



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

MODEL COM422/4

USER MANUAL

Notice

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

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

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

Warranty

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

Terms and Conditions

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

Coverage

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

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

Equipment Not Manufactured by ACCES

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

General

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

Table of Contents

Chapter 1: Introduction	1-1
Specifications	1-1
Chapter 2: Installation	2-1
CD Installation	2-1
3.5-Inch Diskette Installation	2-1
Directories Created on the Hard Disk	2-2
Installing the Card	2-4
Chapter 3: Option Selection	3-1
Chapter 4: Address Selection	4-1
Chapter 5: Programming	5-1
Sample Programs	5-1
Windows Programming	5-1
Reception	5-3
Transmission	5-4
Chapter 6: Connector Pin Assignments	6-1

List of Figures

Figure 1-1: COM422/4 Block Diagram	Page 1-2
Figure 3-1: Simplified Termination Schematic	Page 3-1
Figure 3-2: COM422/4 Option Selection Map	Page 3-2
Figure 1-1: COM422/4 Block Diagram	Page 1-2
Figure 3-1: Simplified Termination Schematic	Page 3-1
Figure 3-2: COM422/4 Option Selection Map	Page 3-2

List of Tables

Table 4-1: Standard Address Assignments for 286/386/486 Computers	Page 4-1
Table 5-1: Baud Rate Divisors	Page 5-2
Table 6-1: Connector Pin Assignments	Page 6-1

Chapter 1: Introduction

The COM422/4 card (for RS422 serial communications) is 7.82 inches long and can be installed in either short or long ISA slots of IBM PC/XT/AT or compatible computers. If installed in an eight-bit slot, interrupts 10 through 15 will not be available.

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% compatibility with the original IBM serial port. However, use of the COM422/4 card is not restricted to the standard DOS addresses COM1-COM4. Continuous address selection is available anywhere within the I/O address range 100 to 3FF hex.

A crystal oscillator is located on the card. This oscillator ensures precise selection of baud rate from 100 to 115.2K. Other crystals for higher baud rate ability are available on special order. If your COM422/4 is so modified, then there will be an Addendum sheet in the front of this manual.

The output transceiver used, the new generation type 75176, is capable of driving extremely long communication lines at high baud rates. It is capable of driving up to 60mA on balanced lines and receiving inputs as low as 200mV differential signal superimposed on common mode noise of maximum -7V/+12V. In case of communication conflict, the transceivers feature thermal shutdown.

Specifications

- **Multipoint:** Compatible with RS422 specifications. Up to 32 receivers allowed on line. Serial communications ACE used is type NS16450. Transceivers used are type 75176.
- **Input Common Mode Voltage:** -7V to +12V.
- **Receiver Input Sensitivity:** $\pm 200\text{mV}$, differential input. Thermal shutdown protection.
- **Output Drive Capability:** $\pm 60\text{ mA max.}$, differential. Thermal shutdown protection.
- **Baud Rate:** 50 to 115.2K baud. Crystal oscillator provided.
- **Address:** Continuously mappable within I/O address range 000 to 3F8 hex, eight bytes.

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 250 mA typical, 500 mA max.
- Size: 7.82" long. Can mount in half- or full-size slot.

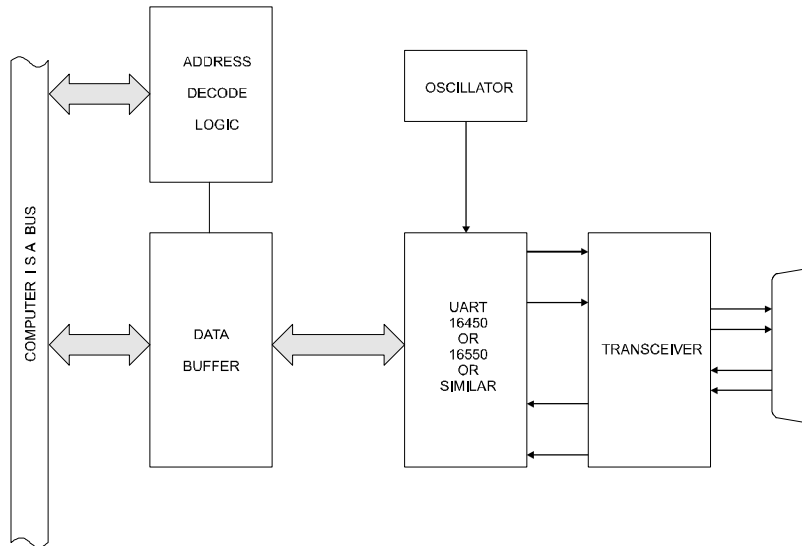


Figure 1-1: COM422/4 Block Diagram

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

- 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 `[?]`.
- 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:

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

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 COM422/4 card can be installed in either an 8-bit or 16-bit slot of an IBM PC/XT/AT or compatible computer. Before installing the card, carefully read the Option Selection and Address Selection chapters of this manual and configure the card according to your requirements. Use the FINDBASE program provided with the card to select the base address for the card. Use the program called SETUP to select other options.

