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

MODEL COM485/8

USER MANUAL

FILE: MCOM485-8.E1c

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.

WARNING!!

ALWAYS CONNECT AND DISCONNECT YOUR FIELD CABLING WITH THE COMPUTER POWER OFF. ALWAYS TURN COMPUTER POWER OFF BEFORE INSTALLING A CARD. CONNECTING AND DISCONNECTING CABLES, OR INSTALLING CARDS INTO A SYSTEM WITH THE COMPUTER OR FIELD POWER ON MAY CAUSE DAMAGE TO THE I/O CARD AND WILL VOID ALL WARRANTIES, IMPLIED OR EXPRESSED.

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 COM485/8 Serial Interface Card contains eight independent ports and provides effective RS485 multipoint communication. The COM485/8 is 13.38 inches long and may only be installed in long slots of IBM PC/XT/AT or compatible computers. If installed in an eight-bit slot, the high interrupts (IRQ 10-15) will not be available to the COM485/8.

RS485 Balanced Mode Operation

The COM485/8 supports RS485 modes that use differential balanced drivers for increased range and noise immunity. 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".

COM485/8 also has the capability 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". The COM485/8 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. However, ports on the COM485/8 card are not restricted to the standard "comport" addresses. Continuous address selection is available anywhere within the I/O address range 100 to 3F8 hex. Our FINDBASE program will scan I/O Bus memory-mapped addresses in your computer for available addresses that can be used by the COM485/8 without conflicting with other computer resources.

A crystal oscillator is located on the card. There are x1 and x4 clock rates available. This oscillator permits precise selection of baud rate from 300 to 115,200 with the x1 setting and 1200 to 460,800 with the x4 setting.

The driver/receiver used, type 75176B, 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

COM485/8 supports Half-Duplex communications with a 2-wire cable connection. Half-Duplex allows traffic to travel in both directions, but only one way at a time. RS485 communications commonly use the Half-Duplex mode since they share only a single pair of wires.

Auto-RTS Transceiver Control

In RS485 communications, the driver must be enabled and disabled as needed, allowing all cards to share a two wire cable. The COM485/8 card controls the driver automatically. With automatic control, 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 receiver is disabled during RS485 transmissions and then enabled when the transmitter driver is disabled. The COM485/8 automatically adjusts its timing to the baud rate of the data. (Note: Thanks to this automatic control feature, the COM485/8 is ideal for use in Win95/98 applications.)

Shared IRQ Support

The COM485/8 card supports sharing of IRQ resources, and includes an on-board IRQ status register for use with operating systems that support this feature, such as Microsoft's WindowsNT. This requires one IRQ line to control all eight ports, greatly simplifying system configuration. If one IRQ is used for all 8 channels, jumpers must still be installed for each channel.

Specifications

Communications Interface

- I/O Connection: 1 25-pin male D-sub connector on board; mating cable provides eight 9-pin male D-sub connectors.
- Serial Ports: Eight compatible with 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: 300 to 460,800, Asynchronous, Type 16550 buffered UART.
- Address: Each port continuously mappable within 000 to 3FF (hex) range of AT I/O bus addresses.
- Interrupts: Jumper selectable at each port for IRQ 2-7, 10-12, 14, and 15.
- Multipoint: Compatible with RS485 specifications. Up to 32 drivers and receivers allowed on line. Serial communications ACE used is type NS16550. Driver/Receivers used are type SN75176EP.
- Receiver Input Sensitivity: ±200 mV, differential input.
- Common Mode Rejection: +12V to -7V.
- Transmitter Output Drive Capability: 60 mA, with thermal shutdown. Can drive 32-node, 2-wire, twisted-pair cable up to 4000 feet long.

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: +5 VDC at 400 mA typical, 800 mA maximum.
- Size: 13.375" long (340 mm) by 3.9" high (99 mm). If the card is installed in an 8-bit XT-style slot, the card will work but higher order IRQs 10-15 will be unavailable.)

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).

