brain boxes
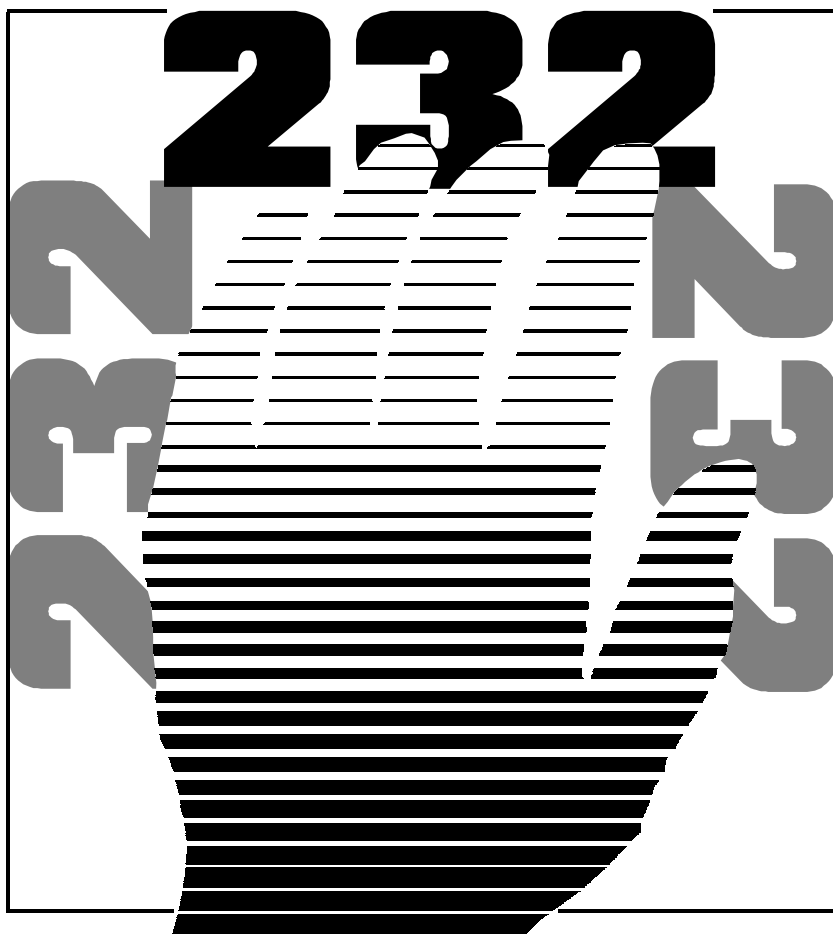
232

AT Triple RS232

2.0 EDITION Feb 1997

Guarantee.

FULL 36 MONTHS GUARANTEE.

BRAIN BOXES guarantee your Serial Port Card for a full 36 months from purchase, parts and labour, provided it has been used in the specified manner. In the unlikely event of failure return your interface to BRAIN BOXES or to your Dealer, with proof of purchase, who will determine whether to repair or replace this product with an equivalent unit.

COPYRIGHT.

Serial Solutions are designed, manufactured, and supported by

BRAIN BOXES.

Unit 3F, Wavertree Boulevard South,
Wavertree Technology Park,
Liverpool, L7 9PF,
England.
Telephone: 0151-220 2500
Fax: 0151-252 0446
E-mail: help@brain_boxes.cybase.co.uk
Web: www.connect.org.uk/merseymall/brain_boxes/

ACKNOWLEDGEMENTS.

BRAIN BOXES is a trademark of BRAIN BOXES.
IBM, Commodore, COMPAQ, Olivetti, AMSTRAD, Hewlett Packard, H.P. and EPSON are trademarks of the relevant companies. OS/2 and Microchannel Architecture are trademarks of IBM. Windows is a trademark of Microsoft.

AT Triple RS232 Manual
Outline contents

Chapter 1. OPTIONAL Serial Solution Software.
Chapter 2. AT Triple RS232 Installation Guide
Chapter 3. Serial Utility Disk & Programming Guide.
Chapter 4. RS232 Port Cabling.
Index.

The Layout of this manual

Chapter 1, Serial Solution Software, is an overview of the optional, ideal companion software package for our range of serial port cards. Buy it from your dealer now!

Chapter 2, AT Triple RS232 Installation Guide, shows you how to configure and install the AT Triple RS232 card in an expansion slot of an AT compatible computer. After a brief discussion of the RS232 interface standard, details of serial port address selection, and Interrupt line selection are given.

Chapter 3, Serial Utility Disk & Programming Guide, describes the sample programs on the disk supplied with the card, e.g. the ADDRCOM3/4, COMTEST, COMM, and RS232 sample programs. The chapter concludes with a detailed discussion of how to program the 16450 serial port chip.

Chapter 4, RS232 Port Cabling, gives details of both standard 25 pin D and 9 pin RS232 pin outs, cabling and connections, showing how to interconnect the serial ports of two PC's, make up a loop back connector for self tests and how to wire a 9 to a 25 pin adapter are given.

The **Index** covers the complete contents of this manual

CHAPTER 1 - Serial Solutions Software

CHAPTER 2 - AT Triple RS232 Installation Guide.

CHAPTER 3 - Serial Utility Disk and Programming

CHAPTER 4 - RS232 Port Cabling

THIS PAGE LEFT INTENTIONALLY BLANK

CHAPTER 1
SERIAL
SOLUTION
SOFTWARE.

Introduction

This chapter is a brief advertisement for the OPTIONAL Serial Solutions software package, this is purchased SEPARATELY and is available from YOUR DEALER.

Introducing Serial Solution Software

The perfect partner for any Serial Port is Serial Solutions Software! Serial Solutions is a fully featured suite of programs designed to squeeze the most from PC serial communications.

Serial Solutions is made up of the following components:-

Serial Solutions for DOS

Serial Solutions for Windows 3.x

Serial Solutions for Windows 95

All the Serial Solutions drivers have the following features:-

Drivers for PC standard UARTs e.g. 8250 and 16450

a) Drivers for PC FIFO UARTs e.g. 16550 as well as the new improved 32 byte 16650 and 64 byte 16750 UARTs.
b) Support for shared interrupt cards e.g. our Quad 232, Lynx 232 and Quad 422 cards.
c) Support for any mix of RS232, RS422 and RS485 handshake schemes.
d) Support for wider range of Baud rates and for more than 4 serial ports.

Serial Solutions For DOS

Serial Solutions for DOS consists of the following programs:-

NewCOM.sys   A device driver, it supports COM1 to COM16,

allowing 16 serial ports to be used under DOS. It also includes an interrupt handler for enhanced performance with user definable buffer sizes. Accessible from all DOS languages. It is the heart of the Serial Solution.

It has extensive handshaking support, implementing both hardware handshaking using any combination of the DTR, DSR, CTS, RTS, and DCD lines, and a software handshake using the XON/XOFF protocol.

NewCOM24.sys   A device driver providing support for 24 ports.

NewCOM32.sys   A device driver providing support for 32 ports.

NewMode.exe   A replacement for the DOS 'mode com...' command. NewMode is used to set the serial parameters, including the port address, IRQ line used, the baud rate, parity and data and stop bit options.
e.g. NEWMODE COM5:38400,E,7,1 01A0 7
Baud rates supported are from 110 baud to 115,200 baud! Included is a very handy query mode that reports the settings of the various serial ports. Flexible and fast!

EASY programs.   The EASY disk contains short, simple to understand and use EASYBAS, EASYC and EASYPAS programs, providing straight forward, file type i/o to serial ports with debug information. Use these FIRST, base your sample applications on them.
Source code, make files and compiled ready to run programs supplied.

TERM programs   A suite of larger terminal emulation programs written in C (Cterm), Assembly language (Aterm), Pascal (Pasterm), BASIC (BASterm)and FORTRAN (FORterm) show how to access the NEWBIOS routines as well

as the simple file i/o to ports. They contain many lines of code and are thus harder to grasp. They demonstrate in depth serial port programming in a variety of languages but they are also useful tools for using serial devices.

Serial Solutions For Windows 3.x

Serial Solutions for Windows 3.x works with Windows 3.0, 3.1 and 3.11 as well as Windows For Workgroups 3.11.

Serial Solutions for Windows 3.x consists of the following programs:-

Setup.exe	The install routine for the package.
Port.DLL	Enhanced Control Panel applet. Allows configuration of extra serial ports from the Windows Control Panel. Supports single as well as multiport cards using shared interrupts.
BbLynx.drv	Replacement for COMM.DRV.
LynxAPI.dll	Enhancement to the Windows Comms API's allowing support for more than 9 ports.
Term.exe	Terminal program.
EasyCWIN	C source code, project files and ready to run.exe program for an easy to understand Windows terminal program. Learn how to write Windows comms apps correctly the easy way.

