic to travel in both directions, but only one way at a time. In Full-Duplex mode data travels in both directions at the same time. RS485 communication only supports half-duplex since the same pair of signal lines are used for both transmission and reception.

AUTO AND MANUAL TRANSCEIVER CONTROL

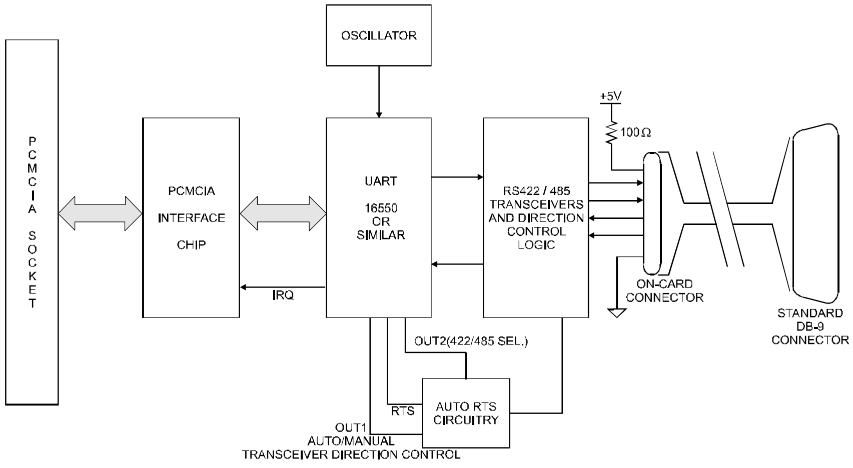
RS485 communications requires the transmitter driver to be enabled and disabled as needed, allowing all cards to share a communications line. The PCM485 card has two methods of controlling the driver; automatic and manual (RTS) control. With automatic control, the driver is enabled when data is ready to be transmitted. The driver remains enabled for the transmission time of one character after data transfer is complete and then is disabled. The PCM485 automatically adjusts it's timing to the baud rate of the data.

CARD AND SOCKET SERVICES

PCMCIA Card and Socket services must be loaded on the host computer system before using the PCM485 card and is typically supplied by the PCMCIA slot provider (the laptop or adaptor manufacturer) to provide the software interface to the card slot drive. In addition, our TSR memory-resident program, described below, is required to configure the card once it is recognized by the card and socket services.

The Card and Socket services handle the PCMCIA slot software interface with your operating system when a new card is inserted (or removed) or when power is applied (or upon shut-down). Once the PCMCIA slot recognizes the PCM485 card it then appears to your application software like a card on the internal ISA bus.

BLOCK DIAGRAM



UTILITY SOFTWARE

We include programs on CD to support the card and to help you develop your application program(s). The card supports I/O bus addresses from 000 to 3F8 (hex) and a program called FINDBASE helps you find an open base address to use with the card.

The TSR setup program has two required command-line parameters (entered in hexadecimal format); the base I/O address and the interrupt request (IRQ) number. Optionally, a “verbose” switch will turn on a detailed report of the communication between the TSR and the PCMCIA card for diagnostic troubleshooting.

Syntax: PCM485 [Base Address in Hex] [IRQ number in Hex] (verbose)

For example: `PCM485 300 0A verbose`

Would configure the PCM485 card at a base address of 300 (hex) and IRQ of 10 (decimal). If no values are used with the TSR program then the following error message is displayed:

```
The command line parameters were incorrect.
syntax: pcm485 <port address> <IRQ>
syntax: pcm485 120 5
syntax: pcm485 300 0A
```

CABLE OPTIONS

There are three cable options available for the PCM485 with connections terminating to a full size female 9-pin D connector. The cable options are:

CAB-PCM-485	for simple connections to RS485 systems.
CAB-PCM-422	for simple connections to RS422 systems.
CAB-PCM-2S	for compatibility with our COM-2S and ICOM-2S cards supporting both RS422 and RS485 communications.