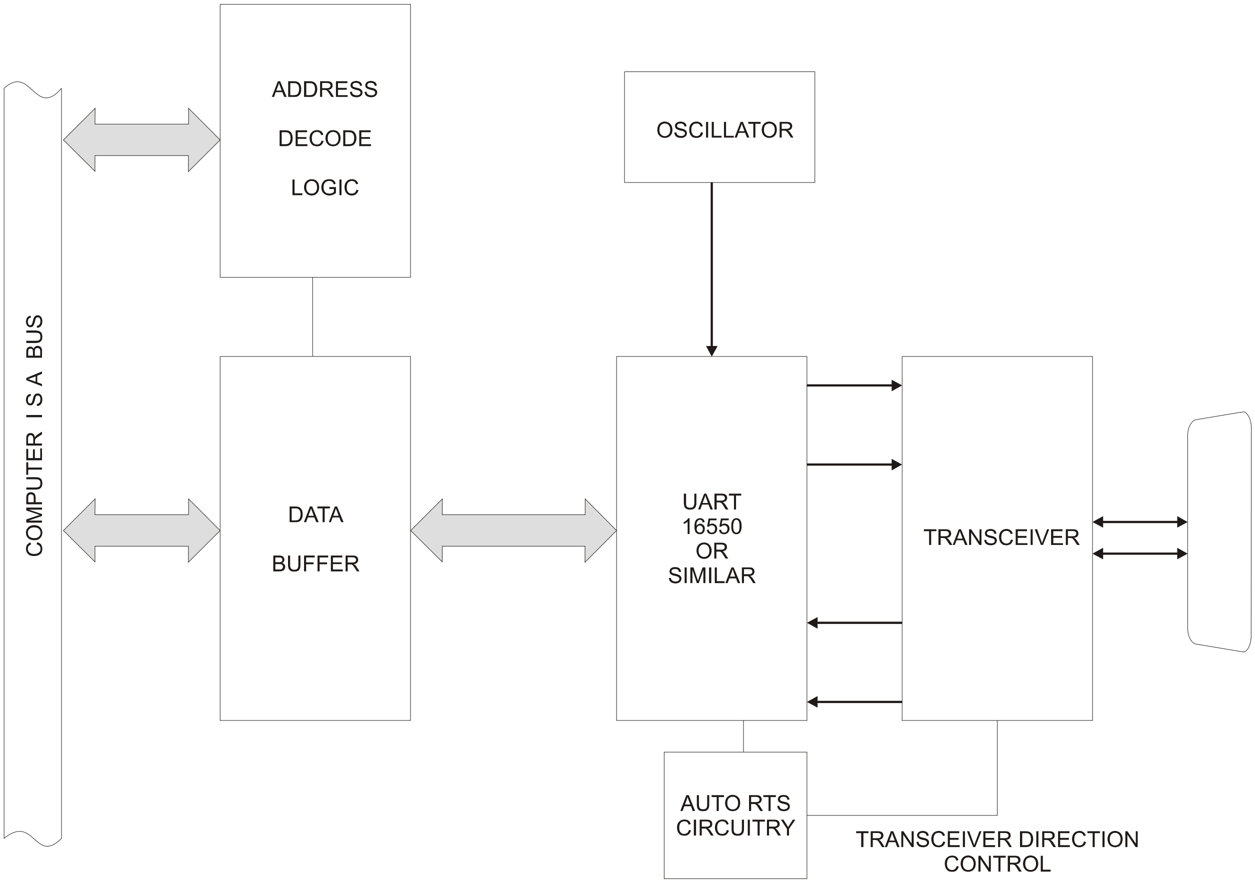

Figure 1-1: COM485/8 Block Diagram
(Only one serial channel shown)

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 D : Enter to change the active drive to the CD-ROM drive.
3. Type INSTALL Enter to run the install program.
4. Follow the on-screen prompts to install the software for this card.

WIN95/98/NT/2000

a. Place the CD into your CD-ROM drive.
b. 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 Enter.
c. 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:

DISKCOPY A: A: Enter

You will need to swap disks as requested by the system.
In a two-disk system, the command is:

DISKCOPY A: B: Enter

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 : Enter.
3. Type I N S T A L L Enter 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.

Setup.exe

This program is supplied with the COM485/8 as a tool for you to use in configuring jumpers and switches on the card. It is menu-driven and provides pictures of the card on the computer monitor. You make simple keystrokes or mouse selections to select functions. The picture on the monitor then changes to show how the jumpers or switches should be placed to effect your choices.

The setup program is a stand-alone program that can be run at any time. It does not require that the card be plugged into the computer. The program is self-explanatory with operation instructions and on-line help.