Serial Solutions For Windows 95

Windows 95 has an improved communication API and directly supports up to 255 ports. Our Windows 95 driver supports the shared interrupt mechanism used on our multiport cards. Serial Solutions for Windows 95 consists of the following programs:-

Bbcommsp.inf	The information file to aid the installation process "Have Disk...."
Bbcommui.dll	The DLL and..
BBcomm.vxd	..the virtual device driver providing the shared

interrupt handler and despatch routines.

The Comtest.exe Program

Comtest is a short but invaluable program that is used to check that the serial port at a particular i/o address is functioning correctly and is connected to the particular IRQ line. The program correctly identifies the UART type e.g. non FIFO, 16550 FIFO, 16550AF FIFO and the improved 16650 32 byte FIFO and 16750 64 byte FIFO's. By employing the built in loop back capability of the PC serial port chip, a full test of the baud rate generator, transmitting and receiving buffer, parity enable and start stop bit is performed. There is no need for a second serial port or a cable when using this utility.

Complete Documentation and Technical Backup.

We believe in supplying complete documentation with every package we sell. The Serial Solution Software Package is no exception, it has an attractive manual in an A5 binder with slip case, containing over 150 pages of in-depth technical detail with comprehensive indexes and table of contents. We guarantee your Serial Solution Software package for a full 12 months from purchase. A complete technical backup service is available to ensure that you get the maximum performance out of your investment.

CHAPTER 2
AT TRIPLE RS232
INSTALLATION
GUIDE.

Introduction.

This chapter explains how to configure and install the AT Triple RS232 card in an PC compatible, giving details for address and IRQ jumper selection.

The half size RS232 card will fit into both long and short slots, in 8 bit PC or 16 bit AT slots and will work happily in any PC compatible including Pentium Pro class machines up to 200Mhz.

AT Triple RS232 Card Features.

* Three independent RS232 Serial ports.
* Reliable communications up to 50 feet, 15m, and beyond!
* 100% PC Compatible serial port up to 115,200 baud.
* Jumper selectable serial port address, COM1- COM8.
* Jumper selectable interrupt level IRQ 2-7, 10-12, 14 & 15.
* CTS and RI signals Full modem control TXD, RXD, DSR, CD, DTR, RTS,.
* Fully double buffered for reliable asynchronous operation.
* High speed integrated circuitry ensures operation with fast PC's e.g. 200Mhz Pentium Pro WITHOUT extra wait states.
* 16550 FIFO as standard provides 16 byte input and 16 byte output buffer each port.

Serial Port Features:-
.

Baud Rate:	50 Baud to 115200 Baud.
Word Length:	5, 6, 7 or 8 bits.
Parity:	Even, Odd, None, Mark or Space.
Start Bit:	1 start bit always sent.
Stop Bits:	1, (1.5 for 5 bit data word length) or 2.
Clock Input:	1.8432 MHz.

Card Hardware Specifications:-
AT Dual Dimensions: 5.10 x 3.50 in, 138 x 89 mm
I/O Connection: Serial Port 1: 9 pin Male D type.
                            Serial Port 2: 9 pin Male D type.
                            Serial Port 3: 9 pin Male D type.

The RS232 Standard.

The RS232 standard is ancient in computer industry terms. Introduced in 1962, it is now widely established. RS232 is a slow speed, short distance, single ended transmission system (i.e. only one wire per signal). Typical RS232 maximum cable length is 50 feet with a maximum data rate of 20K bits per second.

Figure 2-1. RS232 Point To Point Connection.
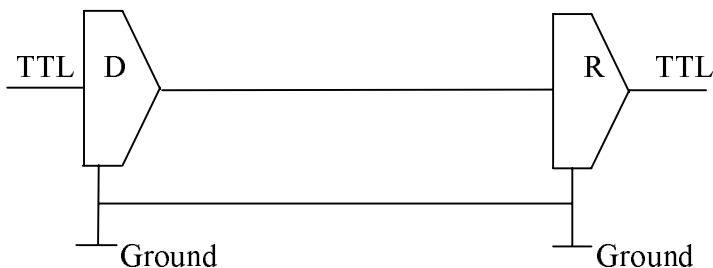
RS232C Standard 1 Driver 1 Receiver	
Line Length	Max Data Rate
50 Feet = 15m	20 Kbits/sec

Configuring The Triple RS232 Card.

In the state it leaves our factory, the Triple RS232 card is ready to plug straight into a PC compatible computer.
Unless there is GOOD REASON, do not alter its default setting. However, due to the presence of other serial ports in the PC, your card may need configuring to suit your setup.

Figure 2-2. AT Triple RS232 Card Layout.

Com # 1 Port #1
Com # 2 Port #2
Com # 3 Port #3
987654321

Port 1

Port 2

IRQ #1   IRQ Port #1
IRQ #2   IRQ Port #2
IRQ #3   IRQ Port #3
11111234567
54210

Port 3
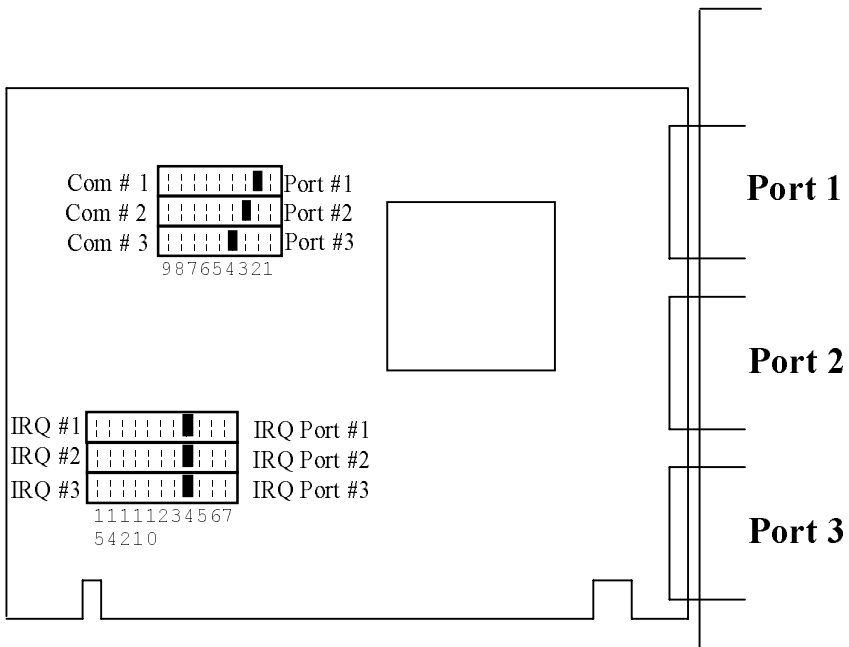
Settings for DOS Software.

Most users will require that the IRQ is set to correspond to the interrupt used by their serial communications software.
Note, that a serial port that is USING interrupts must not share the same IRQ line as another serial port that is USING interrupts AT THE SAME TIME, otherwise some interrupts will be missed. IE no two ports may use the same IRQ line simultaneously.

Some serial communications software, especially older versions, assume that COM1 and COM3 both use IRQ4 and that COM2 and COM4 both use IRQ3. Other older packages assume that COM1 is on IRQ4 and that COM2, COM3 and COM4 all use IRQ3. Both kinds of older packages will therefore not allow simultaneous interrupt driven i/o on all four ports. It may be necessary to upgrade your serial communications software.

Newer, more up to date, serial communications programs are configurable. They allow the user to specify the i/o address and the interrupt used by the serial port. Procomm and many other packages are configurable in this way.

The Ideal Situation.

If your serial communications packages are configurable for I/O address and interrupt then set all the serial ports in your system so that each serial port has its own Separate I/O address, and interrupt not used by any other device in the PC.

Serial Port Configuration.

Serial Port Jumpers.

Each serial port drives a 9 pin D connector Serial Port 1, Serial Port 2 and Serial Port 3 as shown in Figure 2-2. There are 2 sets of jumpers associated with each serial port. The I/O address select jumper on the upper left of centre and the IRQ jumper select on the lower left hand side of the card. Each serial port is completely independent of the other. A port may be completely disabled by removing both I/O address and interrupt jumpers. The port pin outs are given below.