Consult the “CONNECTOR AND CABLES” section for pinouts and wiring information on our cables. See also our appendix on “SERIAL CONNECTIONS” for information on wiring and RS422 and RS485 interconnections.

ADDRESS SELECTION

The PCM485 base address can be selected anywhere within an I/O address range 100-3F8 hex (except 1F0 through 1F8), providing that the addresses do not overlap with other functions. If in doubt, refer to the table below for a list of standard address assignments. (The primary and secondary binary synchronous communication ports are supported by the Operating System.) The base address locator program FINDBASE provided on CD will assist you to select a base address that will avoid this conflict.

STANDARD ADDRESS ASSIGNMENTS FOR 286/386/486 COMPUTERS

Hex Range	Usage
000-01F	DMA Controller 1
020-03F	INT Controller 1, Master
040-05F	Timer
060-06F	8042 (Keyboard)
070-07F	Real Time Clock, NMI Mask
080-09F	DMA Page Register
0A0-0BF	INT Controller 2
0C0-0DF	DMA Controller 2
0F0	Clear Math Coprocessor Busy
0F1	Reset Coprocessor
0F8-0FF	Arithmetic Processor
1F0-1F8	IDE Fixed Disk
200-207	Game I/O
278-27F	Parallel Printer Port 2
2F8-2FF	Asynchronous Comm’n (Secondary)
300-31F	Prototype Card
360-36F	Reserved
378-37F	Parallel Printer Port 1
380-38F	SDLC or Binary Synchronous Comm’n 2
3A0-3AF	Binary Synchronous Comm’n 1
3B0-3BF	Monochrome Display/Printer
3C0-3CE	Local Area Network
3D0-3DF	Color/Graphic Monitor
3F0-3F7	Floppy Diskette Controller
3F8-3FF	Asynchronous Comm’n (Primary)

STANDARD DOS COM PORT ADDRESSES

Serial COM Port	IRQ	Base Hex Address
COM1	IRQ 4	3F8
COM2	IRQ 3	2F8
COM3	IRQ 4	3E8
COM4	IRQ 3	2E8

Notice that only two interrupts are assigned to four serial ports. “Sharing” interrupts is not supported by most computers. Our PCM485 card supports additional addresses and interrupts, limited by your operating system and the PCMCIA port hardware.

PROGRAMMING

SAMPLE PROGRAMS

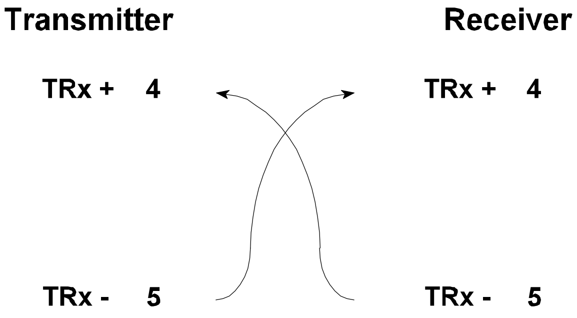
There are two sample programs provided with the PCM485 card. For additional information programming the UART consult any document on the 16550 UART.

Sample 1

This program is provided in C, Pascal, and QuickBASIC. It performs a test of the loopback feature of the UART. It requires no external hardware and no interrupts.

Sample 2

This program is provided in C only and demonstrates interrupt-driven RS485 operation. The program requires multiple cards in multiple computers. To operate this program requires a minimum of two computers or terminals and a two-wire cable interconnecting them. That cable must connect the Tx+ and Tx- pins from terminal 1 to the Rx+ and Rx- pins respectively of terminal 2 and the Tx+ and Tx- pins from terminal 2 to the Rx+ and Rx- pins at terminal 1.



CONTROL REGISTERS

Once the Card & Socket Services (CSS) and Client Drivers are installed, and a base address set using the setup software, the PCM485 card may be controlled by writing to and reading from the control registers. With the AUTO-RTS feature the card is directly compatible with standard DOS and Windows communications routines and drivers for both RS485 and RS422.