Installing the Card

The COM485/8 card can only be installed in a long slot of an IBM PC/XT/AT or compatible computer. Before installing the card, carefully read the chapters on Option Selection and Address Selection of this manual and configure the card according to your requirements. You can find an open base address with the FINDBASE program on the CD provided with the card. Finally, our setup program will lead you through the process of setting the options on the COM485/8. The setup program does not set the options on the card. These must be set by jumpers on the card.

Be especially careful with address selection. If the addresses of two installed functions (cards or devices) overlap you will experience unpredictable computer behavior.

To Install the Card

1. Turn off the computer power.
2. Remove the computer cover.
3. Remove the blank I/O backplate.
4. Install jumpers for selected options from either the OPTION SELECTION section of this manual or the suggestions of our setup software program.
5. Select the base address on the card using either the ADDRESS SELECTION section of this manual or the suggestions for COM485/8 in our FINDBASE setup software program.
6. Install the card in an I/O expansion slot.
7. Install the I/O cable.
8. Inspect for proper fit of the card and cable and tighten screws.
9. Replace the computer cover.

Note

If installed in an eight-bit slot, IRQs 10-15 will be unavailable.

A "spider" cable is provided to interface between the 25-pin connector on the card, and your system cabling. Eight individual 9-pin connectors are provided.

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) must be used for the input/output wiring.

Windows 95/98 Installation

Installation in Windows 95/98 is as above, with the added step of informing Windows of the existence of these ports. To do so, use the Control Panel “Add New Hardware” applet. Choose to manually install the device from a list, select “Port” and “Standard Port Type” (or similar, it varies depending on the version of the OS). Then choose a standard communications card. When Windows states the address and IRQ it has chosen and asks if you want to restart, say no. Repeat this step for each of the ports. Once all eight ports are installed, go to the System applet, Device Manager, select these new ports from the Ports category one at a time, and go to Properties, Resources. Select Manually Configure and change the “standard configuration” to the last one in the list (if any). Then change the address and IRQ to the ones selected earlier by Findbase etc.

Windows NT 4.0 Installation

To install the board under Windows NT, sharing IRQ:

1. Set the base address according to the manual. This example uses base address 300 hex. The

addresses set consecutively will look like:

COM485 Port	Address
A	300
B	308
C	310
D	318
E	320
F	328
G	330
H	338

2. Select the IRQ.
3. With power off, install into an ISA slot on the computer. Replace screw for hold down bracket.
4. With a DOS boot diskette that has DEBUG.EXE in the floppy drive, turn the computer on.
5. Go into BIOS setup, verify under PCI/PNP Setup that the IRQ you selected is set to ISA/LEGACY and not PCI/PNP. Save settings and exit.
6. From a DOS prompt, type DEBUG and press Enter. You will receive a "-" on the left-hand side of the screen.
7. What we are going to do is read the control port for each new COM port added. If you do not receive a 60 back after you read it then you need to re-verify the address switch settings. At the "-", type "i base+5", substituting the address for the first port, plus 5. For base address 300, the read would be "i 305" and the next port would be "i 30d". Fill in the following table with your addresses and the response you got in return.

Example	Address	Result
305		
30D		
315		
31D		
325		
32D		
335		
33D		

8. Once you have received a "60" response from reading all of the control registers, remove the floppy and reboot the machine, going into Windows NT.
9. In Windows NT, click on Start->Settings->Control Panel->Ports.

10. Set up the ports following these steps:

1. Click Add (Make note of the Com Port Number in the table below.).
2. Set the base address to the first port address on your card.
3. Set the Interrupt Request Line (IRQ) to the IRQ you selected.
4. Click OK.
5. Click on Don't Restart Now. Repeat these steps for the remaining seven ports, writing down the com port number in the table below.

COM485/8(-CE) Port	COM Port Number
A	
B	
C	
D	
E	
F	
G	
H	

11. Once all the ports have been added, close the Ports Applet and the Control Panel without restarting.
12. Click Start->Run, and type "regedt32" and click OK.

Caution

You will be making modifications to the Windows NT registry. Use extreme caution while doing so. A mistake could mean reloading the operating system.