Figure 2-3. Serial Port 1 Pin Outs.
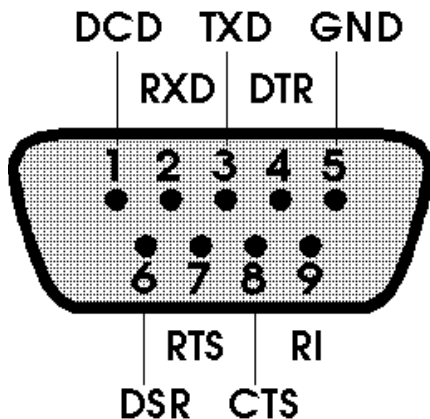
PIN	SIGNAL	DESCRIPTION	DIRECTION
1	DCD	DATA CARRIER DETECT	INPUT TO PC
2	RXD	RECEIVED DATA	INPUT TO PC
3	TXD	TRANSMITTED DATA	OUTPUT FROM PC
4	DTR	DATA TERMINAL READY	OUTPUT FROM PC
5	GND	GROUND	
6	DSR	DATA SET READY	INPUT TO PC
7	RTS	REQUEST TO SEND	OUTPUT FROM PC
8	CTS	CLEAR TO SEND	INPUT TO PC
9	RI	RING INDICATOR	INPUT TO PC.
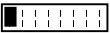
Serial Port Address Selection.

The position of the movable jumper on the address jumper block, upper left side of the card, determines the i/o address of the serial ports, configuring the card as COM1, COM2, COM3, COM4 or COM5- COM8.

The default, factory set position, for serial port 1 is COM2, for serial port 2 is COM3 and for serial port 3 is COM4. Depending on how many other serial port cards are already installed in the PC the i/o address may need to be changed.

All PCs detect the presence of COM1 and COM2. Most modern PCs have a BIOS that automatically detects the presence of COM3 and COM4, usually displaying the ports found on screen at power on time. If your PC does not automatically detect these ports please see the note later about the ADDRCOM3 and ADDRCOM4 utility programs.

Figure 2-4. Serial Port Address Jumper Block.

```
 JUMPER          PORT  ADDRESS      NORMAL IRQ ALLOCATION.
87654321
[█|:|:|:|:|:|:|:]    COM8  01B8hex    IRQ 3.

[:|█|:|:|:|:|:|:]    COM7  01B0hex    IRQ 3.

[:|:|█|:|:|:|:|:]    COM6  01A8hex    IRQ 3.

[:|:|:|█|:|:|:|:]    COM5  01A0hex    IRQ 3.

[:|:|:|:|█|:|:|:]    COM4  02E8hex    IRQ 3.      DEFAULT SERIAL 3

[:|:|:|:|:|█|:|:]    COM3  03E8hex    IRQ 3.      DEFAULT SERIAL 2

[:|:|:|:|:|:|█|:]    COM2  02F8hex    IRQ 3.      DEFAULT SERIAL 1

[:|:|:|:|:|:|:|█]    COM1  03F8hex    IRQ 4.
```

Serial Port IRQ Interrupt Jumper Selection.

By referring to the serial card locate the serial port IRQ interrupt jumper block at the lower left of the board. The movable jumper on the jumper block is used to specify which hardware interrupt is to be generated by the PC serial port.

Figure 2-5. Serial Port IRQ Jumper Block.

```
11111
54210234567              NORMAL USE OF INTERRUPT.

|!|!|!|█|!|!|!|!|       IRQ2 EGA/VGA GRAPHICS CARD, Usually Free
                        Used by only very few graphics cards.

|!|!|!|!|█|!|!|!|       IRQ3 COM 2-8. DEFAULT for SERIAL 1 - 3


|!|!|!|!|!|█|!|!|       IRQ4 COM 1.


|!|!|!|!|!|!|█|!|!|     IRQ5 PRINTER PORT#2 OK for DOS & Windows
                        Avoid for OS/2,Windows NT & Novell

|!|!|!|!|!|!|!|█|!|     IRQ6 FLOPPY DISK STATUS, AVOID!.


|!|!|!|!|!|!|!|!|█|     IRQ7 PRINTER PORT#1 OK for DOS & Windows
                        Avoid for OS/2,Windows NT & Novell

|!|!|!|█|!|!|!|!|!|     IRQ10 Usually Free
                        Good for COM3

|!|!|█|!|!|!|!|!|!|     IRQ11 Usually Free
                        Good for COM4

|!|█|!|!|!|!|!|!|!|     IRQ12 POINTING DEVICE, Usually Free
                        Free when mouse is on a COM port.

|█|!|!|!|!|!|!|!|!|     IRQ14 IDE HARD DISK. Usually In Use.
                        Free when SCSI disks are used.

|█|!|!|!|!|!|!|!|!|     IRQ15 Usually Free
```

Note:
   IRQ0 & 8, timer & clock interrupts, not on expansion bus.
   IRQ1, keyboard interrupt, is not on expansion bus.
   IRQ9, best left unexplained, is not on expansion bus.
   IRQ13, maths coprocessor int., is not on the expansion bus.

Microsoft Windows Version 3.x.

The Windows environment now supports up to four serial ports, RS232, RS422 and RS485 etc.

If more than 2 serial ports are installed then the default assignments assumed by Windows need to be changed. Windows assumes that both COM1 and COM3 are on IRQ4 and that both COM2 and COM4 are on IRQ3. Since the PC bus does not allow interrupt sharing, these pairs of ports cannot be used simultaneously, seriously limiting the flexibility of Windows 3.1.

To obtain trouble free mix and match of the COM ports, assuming that your PC has only 1 serial port:

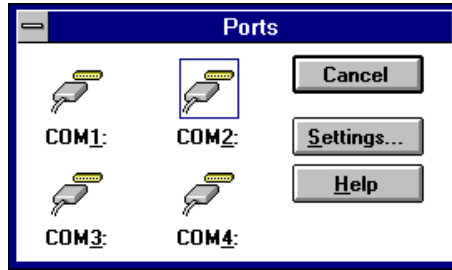● Set COM3 and COM4 interrupts to ones not used by other devices on your system, say IRQ10 and IRQ11.

● From the Windows Program Manager select **Main**. Then double click the **Control Panel** icon (the following example is of the AT Triple RS232 with default settings).

● From the Control Panel window select **Ports**

● From the Ports window select **COM2.** Then click the **Settings** button.

● From the COM2 window click the **Advanced** box.
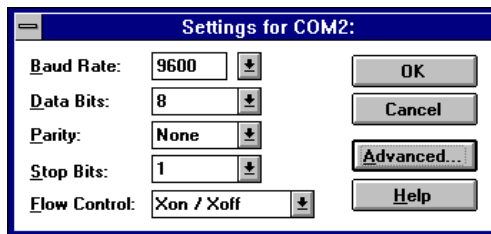
● From the Advanced Settings for COM2: window select
Interrupt Request Line(IRQ):

> ● Scroll through the list of interrupts and select IRQ3 by clicking on 3.
> ● Return from the Advanced Settings for COM2: window by clicking on **OK**
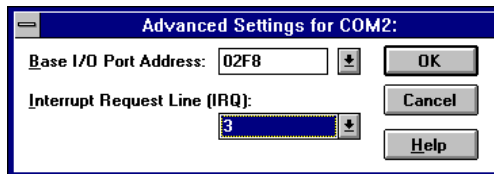
Windows displays a Systems Setting Change, saying "You will need to quit and restart Windows so that the new setting(s) can take effect."

● **Either:-** Click on Restart Now and let windows restart itself able to run with the new configuration.
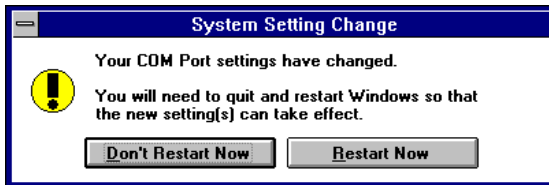
Or:- Click on Don't Restart Now and step back though the various windows and to configure other ports or run other applications. The new settings for the COM ports will only take effect the next time you boot up Windows from cold.

Please Note:
Microsoft documentation for Windows STRONGLY recommends the use of 16550 FIFO chips on serial ports. The Triple RS232 card provides this as standard.

Microsoft Windows 95.

The Windows 95 environment now supports up to 255 standard serial ports, RS232, RS422 and RS485 etc.

If more than 2 serial ports are installed then the default assignments assumed by Windows 95 will most probably need to be changed. Windows 95 assumes that COM1 is on IRQ4 and that COM2 is on IRQ3. For ports above COM2, Windows 95 makes an attempt to detect the interrupt line being used. It is not always successful. Obviously having more than just a couple of standard serial ports in your PC will lead to Interrupt Request Line resource conflicts, Windows 95 has gotten around this problem by letting multiple ports be defined as sharing the same interrupt. Since the PC bus does not allow interrupt sharing, any ports defined as sharing interrupts cannot be used simultaneously, and even using them individually slows down performance, as Windows 95 must search through all defined ports using that interrupt to determine which one has valid data arriving. Although an improvement on

Windows 3.x, the solution is still not ideal.