Port Addresses & Functions

All I/O accesses may be performed as bytes. Eight-bit or sixteen-bit addressing is controlled by the CSS which is currently set for eight bit addressing. The CSS and Client driver follows the eight-bit addressing used by the 16550 UART device. The registers, in eight-bit format, are presented here:

Address	Write	Read
Base + 0	Transmit Holding	Receive Holding
Base + 1	Interrupt Enable	No Function
Base + 2	FIFO Control	Interrupt Identification Register
Base + 3	Line Control	No Function
Base + 4	Modem Control	No Function
Base + 5	No Function	Line Status
Base + 6	No Function	Modem Status
Base + 7	Transmit/Receive Enable	Signal Confirmation

UART PROGRAMMING

Programming the UART for communication can be divided into three distinct sections: initialization, reception, and transmission. Initialization deals with option setup on the chip including baud rate selection. Reception deals with incoming-character processing which can be done using either polling or interrupts. Transmission deals with the process of sending data out.

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:

desired baud rate = (crystal frequency) / (128 * divisor)

On the PCM485 card, the crystal frequency is 1.8432 MHZ. Below is a table for the popular divisor frequencies:

Baud Rate	Divisor	Notes	Approx Max Cable length[1]
115200	1		375 feet
57600	2		660 ft
38400	3		920 ft
28800	4		1160 ft
19200	6		1620 ft
14400	8		2050 ft
9600	12	Most common industrial speed	4000 ft
4800	24		4000 ft
2400	48		4000 ft
1200	96		4000 ft

[1] These are theoretical maximums based on typical conditions and good quality cables.

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

```
outportb(BASEADDR +0, 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. The bit assignments are:

Bit 0	DTR Control Bit	Leave low, Forces DATA TERMINAL READY line.
Bit 1	RTS Control Bit	This bit should be left low until transmission time.
Bit 2	AUTO-RTS enable	Should not be used with manual control of RTS bit 1.
Bit 3	RS422 enable	This bit selects between RS422 and RS485 modes, setting bit 3 true also sets bit 2 false.
Bit 4	LOOPBACK	Leave low, Used in Loopback tests
Bit 5-7	Unused	Leave low

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 PCM485 card is ready to use.

Reception

Reception can be handled in two ways: polled 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. Polling is not effective at high data rates because the program cannot do anything else when it is polling or data could be missed. 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 sets the received-data-available bit (bit 0), the handler then reads the data. A sample handler, written in C, is as follows:

```
do
{
        readback = inportb(BASEADDR + 2);
        if (readback & 4)       /*Readback bit 2 will be set if data are available*/
        data[i++]=inportb(BASEADDR);
}
 while(readback!=1);
 outportb(0x20,0x20);          /*Write EOI to 8259 Interrupt Controller*/
 return;
```

Transmission

RS485 transmission is simple to implement but requires control of the transmit direction. The PCM485 card supports manual and automatic transmitter control. Once installed, the PCM485 card operates like any other COM port except for the control of the transceiver state which is explained in the following sections:

Manual RTS Control: 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. Note: Bit 2 of modem control register must be low to enable manual RTS mode.

First the RTS line should be set high on Bit 1 of the Modem Control Register at Base Address +4. 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 setting Bit 1 of the Modem Control Register low.

Automatic RTS Control: When the AUTO-RTS bit (bit 2 of the Modem Control Register) is set high then the PCM485 uses the state of the transmit buffer to automatically control the transmit direction.

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*/
```

RS422 vs RS485 Programming

Programming for RS422 operation is a simplified version of RS485 communications, without the overhead of multiple devices on the same line. RS422 also supports multiple devices but only if one port is limited to transmitting and all the other ports are only receivers.

CONNECTOR AND CABLES

The PCM485 requires the appropriate cable to provide the transition between the microminiature 15-pin PCMCIA connector and the 9-pin D subminiature connector. Screw type 4-40 threaded standoffs (female screw lock) are used to provide strain relief.