13. Select the HKEY_LOCAL_MACHINE window. Double click on System->Current Control Set->Services->Serial in the left hand window.
14. In the left window, find "Parameters" and double click on it. Select the first Serial##### (Each system will have different numbers for the com ports based on currently installed com ports.) and verify that the DosAddress value is the same as your Port A value. If not, select the next Serial##### until you find a match.

15. Perform the following steps on each Serial##### for the board.

    1. Click on Edit-Add Value.

       i. In Value Name, type "InterruptStatus".
       ii. In Data Type, select REG_DWORD.
       iii. Click OK.
       iv. Enter in a value equal to the first port base address plus 400 for all ports. In the example of 300 hex being the first port, all of these entries would be 700. If you are using base address 100 hex, the entry would be 500.

    2. Click on Edit-Add Value.

       i. In Value Name, type "PortIndex".
       ii. In Data Type, select REG_DWORD.
       iii. Click OK.
       iv. Enter a 1 for the first port, 2 for the second port and so on until all eight ports are indexed.

16. Close regedt32 and restart Windows NT.
17. Loop Port A to Port B with the following pins.

    1. Port A pin 2->Port B pin 2
    2. Port A pin 3->Port B pin 3
    3. Port A pin 5->Port B pin 5

18. Click on Start->Programs->Accessories->Hyperterminal->Hyperterm. Set to the first com port number for the board.
19. Repeat step 18 and select the second com port number.
20. Size the window so you can see both windows. By typing in either window, you should see the characters in the other window. The board is now successfully installed.

Chapter 3: Option Selection

To help you locate the jumpers described in this section, refer to the 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 location labeled TERM applies a 120Ω load across the transmit/receive input/output for RS485 operation.

In RS485 operations where there are multiple terminals, only the RS485 ports at each end of the network should have terminating resistors as described above. Also, for RS485 operation, there must be a bias on the RX+ and RX- lines. If the COM485/8 card is not to provide that bias, contact the factory technical support.

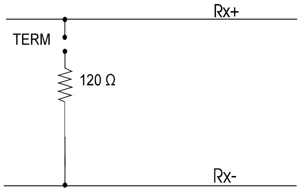

Figure 3-1: Simplified Termination Schematic

Data Cable Wiring

Signal	**Pin Connection**
Tx+ and Rx+	2
Tx- and Rx-	3
Ground	5

Interrupts

In addition to standard interrupt levels IRQ2 through IRQ7, the COM485/8 also supports higher interrupts IRQ10 through IRQ15 (Except IRQ13 and others reserved by other installed hardware). Select the desired level by installing a jumper in one of these locations for each channel. If the COM485/8 is installed in a short eight-bit slot, the higher interrupts (IRQ10 through IRQ15) will be unavailable to the card.

Please note: In WindowsNT, 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.Document id: mk:@ivt:nt40res/D15/S55FC.HTM, also available in the Windows NT Resource Kit. The text enclosed in brackets ("[]") denotes a comment.

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 board has a separate subkey under the CurrentControlSet\Services\Serial\Parameters subkey in the Registry. In each of these subkeys, you must add values for **DosDevices, Interrupt, InterruptStatus, Port Address,** 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 Windows NT Workstation Resource Kit CD.)

For example, if you have an eight-port COM485/8 board configured to use address 0x100 with an interrupt of 0x5, the values in the Registry are: [assuming every port is configured to use the same IRQ, and the addresses are configured to be consecutive and contiguous]

Serial##### Subkey:
PortAddress = REG_DWORD 0x100
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM3
InterruptStatus = REG_DWORD 0x500
PortIndex = REG_DWORD 1

Serial##### Subkey:
PortAddress = REG_DWORD 0x108
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM4
InterruptStatus = REG_DWORD 0x500
PortIndex = REG_DWORD 2

Serial##### Subkey:
PortAddress = REG_DWORD 0x110
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM5
InterruptStatus = REG_DWORD 0x500
PortIndex = REG_DWORD 3

Serial##### Subkey:
PortAddress = REG_DWORD 0x118
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM6
InterruptStatus = REG_DWORD 0x500
PortIndex = REG_DWORD 4

Serial##### Subkey:
PortAddress = REG_DWORD 0x120
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM7
InterruptStatus = REG_DWORD 0x500
PortIndex = REG_DWORD 5

Serial##### Subkey:
PortAddress = REG_DWORD 0x128
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM8
InterruptStatus = REG_DWORD 0x500
PortIndex = REG_DWORD 6