To obtain trouble free mix and match of the COM ports, assuming that only 1 Serial Port already exists on your PC:

● Turn off the computer, plug in the card and restart.

● Click on the **Start** button and select **Settings**, then **Control Panel**
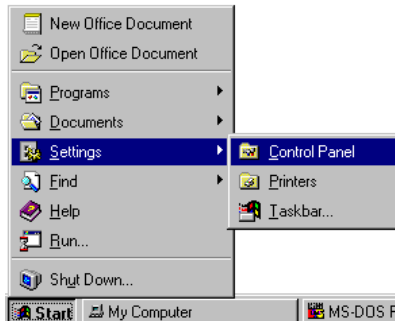
● Select **Add New Hardware**
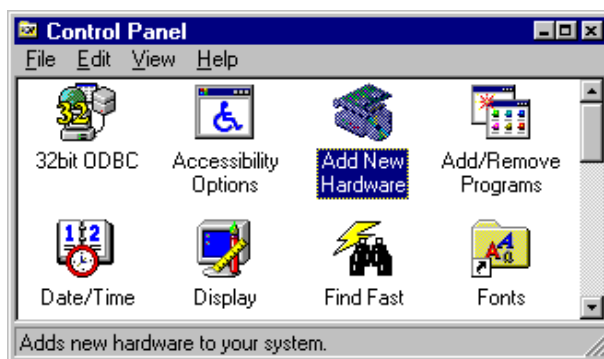
● Click **Next**

If you are installing the AT Triple RS232 as above COM4 then step to the next section.

Installing as COM4 and Lower.

If you are installing the AT Triple RS232 as COM4 and lower then continue.

● Click the **Yes** option button and then **Next**.

Windows will prompt you with the following dialogue box,

● Click **Next.**

Windows will now start detecting your newly installed hardware. This may take some time, so be prepared to wait. The following Window will be shown during detection, with a status bar indicating progress.

When Detection has finished, Windows will inform you that it has finished detecting your new hardware.
● Click on **Details** to see what Windows has detected,

If you have set up the card as COM4 or lower then the view of detected ports should look somewhat like the following:

For ports of up to COM4 Windows generally detects the correct settings, if you wish to check the settings before rebooting, go to the section entitled 'Changing Port settings in Windows 95' for further details.

● Click **finish** to complete the installation of your ports.

Windows will then ask you if you wish to restart.
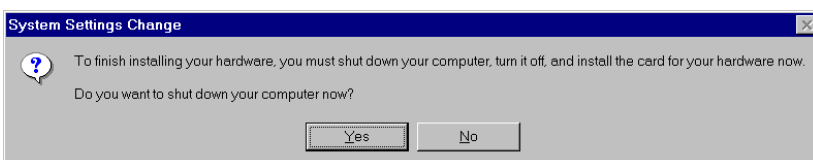
If you have finished the installation procedure, and you are happy with the hardware configuration, then:
● Click **Yes**
Otherwise
● Click **No** and either install any other port or ports by following the procedure above, or reconfigure the port you have just installed by following the procedure shown in the section 'Changing Port Settings in Windows 95'.

Installing as COM5 and Higher

If you are installing the AT Triple RS232 as COM5 and higher, then on the second screen of the Add New Hardware Wizard, you should:
● Click the **No** Option Button. Do Not let Windows Detect your new serial ports. It will take forever and is not guaranteed to detect the correct settings.
● Click **Next**.

● Select the **Ports (Com & LPT)** entry, then click **Next**

● Highlight **Standard Port Manufacturers** in the left window, and **Communications Port** in the right one, Click **Next.**
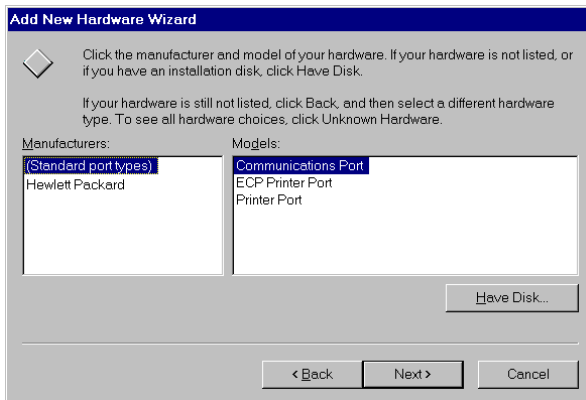
Windows 95 then guesses the settings for your new port. This guess will more than likely not match the setting that your card is physically set to. If this is the case, refer to the next section for changing the settings.

● Click **next**.

Add New Hardware Wizard

Windows can install your hardware, using the following settings.

WARNING: Your hardware may not be set to use the resources listed. If you need to, you can adjust these settings by using the Device Manager in the System control panel before restarting your computer. To change your hardware settings, see the documentation that came with your hardware.

To continue installing the software needed by your hardware, click Next.

Resource type	Setting
Input/Output Range	02E8 - 02EF
Interrupt Request	05

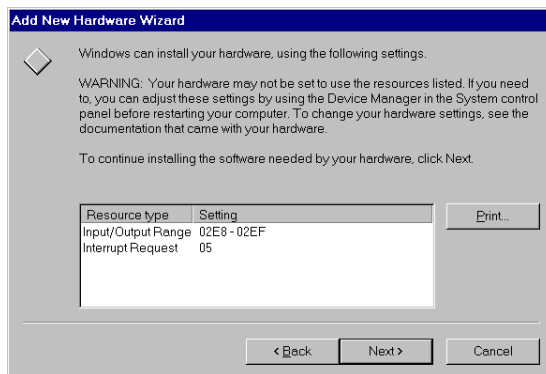Print...

< Back   Next >   Cancel

Windows will now inform you that it has finished installing the new hardware.
● Click **finish** to complete the installation of your port.

Add New Hardware Wizard

Windows has finished installing the software necessary to support your new hardware.

< Back   Finish   Cancel

Windows will then ask you if you wish to restart.

System Settings Change

To finish installing your hardware, you must shut down your computer, turn it off, and install the card for your hardware now.

Do you want to shut down your computer now?

Yes   No
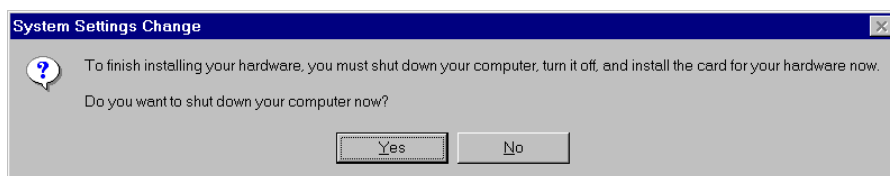
If you have finished the installation procedure, and you are happy

with the hardware configuration, then:

● Click **Yes**

Otherwise

● Click **No** and either install any other port or ports by following the procedure above, or reconfigure the port you have just installed by following the procedure below.

Changing Port Settings in Windows 95

All port settings are changed via the Windows 95 Device manager. This is most easily reached by opening My Computer, Right clicking in white space and selecting properties from the drop-down list.
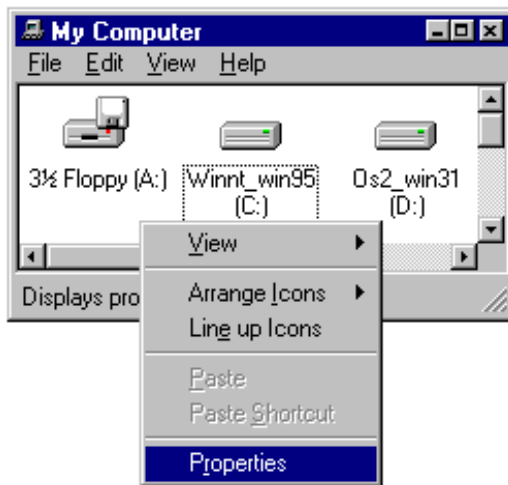
● Select the **Device Manager** Tab, then click the **View devices by type** button, then double click **Ports (COM & LPT).**
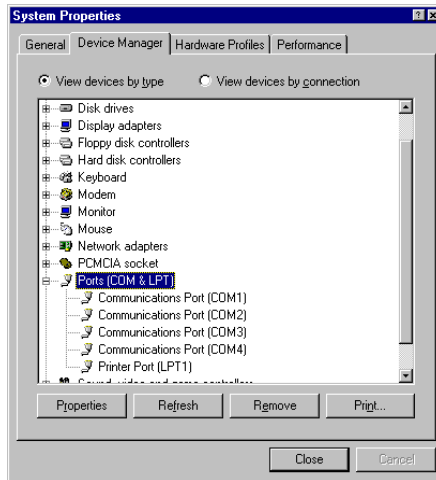
● Double Click the port whose settings you wish to change, then select the **Resources** tab
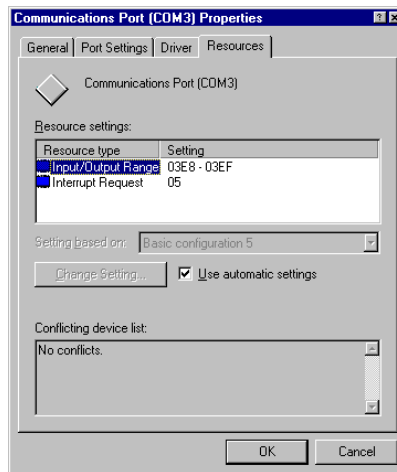
Un-check the Use automatic settings box, and double click on the resource that you wish to modify. In some circumstances, Windows will not let you change these settings. This is easy to work around. Simply click the downward pointing arrow on the side of the box labelled **Setting based on**. Change the