Be especially careful with address selection. If the addresses of two installed functions overlap, you will experience unpredictable computer behavior.

To Install the Card

1. Turn off computer power.
2. Remove the computer cover.
3. Remove the blank mounting bracket.
4. Install jumpers for selected options. See section 3.
5. Select the base address on the card. See section 4.
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.

Four DB9 connectors are provided for I/O connections: two on the card mounting bracket (for ports A and B) and two on a separate bracket (for ports C and D). The COM422/4-S01 has a single DB25 connector on the card mounting bracket and an external "spider" cable includes four female DB9 connectors for I/O.

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

Chapter 3: Option Selection

Refer to Figure 3-2, Option Selection Map on the next page when reading this section of the manual. Card operation is determined by jumper installation as described in the following paragraphs.

Terminations and Bias

A transmission line should be terminated at the receiving end in its characteristic impedance. Installing a jumper at the locations labeled COM 422 LD and COM485 LD apply a 120 Ω load across the input and output for RS422 mode.

Data Cable Wiring

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

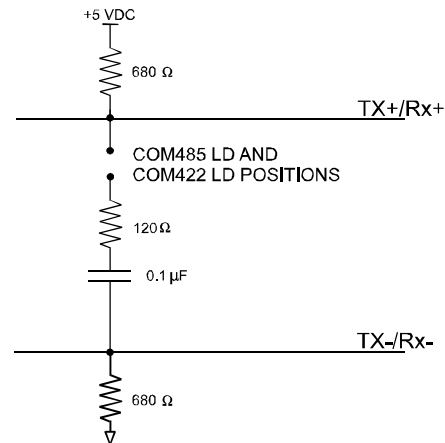


Figure 3-1: Simplified Termination Schematic

Interrupts

In addition to standard interrupt levels IRQ2 through IRQ7, the COM422/4 also supports higher interrupts IRQ10 through IRQ15 (Except IRQ13 which is reserved by other hardware). Select the desired level by installing a jumper in one of these locations. If the COM422/4 is installed in an eight-bit slot, the higher interrupts (IRQ10 through IRQ15) will be unavailable to the card.

If you intend to use the IRQ sharing feature of operating systems like WindowsNT, place the IRQ jumpers for all ports that will share the IRQs on the same IRQ level. As many or as few ports as desired can share IRQs under WindowsNT.

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, documentid:mk:@ivt:nt40res/D15/S55FC.HTM, also available in the WindowsNT 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 card has a separate subkey under the CurrentControlSet\Services\Serialssubkey 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.)

COM422/4 Manual

For example, if you have a four-port COM422/4 card 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]

Serial2 Subkey:

PortAddress = REG_DWORD 0x100
 Interrupt = REG_DWORD 5
 DosDevices = REG_SZ COM3
 InterruptStatus = REG_DWORD 0x500
 PortIndex = REG_DWORD 1

Serial4 Subkey:

PortAddress = REG_DWORD 0x110
 Interrupt = REG_DWORD 5
 DosDevices = REG_SZ COM5
 InterruptStatus = REG_DWORD 0x500
 PortIndex = REG_DWORD 3

Serial3 Subkey:

PortAddress = REG_DWORD 0x108
 Interrupt = REG_DWORD 5
 DosDevices = REG_SZ COM4
 InterruptStatus = REG_DWORD 0x500
 PortIndex = REG_DWORD 2

Serial5 Subkey:

PortAddress = REG_DWORD 0x118
 Interrupt = REG_DWORD 5
 DosDevices = REG_SZ COM6
 InterruptStatus = REG_DWORD 0x500
 PortIndex = REG_DWORD 4