CAB-PCM-485 CABLE: For simple RS485 only connections:

PCM485 Connector		**Cable**	
Pin	Description	Pin	RS485 Mode
1	Ground	1	Ground
2	Mode		
3	Ground	3	Ground
4	TX+ / RX+	4	TX+/RX+
5	TX- / RX-	5	TX-/RX-
6	+5 VDC (thru 100? limiting resistor)	6	+5 VDC
7	Ground		
8	TX+ / RX+	8	TX+/RX+
9	TX- / RX-	9	TX-/RX-
10	TXDATA+		
11	TXDATA-		
12	CTS+		
13	CTS-		
14	RTS+		
15	RTS-		

CAB-PCM-422 CABLE: For simple, RS422 only connections:

PCM485 Connector		**Cable**	
Pin	Description	Pin	RS422 Mode
1	Ground	1	Ground
2	Mode		
3	Ground		
4	TX+ / RX+		
5	TX- / RX-		
6	+5 VDC (thru 100? limiting resistor)		
7	Ground		
8	TX+ / RX+	8	RXDATA+
9	TX- / RX-	9	RXDATA-
10	TXDATA+	4	TXDATA+
11	TXDATA-	5	TXDATA-
12	CTS+	7	CTS+
13	CTS-	6	CTS-
14	RTS+	2	RTS+
15	RTS-	3	RTS-

CAB-PCM-2S CABLE: For Compatibility with our COM-2S and ICOM-2S cards

PCM485 Connector		**Cable**		
Pin	Description	Pin	RS422 Mode	RS485 Mode
1	Ground			
2	Mode	4	Open	Ground*
3	Ground			
4	TX+ / RX+	9	RXDATA+	TX+/RX+
5	TX- / RX-			
6	+5 VDC (thru 100? limiting resistor)	6	+5VDC	+5VDC
7	Ground	5	Ground	Ground
8	TX+ / RX+			
9	TX- / RX-	1	RXDATA-	TX- / RX-
10	TXDATA+	2	TXDATA+	
11	TXDATA-	3	TXDATA-	
12	CTS+	8	CTS	CTS
13	CTS-			
14	RTS+	7	RTS	RTS
15	RTS-			

** To operate in RS485 mode, provide a ground at Pin 4 from pin 5 of the DB9 connector.*

SPECIFICATIONS

Communications Interface

Multipoint: Compatible with RS422 and RS485 specifications. Serial communications ACE used is a type 16550. Transceivers used are of the type MAXIM485.

Serial Ports: Cable adaptors provide compatibility with COM-2S, ICOM-2S and other popular connection formats.

RS-485 Drive: A total of 32 transmitters and receivers allowed on one line.

Receiver Input Sensitivity: ±200 mV, differential input.

Transmitter Output Drive Capability: 60 mA (100 mA short-circuit current capability).

Serial Data Rates: 50 to 115,200 baud. Asynchronous.

Address: Continuously mappable within 000 to 3FF (hex) range of AT I/O bus addresses.

Interrupt: Selectable for any available IRQ from 2-7, 10-12, 14 & 15 using the software configuration program.

Environmental:

Operating Temperature Range: 0 to +60 °C

Storage Temperature Range: -50 to +120 °C

Humidity: 5% to 95%, non-condensing.

Power Required: 5 VDC, 5 mA (at standby) 105 mA max.

Size: Type II PCMCIA card slot.

Regulatory Compliance:

FCC Part 15, Level A: Designed to meet.

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 conditions.

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 ACCES' designated Service Center. After service, the units/components 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 for parts that are not excluded by Warranty. Warranty period 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 arising from use or misuse of our products. 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 or all such equipment furnished or sold by ACCES.

APPENDIX A
SERIAL CONNECTIONS

INTRODUCTION

Working with RS422 and RS485 devices is not much different from working with standard RS232 serial devices and these two standards overcome deficiencies in the RS232 standard. First, the cable length between two RS232 devices must be short; less than 50 feet at 9600 baud. Second, many RS232 errors are the result of noise induced on the cables. The RS422 standard permits cable lengths up to 5000 feet and, because it operates in the differential mode, it is more immune to induced noise.