configuration to a higher number until Windows will let you change the resource setting that you want.
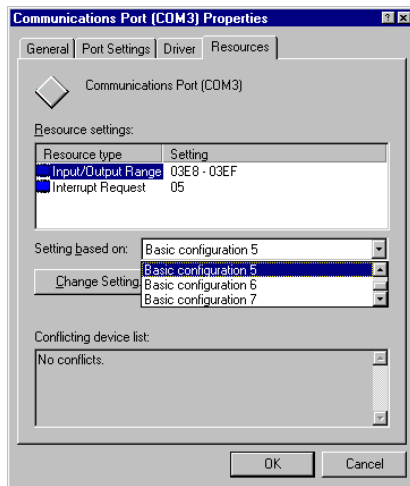
If changing the I/O address setting, the following dialogue box will appear. To change the I/O address setting, either use the scroll bars to select the appropriate value, or simply type in the required range.
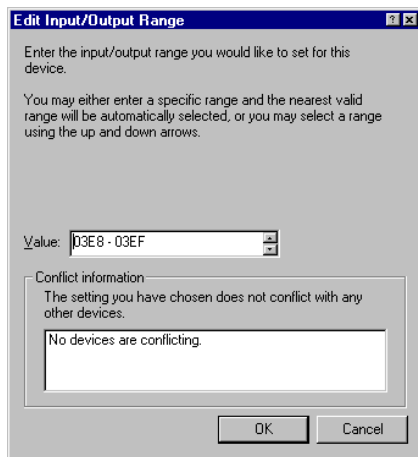
If you are changing the interrupt value, the following dialogue appears:
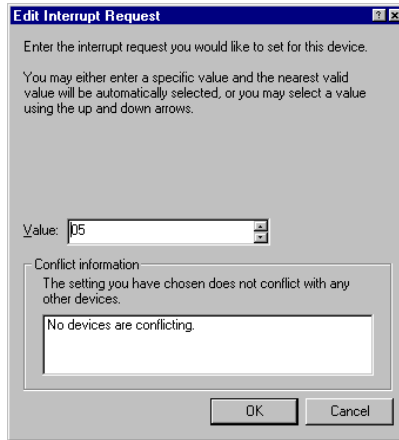
When you have finished changing the settings for that particular port, click **OK**. If you have made changes, Windows will ask if you wish to restart.
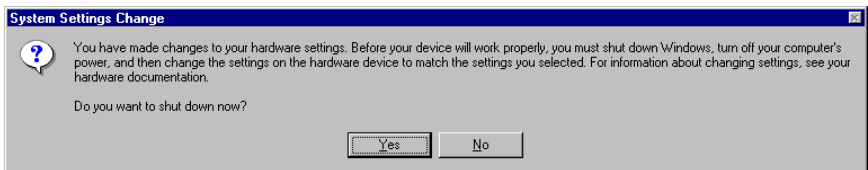
If you have further changes to make to other ports click **No** else click **Yes**.

Microsoft Windows NT

Microsoft Windows NT Provides built in support for 255 standard serial ports. The installation procedure is very similar to that of Windows 3.x. Successful operation of the ports is again dependant on using separate interrupts for each of the ports which you are installing.

Windows NT versions 3.51 and 4.0 may look much different, but the installation procedure for Serial Ports remains the same.

To obtain trouble free mix and match of the COM ports, assuming that only 1 Serial Port already exists on your PC,

Version 3.51: ● From Windows NT's **Program Manager**, select the **main** program group, and double click the **control panel** icon.

Version 4.0: ● Open 'My Computer' and double click the **Control Panel** icon or click the start button, select **Settings** and then **Control Panel**
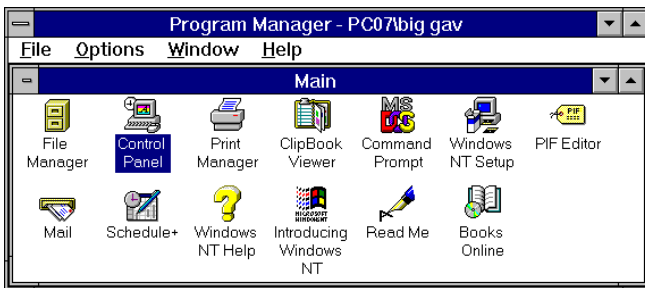
From control panel:
● Double Click the **Ports** icon.
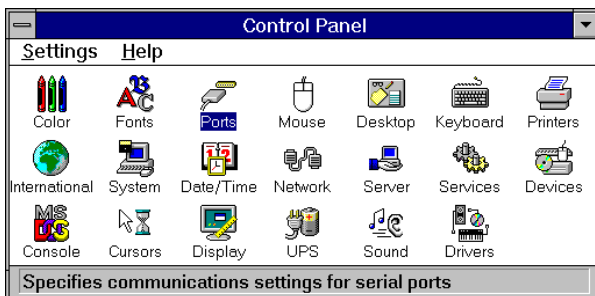
If you only have one existing port in your PC then your ports applet will look something like the following figure.
Your three new ports will need to be added one at a time. Each ports installation procedure is equivalent to the other from this

point.
● Click the **Add** button.
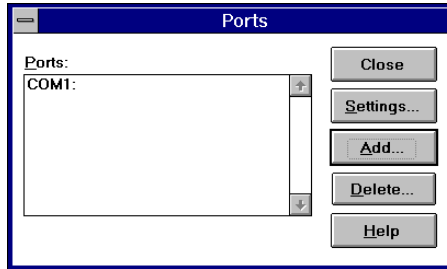
Change any of the settings in the box as appropriate to match your hardware, as Windows NT may not correctly determine the hardware settings. For our example com2 should be set to IRQ3. The FIFO enabled check box should be checked if you wish to enable the hardware buffer. This option is only available on cards which have a 16C550 or better UART chip on board. It is recommended that this option be set.

When you are happy with your settings:
● Click **OK**.
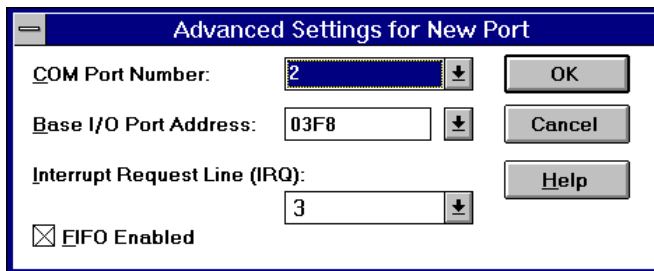
Windows will now tell you that for your settings to take effect, you will need to restart your machine. If you have more ports to install:
● Click **Don't Restart Now**, and continue with the installation of your other ports,

 else

● Click **Restart Now** to complete the installation.

OS/2 Version 2.x and OS/2 Warp.