Serial##### Subkey:
PortAddress = REG_DWORD 0x130
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM9
InterruptStatus = REG_DWORD 0x500
PortIndex = REG_DWORD 7

Serial##### Subkey:
PortAddress = REG_DWORD 0x138
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM10
InterruptStatus = REG_DWORD 0x500
PortIndex = REG_DWORD 8

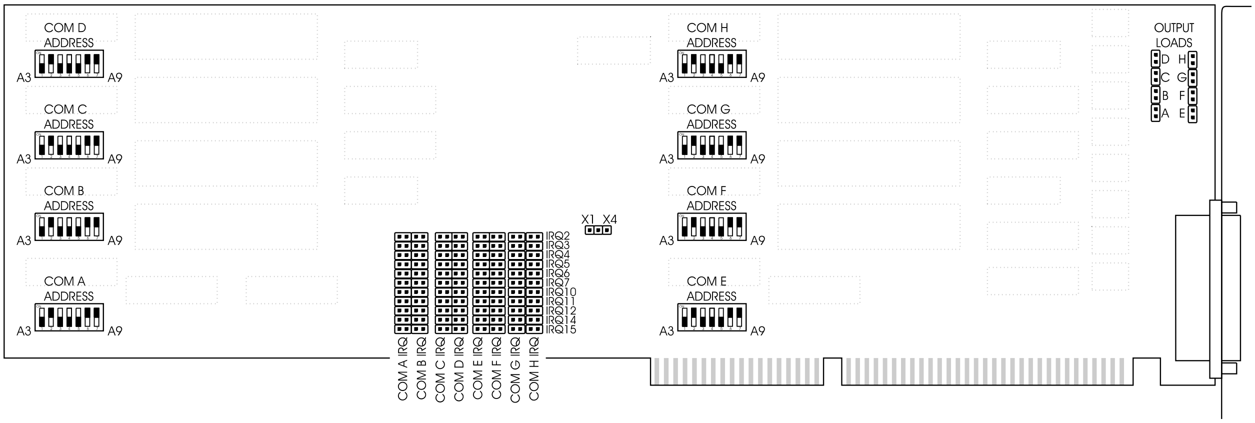

Figure 3-2: COM485/8 Option Selection Map

Chapter 4: Address Selection

Each COM485/8 port's base address can be selected anywhere within an I/O address range of 100-3F8 hex, providing that the address does not overlap with other functions. The FINDBASE base address locator program provided with your card will assist you to select a base address that will avoid this conflict. If the addresses of two installed functions overlap you will experience unpredictable computer behavior. I/O bus address switches are marked A3-A9. The following table lists the switch labels vs. the address line controlled and the relative weights of each.

Address Switch Settings	1st Digit		2nd Digit				3rd Digit
Switch Label	7	6	5	4	3	2	1
Address Line Controlled	A9	A8	A7	A6	A5	A4	A3
Decimal Weight	512	256	128	64	32	16	8
Hexadecimal Weight	200	100	80	40	20	10	8

Table 4-1: COM485/8 Address Switches

In order to read the address switch setup, assign a binary "1" to switches that are turned OFF and a binary "0" to switches in the ON position. For example, as illustrated in the following table, switch selection corresponds to binary 10 1101 1xxx (hex 2D8). The "xxx" represents address lines A2, A1, and A0 used on the card to select individual registers. See Chapter 5, Programming of this manual.

Switch Label	A9	A8	A7	A6	A5	A4	A3
Setup	OFF	ON	OFF	OFF	ON	OFF	OFF
Binary Represent'n	1	0	1	1	0	1	1
Conversion Factors	2	1	8	4	2	1	8
HEX Represent'n	2		D				8

Table 4-2: Example Address Setup

The following table lists standard DOS COM Port addresses.

Serial COM Port	IRQ	Base Hex Address	Address Switch Settings A9	A8	A7	A6	A5	A4	A3
COM1	IRQ 4	3F8	1	1	1	1	1	1	1
COM2	IRQ 3	2F8	1	0	1	1	1	1	1
COM3	IRQ 4	3E8	1	1	1	1	1	0	1
COM4	IRQ 3	2E8	1	0	1	1	1	0	1

Table 4-3: Standard DOS COM Port Addresses

Notice that only two interrupts are assigned to four serial ports. "Sharing" interrupts is not a good idea if both ports are used at the same time, unless using an operating system designed for it, like Windows NT.