Proper selection depends on the wire connections used to connect the two serial ports. Wires labeled Tx are the transmit wires and the receive wires are labeled Rx.

CABLE OPTIONS		MODE	Card A	Card B
Simplex	2-wire Receive Only	422	Rx+ Rx-	Tx+
Simplex	2-wire Transmit Only	422	Tx+ Tx-	Rx+
Half Duplex	2-wire with local echo	422	Rx+,Tx+ Rx-,Tx-	Rx+,Tx+
Half Duplex	2-wire without local echo	422/485	Rx+ Tx+	Tx+ Rx+
Full Duplex	4-wire without local echo	422	Tx+ Rx+ Tx- Rx-	Tx-

A third deficiency of RS232 is that more than two devices cannot share the same cable. This is also true for RS422 *but RS485 offers all the benefits of RS422 plus allows up to 32 devices to share the same twisted pairs*. An exception to the foregoing is that multiple RS422 devices can share a single cable if only one will talk and the others will all receive.

BALANCED DIFFERENTIAL SIGNALS

The reason that RS422 and RS485 devices can drive longer lines with more noise immunity than RS232 devices 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. A maximum operating voltage range is from +6V to -6V allows 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.

RS422 Specification Summary

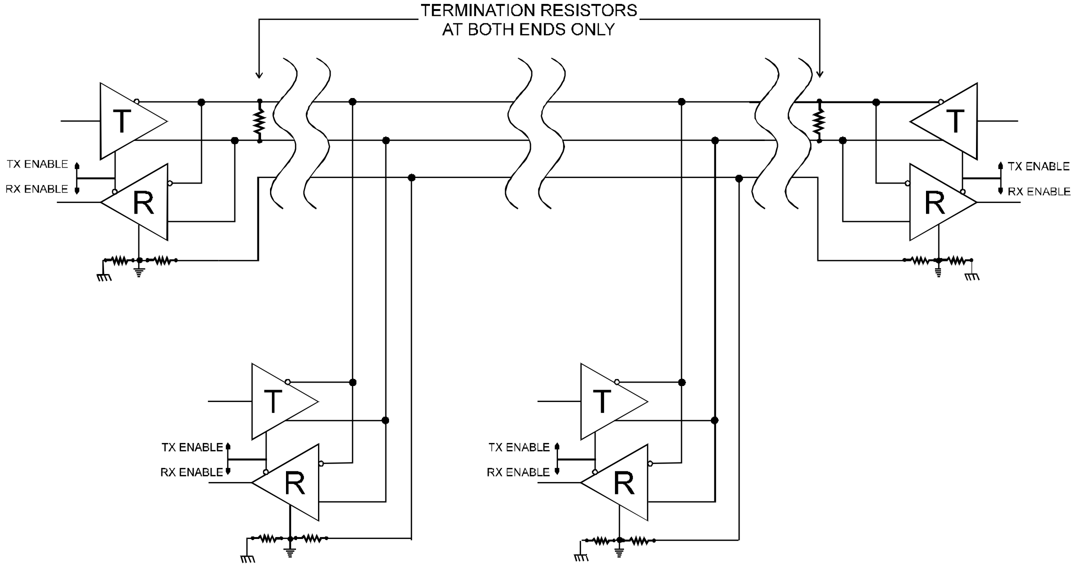
Parameter	Conditions	Min.	Max.
Driver Output Voltage (unloaded)		4V -4V	6V -6V
Driver Output Voltage (loaded)	LD and LDGND jumpers in	2V -2V	
Driver Output Resistance			50?
Driver Output Short-Circuit Current			±150 mA
Driver Output Rise Time			10% unit interval
Receiver Sensitivity			±200 mV
Receiver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

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. (An exception to this is the case where the line is driven by an RS422 driver that is never “tri-stated” or disconnected from the line. In this case, the driver provides a low internal impedance that terminates the line at that end.)

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 tri-stated) from the line, it must withstand this common mode voltage range while in the tristate condition.

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.



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.