OS/2 provides built in support for up to four serial ports. OS/2 assumes that COM1 is on IRQ4 and that COM2 is on IRQ3. To use COM3 and COM4 ports on AT bus machines it is necessary to include parameters on the DEVICE=COM.SYS statement in the CONFIG.SYS file. That is:-

DEVICE=COM.SYS (N,XXX,I)
where N is the COM port number (3 or 4)
           XXX is the I/O port address (3E8, 2E8, etc.)
           I is the IRQ (interrupt) level (from 1 to 15)

For COM2 using IRQ3 the CONFIG.SYS file entry should be:-
DEVICE=COM.SYS (2,2F8,3)

For COM3 using IRQ10 the CONFIG.SYS file entry should be:-
DEVICE=COM.SYS (3,3E8,10)

For COM4 using IRQ11 the CONFIG.SYS file entry should be:-
DEVICE=COM.SYS (4,2E8,11)

For COM2 using IRQ 3, COM3 using IRQ10 and COM4 using IRQ11 the CONFIG.SYS file entry should be:-
DEVICE=COM.SYS (2,2F8,3) (3,3E8,10) (4,2E8,11)

Ensure that a 1 space character is between each of the bracket enclosed variables
When using OS/2 note that these settings DO NOT get passed on to the WINOS/2 control panel but also have to be set as for Windows 3.1, see above.
If the standard COM3 or COM4 addresses are not used, and a mouse is on COM1/2, in DOS VDMs the ADDRCOM3/4 programs may report COM port addresses in a different order to that expected. However the MODE command etc. correctly accesses the right port in both DOS VDM and OS/2 sessions.

Please Note:

As with Windows 3.1, OS/2 STRONGLY recommends, and automatically detects, the use of 16550 FIFO chips on serial ports.

Serial Card Installation.

Once the card has been correctly configured then it can be installed in the PC. It is best to make a note of the serial port i/o address selection and IRQ jumper settings for later use.

In general, the card should be configured, then installed and if your BIOS only detects 2 serial ports, run or add to the AUTOEXEC.BAT file, the ADDRCOM3/4 programs, see below.

Finally cables should be attached and communication with the serial peripheral devices should be established.

Provided that the RS232 installation is attacked in this orderly manner, every thing should work first time. If it does not then check the software selectable communications parameters, Baud rate, Parity, stop bits first, and that the communications program is attempting to access the serial port installed. If this fails to solve the problem check the cable connections. Finally check that the card is indeed configured as you believed.!

Informing The PC Of The Port Address.

The BIOS of most PC's automatically detects whether serial ports at COM1 and COM2 addresses are present in a machine. The problem comes with serial ports COM3 and COM4. PC's have a table of information that stores the setup of the PC, this is the BIOS data area. It has space for 4 serial port addresses. The trick is to get the right addresses into the third and fourth port areas.

On the utility disk included with your card there are several programs for setting the COM3 and COM4 addresses. To set COM3 address from the DOS command line enter:-

ADDRCOM3 03E8 <return>

To set COM4 address from the DOS command line enter:-

ADDRCOM4 02E8 <return>

To set the COM port address from within the AUTOEXEC.BAT file use a text editor to add the following lines to the file.

ADDRCOM3 03E8
ADDRCOM4 02E8

DOS 4, 5 and DOS 6 have features allowing access to COM3 and COM4. DOS 2 and DOS 3 only allowed access to COM1 and COM2, though some manufacturers their own DOS versions, e.g. COMPAQ, had enhanced MODE commands etc. to set up and use the extra ports. Please see Chapter 4, Serial Utility Disk for more details.

If you are uncertain how many serial and parallel ports are in the machine, it is wise to run the ADDRCOM3 program BEFORE installing our AT Triple card. Type :-

ADDRCOM3 <return>**.**

The PC responds with something like:-

COM1: 03F8
COM2: 02F8
COM3: <not set>
COM4: <not set>
LPT1: 03BC
LPT2: <not set>
LPT3: <not set>

Provided that the RS232 installation is attacked in this orderly manner, every thing should work first time. If it does not then check the software selectable communications parameters, Baud rate, Parity, stop bits first, and that the communications program is attempting to access the serial port installed. If this fails to solve the problem check the cable connections. Finally check that the card is indeed configured as you believed.

Installing The PC Serial Card In The Computer.

NOTE: Always turn the computer OFF before installing or removing any interface board..!!!

After having made sure that the I/O address and jumpers are correctly set, now is the time to insert the PC Serial card into the i/o connector slots in the computer.

STEP 1: Before the PC card can be installed the power to the PC **MUST** be switched **OFF!**

Figure 2-10. Remove Cover Mounting Screws.
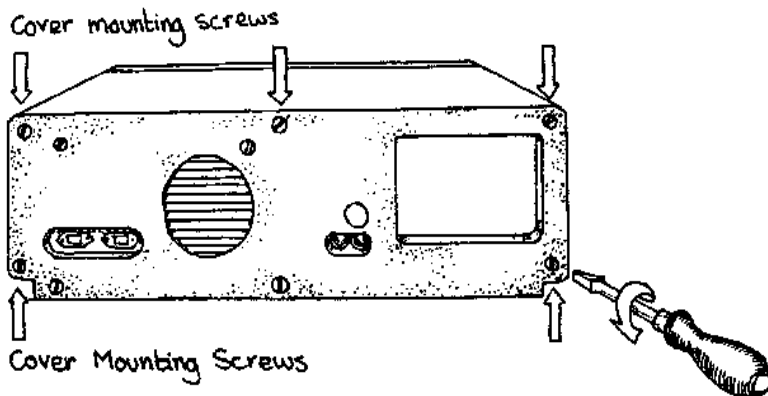
STEP 2: Then using a screw driver, remove the cover mounting screws on the back panel of the PC system unit.

Figure 2-11. Removing The PC Cover.
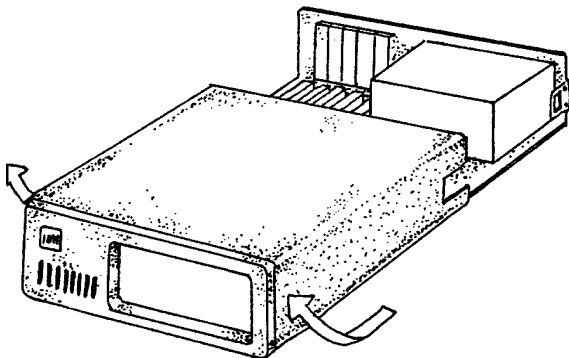
STEP 3: Next, remove the PC's cover by sliding it forward and up. It usually helps to disconnect the keyboard from the PC since it tends to get in the way when the case is removed.

Figure 2-12. Removing Blanking Cover.
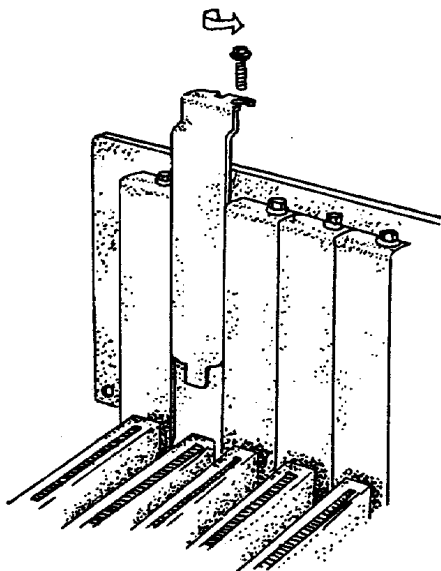
STEP 4: Choose an empty expansion slot. The PC Serial card will fit either a full length or a short slot, a 8 bit PC type slot or a 16 bit AT type slot. In general it is wise to leave the larger slots free for those boards that insist on greater room. Remove the blanking cover protecting the slot on the PC back panel. KEEP the blanking cover screw safely for later.

Figure 2-13. Inserting The PC Serial Card.
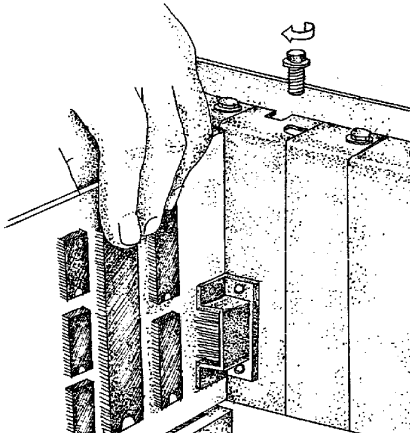
STEP 5: Now insert the PC Serial card in the slot. Be careful to ensure that the gold plated PCB fingers fits neatly into the i/o expansion connector. Press down firmly but evenly on the top of the PC Serial card.

STEP 6: The three 9 way D connectors should fit neatly through the slot's aperture to the outside world. NB. Use the screw kept back from the blanking cover to screw the PC Serial retaining bracket into the PC back panel housing.

STEP 7: Now replace the system units cover by carefully sliding it down and back over the system unit. Replace the cover mounting screws.

STEP 8: After attaching all the monitor and keyboard cables, power up the PC. Do not forget the mains power cable!

The PC should power on in the normal way.

Problems!