IRQ Status Registers

The COM485/8 uses the industry standard 16550 UART and includes the ability to share up to eight interrupts on one IRQ line. This requires the addition of an IRQ Status Register. This register is located at COMA base address + 0x400. So, if COMA is set to 0x350, this status register is located at 0x750.

Chapter 5: Programming

Sample Programs

There are sample programs provided with the COM485/8 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 COM485/8 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:

desired baud rate = (crystal frequency) / (32 * divisor)

On the COM485/8 card, the UART clock frequency is selectable to 1.8432 MHz (X1) or 7.3728 MHz (X4). Below is a table for the popular divisor frequencies:

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 Divisor Values

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

RS485 transmission is simple to implement. The AUTO feature of the COM485/8 card automatically enables the transmitter when data are ready to send so no software enabling is required.

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.	RS485 Functions
1	
2	Tx+ and Rx+
3	Tx- and Rx -
4	
5	GND
6	
7	
8	
9	

Table 6-1: Connector Pin Assignments

Note

If connections are to be made to a COM485/8-S04 (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

Introduction

Working with RS485 devices is not much different from working with standard RS232 serial devices and this standard overcomes deficiencies in the RS232 standard. First, the cable length between two RS232 devices must be short; less than 50 feet. Second, many RS232 errors are the result of noise induced on the cables. The RS485 standard permits cable lengths up to 4000 feet and, because it operates in differential mode, it is more immune to induced noise.

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 always 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. 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)	TERM	2V	
	jumpers in	-2V	
Driver Output Resistance			50Ω
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			4KΩ

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 COM485/8 card. Termination resistors for the RX+ and RX- lines are provided on the card and are placed in the circuit when you install the LOAD (LD) jumpers. (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.

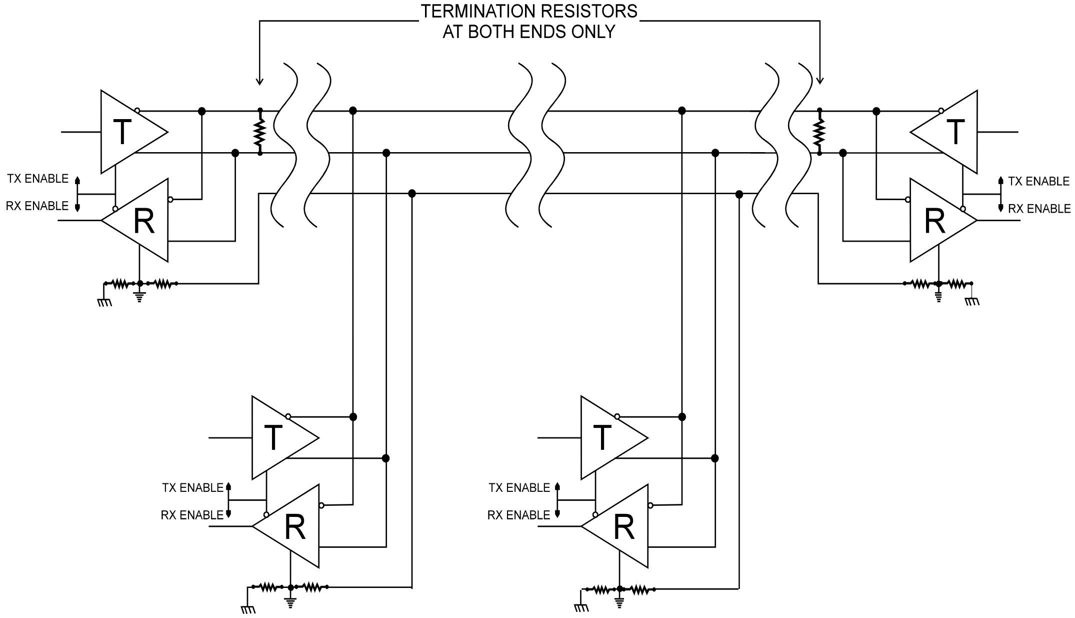

Figure A-1: Typical RS485 Two-Wire Multidrop Network

Customer Comments

If you experience any problems with this manual or just want to give us some feedback, please email us at: ***manuals@accesio.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.accesio.com