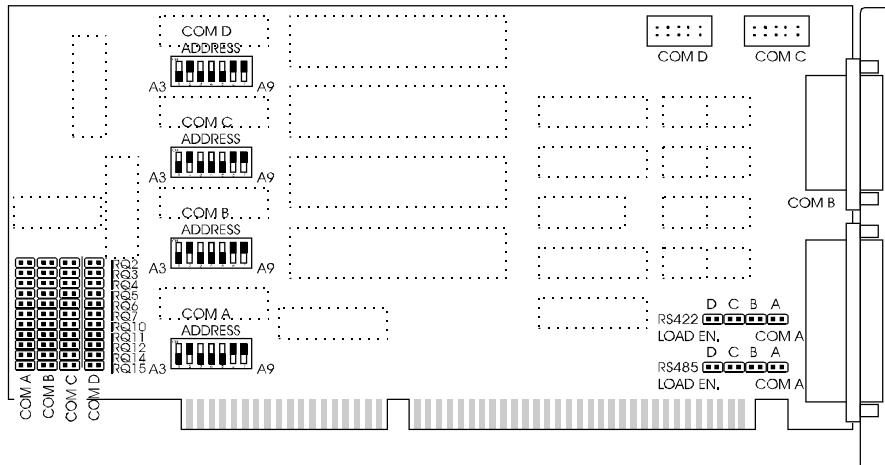


Figure 3-2: COM422/4 Option Selection Map

Chapter 4: Address Selection

The COM422/4 Serial Interface card base address can be selected anywhere within an I/O address range 100-3FF hex in ATs (except 1F0 through 1F8) and 200-3FF in XTs, providing that the address does not overlap with other functions. (Note: 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 any problem.

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	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)

Table 4-1: Standard Address Assignments for 286/386/486 Computers

Address Setup switches are marked A3-A9. In order to configure the desired address, assign "0" to all Address Setup switches turned OFF and assign "1" to all Address Setup switches turned ON. These 1's and 0's are a binary representation of the base address. This binary number is then converted to a hexadecimal number.

Switch Label	A9	A8	A7	A6	A5	A4	A3
Addr. Line Contr'ld	A9	A8	A7	A6	A5	A4	A3

For example, as illustrated in the following table, switch selection corresponds to binary 10 1101 1xxx (or 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	ON	OFF	ON	ON	OFF	ON	ON
Binary Represent'n	1	0	1	1	0	1	1
Conversion Factor	2	1	8	4	2	1	8
Hex Represent'n	2		D				8

Review the Address Selection Table carefully before selecting the card address. If the addresses of two installed functions overlap you will experience unpredictable computer behavior.

Chapter 5: Programming

Sample Programs

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

Windows Programming

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

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

See the documentation for your chosen language for details.

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

Initialization

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

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

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

$$\text{desired baud rate} = (\text{crystal frequency}) / (16 * \text{divisor})$$

On the COM422/4 card, the crystal frequency is 1.8432 MHz. Commonly used divisors are: 12 for 9600 baud, 48 for 2400 baud, and 96 for 1200 baud. In C, the code to set the chip to 9600 baud is:

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


The UART clock frequency is 1.8432MHz. The following table lists popular divisor frequencies.

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

*These are theoretical maximums based on typical conditions and good quality cables based on the EIA 485 and EIA 422 standard for balanced differential drivers.

Table 5-1: Baud Rate Divisors

The second step in initializing the UART 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 0's.

Bit 7 is the DLAB. It must be set to 0 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. This register controls functions on some cards. Bit 1 is the Request to Send (RTS) control bit. This bit should be left low until transmission time. 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 this is 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 available to be read from the chip.

A simple polling loop must continuously check this bit and read in data when it becomes available. Polling is not effective at very high data rates because the program cannot do anything else when it is polling or data could be missed. The following is a code fragment that implements a polling loop and uses a value of zero as an end-of-transmission marker:

```
do

    while (!(inportb(BASEADDR +5) & 1)); /*Wait until data ready*/
    data[i++]=inportb(BASEADDR);

    while (data[i]!=0); /*Reads the line until null character rec'd*/
```

Interrupt-driven communications should be used whenever possible and is required for data speeds. Writing an interrupt-driven receiver is not much more complex than writing a polled receiver. However, care should be taken when installing or removing your interrupt handler because there is a danger of writing the wrong interrupt, or disabling the wrong interrupt, or even 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. As sample handler, written in C, is as follows:

```
do

    readback = inportb(BASEADDR +2);
    if(readback & 4) /*Readback will be set to 4 if data are available*/
        data[i++]=inportb(BASEADDR);

    while(readback!=1);
    outportb(0x20,0x20); /*Write EOI to 8259 Interrupt Controller*/
    return;
```

The preceding examples accept only a single character input. The code would need to be modified to handle larger data structures.

Transmission

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); /*SetRTSbit without
altering states of other bits*/
while(*data) /*While there is data to send*/
{
while(!(inportb(BASEADDR +5)&0x20)); /*Wait until transmitter is empty*/
outportb(BASEADDR,*data);
data++;
}
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. The mating connector is AMP type 17D-E9S or equivalent. We recommend using vinyl-jacketed, multiple twisted-pair cable.

Pin No.	Assignment
1	Receive Line - (RX-)
2	Transmit Line + (TX+)
3	Transmit Line - (TX-)
4	
5	Signal Ground
6	
7	
8	
9	Receive Line + (RX+)

Table 6-1: Connector Pin Assignments

Customer Comments

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



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