If the system fails to power up normally check the following.:

i.)   Ensure that the PC Serial card is installed correctly.

ii.)   Ensure that other cards in the PC have not been upset.

iii.)   Ensure that the power is connected and the PC is switched ON!

iv.)   Ensure that the PC Serial i/o address is set correctly (Port1 default is COM2 02F8h, Port2 default COM3 03E8h, Port3 default COM4 02E8h).

*If all these have been checked and the PC still does not power up then there is probably a conflict of I/O address between the PC Serial card and another board in the PC. Ask your dealer to check this or contact the HELP line as given on Intro. page 2.

CHAPTER 3
SERIAL UTILITY DISK
& PROGRAMMING
GUIDE.

Introduction.

This describes the programs on the Serial Utility Disk supplied with all our serial cards.

The ADDRCOM3/4 programs are usually used in the AUTOEXEC.BAT file to automatically set the BIOS addresses for COM3 and COM4 ports every time the PC is powered on in older PC's that do not automatically detect COM3 and COM4.

Utility Disk Contents.

The utility disk contains the following programs:-

ADDRCOM3.EXE	Set/reads the address of serial port COM3.
ADDRCOM4.EXE	Set/reads the address of serial port COM4.
ADDRLPT2.EXE	Set/reads address of printer port LPT2.
ADDRLPT3.EXE	Set/reads address of printer port LPT3.
COMTEST.EXE	Comprehensive serial port test program. Utilises the LOOP BACK self test mode of the serial port chip.
COMM.BAS	Ultra simple BASIC RS232 port sample program.
COMM.EXE	Exe file of the complied BASIC source code.
RS232.BAS	BASIC program useful in debugging RS232 problems, not interrupt driven.
RS232.EXE	Ready to run, compiled version of BASIC program.
RS422.BAS	BASIC program useful in debugging RS422 problems, not interrupt driven. Demonstrates RS422 handshaking method.

RS422.EXE	Ready to run, compiled version of BASIC program.
RS485.BAS	BASIC program useful in debugging RS485 problems, not interrupt driven. Demonstrates RS485 handshaking method.
RS485.EXE	Ready to run, compiled version of BASIC program.

In addition the 3.5 inch disk contains the following files:-

@69E1.ADF	ADF file for PS DUAL RS232 card.
@69E0.ADF	ADF file for PS DUAL RS422/485 card.

ADDRCOM Program.

The ADDRCOM programs are used to set the serial port address in the BIOS. The ADDRCOM3/4 programs are typically used in the AUTOEXEC.BAT file to automatically set the BIOS addresses for COM3 and COM4 ports every time the PC is powered on.

The BIOS of most PC's automatically detects whether serial ports at COM1 and COM2 addresses are present in a machine. The problem comes with serial ports COM3 and COM4. PC's have a table of information that stores the setup of the PC, this is the BIOS data area. It has space for 4 serial port addresses. The trick is to get the right addresses into the third and fourth port areas hence the ADDRCOM3 and ADDRCOM4 programs.

If no parameters are given then the program prints out the current serial and parallel port assignments. Thus:-

```
ADDRCOM3 <return>
COM1: =3F8
COM2: =2F8
COM3: <Not Set>
COM4: <Not Set>
LPT1: =3BC
LPT2: <Not Set>
```

```
LPT3: <Not Set>
```

The ADDRCOM programs take one parameter, the hexadecimal port address. The programs can be run from the DOS command line or from batch files like AUTOEXEC.BAT.

To set COM3 address from the DOS command line enter:-
ADDRCOM3 03E8 <return>
To set COM4 address from the DOS command line enter:-
ADDRCOM4 02E8 <return>

To set the COM port address from within the AUTOEXEC.BAT file use a text editor to add the following lines to the file.
ADDRCOM3 03E8
ADDRCOM4 02E8

ADDRLPT Program.

The ADDRLPT programs are used to set the parallel port address in the BIOS and work in exactly the same manner as ADDRCOM3.

The COMTEST Program.

The COMTEST program is a comprehensive serial port test program. It utilises the LOOP BACK self test mode of the serial port chip. The program is menu driven and allows the user to enter their own serial port address or choose from a range of predefined ones.
COMTEST tests the scratch register, an excellent way of checking the read write path to the serial port chip.
Older serial port cards using the 8250 chip will fail this test.
COMTEST tests the input and output handshake lines and it also tests the transmitter and receiver at a variety of baud rates from 115,200 baud to 110 baud.

Any serial port that fails any of the tests should be considered suspect. Note an interface card may still fail to perform even if it passes COMTEST since this program does not check the line drivers or external connection of the card.

COMM.BAS - A Simple BASIC Comms Program.

Figure 3-1 is the most simple two way communications program possible in BASIC. This source code is included on the disk as COMM.BAS, the program COMM.EXE is the compiled ready to run version.

Figure 3-1. COMM.BAS Simple BASIC Program.

```
10 A$ = "COM2:2400,N,8,1,CS0,DS0,CD0,RS"
                      REM CHANGE TO SUIT YOUR SETUP!
20 CLS : PRINT "COMMS PARAMS ARE "; A$
30 OPEN A$ FOR RANDOM AS #1
40 OPEN "CON" FOR OUTPUT AS #2
80 :
100 B$ = INKEY$:
110 IF B$ = "" THEN 130: REM IF NO DATA THEN SKIP
130 PRINT #2, B$;
160 IF B$ = CHR$(27) THEN END :REM EXIT IF USER
                                PRESSED ESCAPE
170 PRINT #1, B$;
180
190 IF EOF(1) THEN 100: REM IF ANY DATA TO READ
                        THEN GET IT
200 A$ = INPUT$(LOC(1), #1)
210 PRINT #2, A$;
220 :
300 GOTO 100
```

Note:

Basic, GW-BASIC, Quick Basic and MS DOS version 5
QBasic all expect:-
COM1: to be address 03f8hex using interrupt IRQ4
COM2: to be address 02f8hex using interrupt IRQ3.

Line 10   The serial port parameters are set to COM2, 2400 baud, No parity, 8 data bits 1 stop bit, ignore handshake lines. This may be changed to COM1 or COM2 at any suitable baud rate and parity options.

Line 30   A file is opened to the specified serial port.

Line 40   A file is opened to the screen.

Line 100   Any data entered at the keyboard by the user is placed in B$ and

Line 130   The keyboard input is echoed to the screen.

Line 160   If the user pressed ESCAPE then exit program.

Line 170   The keyboard input is sent to the serial port.

Line 190   Test for any incoming data from the serial port waiting in the buffer.

Line 200   The number of bytes actually in the buffer is placed into A$.

Line 210   Incoming data printed on the screen.

Line 300   GOTO Line 100 to start loop again!

BASIC's Serial Communications Support.

As can be seen from the program in Figure 3-1, before any data can be sent or received using BASIC, a file must be opened. BASIC has a special OPEN command, OPEN "COMn, that is used to set the communication parameters and handshake options used by the serial port. The full range of option is shown in Figure 3-2. Parameters within square brackets [ ] are optional.

Figure 3-2. BASIC Open "COM..

OPEN "COMn:[speed][,parity][,data][,stop][,RS][,CS[n]][,DS[n]] [,CD[n]][LF][PE] AS [#] filenum

COMn   n is 1 or 2 indicating COM1: or COM2:.

speed   The communications speed, BAUD rate , in bits per
               second. Valid Baud rates are 75, 110, 300, 600,
               1200, 1800, 2400, 4800 and 9600.
parity   Valid parity options are:-
               N =None   E =Even   O =Odd
               S =Space   M =Mark
 [PE]   The PE option must be specified if Basic is to
               check parity for each data byte received, the default
               is no parity checking. If it detects a parity error then
               a "Device I/O Error" is reported.
               Framing and overrun errors ALWAYS cause a
               "Device I/O Error".
data   The number of data bits per byte sent. Valid options
               are:- 5, 6, 7 or 8. Note if 8 data bits are specified
               then the None parity option must be used.
stop   Number of stop bits. Valid options are 1, 1.5, or 2.

Handshake lines.

  Serial ports have up to 6 handshake lines, there are 2 OUTPUT
handshake line and 6 INPUT handshake lines.

Output Handshake Lines - DTR and RTS.

  DTR: BASIC always sets DTR out true during the OPEN"COM..
statement.

  RTS: The level of the RTS line will be set true unless it is
suppressed by the RS option in the OPEN"COM statement.

Input Handshake Lines CTS, DSR, DCD and RI.

  CTS: Basic expects CTS to be true during OPEN"COM else the
open fails. Disable CTS checking with the CS0 option.

  DSR: Basic expects DSR to be true during OPEN"COM else
           the open fails. Disable DSR checking with the DS0
           option.

DCD: Normally Basic ignores the DCD line, it can be either
         high or low. Basic can be made to check the state of the
         DCD line by using the CD option. When CD[n] is non
         zero then BASIC will only succeed in opening the
         comn port when DCD is true. Disable DCD checking
         with the CD0 option or by leaving the CD option out of
         the OPEN"COM statement.

RI: Normally Basic ignores the RI, Ring Indicator, line.

   More details of the use of the serial port from BASIC can be
found in the BASIC manual.

RS232.BAS - Demonstrates Using Serial Port Chips.

        The RS232.BAS is a demonstration program showing how
to directly program the serial ports registers. The program is NOT
interrupt driven and thus will loose data if attempts to send more
than 10 or 20 bytes a second are made. RS232.BAS does not use
BASIC's built in serial port handler, i.e. it does not use OPEN
"COM.
        As a teaching aide it helps the user understand how to
control the serial port in the PC.
        As a debugging aide it is so simple that if it cannot establish
communication with an external serial device then either the
external device is dead or the cable is wired wrong. RS232.BAS is
an invaluable debugging aide since it gives the state of the input
handshake lines, allows the user to change the state of the output
handshake lines using the function keys and gives the decimal value
of the last byte received. The 25th line of the screen displays the
RS232 communication options selected by the user.

   The source code is included on the disk as RS232.BAS, the
program RS232.EXE is the compiled, ready to run version.

Line

10-100	Clears the screen and defines some constants.
200-299	Port i/o address selection and set up.
300-399	Baud rate selection and set up.
400-499	Data Bits, Parity and Stop bit selection and set up.
500	Sets up the initial state of the RTS & DTR output handshake lines.
510-699	Enables the function keys.
800-999	Displays the program name and explains the use of the function keys.
1000-2999	Main Program loop.
3000-3999	Subroutine: set the baud rate.
4000-4999	Subroutine: set parity data length stop bits.
5000-5199	Subroutine: defines the bits value of input and output handshake lines.
5200-5299	Subroutine: set RTS out true.
5300-5399	Subroutine: set RTS out false.
5400-5499	Subroutine: set DTR out true.
5500-5599	Subroutine: set DTR out false.
6000-6999	Output data string to serial port. Copy to screen.
7000-7199	Define serial port addresses, baud, parity, stop etc.
7400-7599	Subroutine: menu selection of serial port address.
7600-7799	Subroutine: menu selection of Baud rate.
7800-7999	Subroutine: menu selection of Parity, Stop bits and data word length.
8000-8099	Subroutine: update 25th line of the display.
8100-8199	Subroutine: generate original text for 25th line of the display.

The following description will discuss how the program manages the serial port chip. The details of menu selection are apparent upon inspection of the program listing.

The following steps will be explained:-

i)   Setting the baud rate.
ii)   Setting data word length, parity and stop bits.
iii)   Setting output lines RTS and DTR.
iv)   Reading the state of input lines CTS, DSR and DCD.
v)   Detecting incoming data, RXD.
vi)   Sending outgoing data, TXD.

Serial Port Address.

The address of the serial port is used to access the chip it is placed in the variable S.

Baud Rate Selection.

Baud rate selection is made by writing an 16 bit value, as two 8 bit bytes to the Divisor Latch Registers.
To set the baud rate the correct divisor, RATE, for the chip must be determined. See line 3000-3025. e.g. line 3018 below.

```
3018 IF BAUD = 1200 THEN RATE = 96
```

For standard PC serial cards, with a 1.8432MHz clock input the divisor value RATE is given by

```
RATE= 115200/BAUD
```

To access the baud rate divisor bit 7, DLAB, of serial port register 3, the LCR register, must be set true, then the divisor value RATE is written low byte to register 0, high byte to register 1. Finally DLAB, bit 7 of register 3 is set false, allowing normal use of registers 0 and 1.

```
3040 S1 = INP(S + 3): REM" remember parity etc.
3044 :
3050 REM" set baud rate
3052 RATELO = (RATE AND 255): REM set baud lab
3054 RATEHI = INT(RATE / 256): REM set baud lab
3058 :
```

```
3060 REM" set baud rate port
3062 OUT S + 3, S1 OR 128: REM "access dlab
3064 OUT S, RATELO: REM set baud lab
3066 OUT S + 1, RATEHI: REM set baud msb
3068 OUT S + 3, S1: REM restore
3069 :
3081 OUT S + 3, 3
```

Data Word Length, Parity and Stop Bit Selection.

The Data Word Length, Parity and Stop Bit Selection is made by writing an 8 bit byte to the Line Control Register, register 4.

```
4000 REM Set selected Parity Data Length Stop Bits
4020 OUT S + 3, PDS(P):
4030 RETURN
4040 :
4050 :
```

Setting The RTS and DTR Output Handshake Lines.

Setting the RTS and DTR output lines is done by writing an 8 bit byte to the Modem Control Register, register 4.
  DTR is controlled by bit 0
  RTS is controlled by bit 1

Note: OUT2 must be set in all PC and PS/2 design serial cards if interrupt driven i/o is to be performed. This is in addition to setting the IER, Interrupt Enable Register.
In setting the new RTS or DTR state, the current state of the bits NOT being changed, particularly OUT2, must be preserved and restored.

```
5027 DTR = 1: RTS = 2
5030 HSON = DTR + RTS: REM DTR AND RTS TRUE WHEN
                                  READY FOR DATA
5200 REM TURN RTS HANDSHAKE ON
5205 HSON = (INP(S + 4) AND 255 - RTS): REM Mask
                                         OFF RTS
5210 OUT (S + 4), (HSON OR RTS): PRINT "RTS OUT
     TRUE"
5220 GOSUB 8000: REM update bottom line
```

```
5230 RETURN
5240 :
5250 :
5300 REM TURN RTS HANDSHAKE OFF
5305 HSON = (INP(S + 4) AND 255 - RTS): REM Mask
                                        OFF RTS
5310 OUT (S + 4), HSON: PRINT "RTS OUT FALSE"
5320 GOSUB 8000: REM update bottom line
5330 RETURN
5340 :
5350 :
```

Reading The CTS, DSR & DCD Input Handshake Lines.

The state of the CTS, DSR and DCD input handshake lines is detected by reading the Modem Status Register, register 6.

  CTS current state is in bit 4, change in CTS state sets bit 0.
  DSR current state is in bit 5, change in DSR state sets bit 1.
  RI   current state is in bit 6, change in RI state sets bit 2.
  DCD current state is in bit 7, change in DCD state sets bit 3.

  Note that on the Dual RS422/485 cards DCD and DSR are permanently tied TRUE and RI is permanently tied FALSE.

```
5063 REM DCD is normally IGNORED in serial comms
     programs
5064 REM DSR is normally used to check that a
     device is present i.e. is on line.
5066 :
5067 CTS = 16: DSR = 32: DCD = 128
8040 CTS$= "1": CTSI = (INP(S + 6) AND 16): IF
     CTSI= 0 THEN CTS$ = "0"
8042 DSR$= "1": DSRI = (INP(S + 6) AND 32): IF
     DSRI= 0 THEN DSR$ = "0"
8044 DCD$= "1": DCDI = (INP(S + 6) AND 128): IF
     DCDI= 0 THEN DCD$ = "0"
```

Detecting Incoming Data.

Incoming data is received by reading the Receiver Buffer Register, RBR, register 0, a read only register.

Whenever a byte is received, the serial port chip sets the DR bit, Data Ready, bit 0 of LSR, indicating the RBR register is full.

```
  2000 IF ((INP(S + 5) AND 1) = 1) THEN PRINT
CHR$(INP(S));
```

Bits 1, 2 and 3 of the LSR should also be checked as they indicate possible errors in the received data.

Bit 1, OE, Overrun Error, This means that a second data byte has been received before a previously received byte has been read from the RBR register, thus the first byte has been lost. Should have used an interrupt driven program!

Bit 2, PE, Parity Error

Bit 3, FE, Framing Error, These bits mean that either a noisy line has caused the incoming data byte to be corrupted or that the external serial device transmitting the data is using a different communications protocol, wrong baud rate, stop bits, parity or data word length.

  The sample program does not check for any of these errors.

Sending Outgoing Data.

Outgoing data is sent to serial devices by writing each data byte to the write only Transmitter Holding Register, THR, register 0.

Before each byte is sent the THRE bit, bit 5 of LSR, must be set indicating that the transmitter holding register is empty and thus can accept the next data byte for transmission.

```
  6050 IF ((INP(S + 5) AND 32) <> 32) THEN 6050:
```

If any handshake lines must be true before transmission is allowed then the state of the input lines must first be checked by reading register 6, MSR, the Modem Status Register.

```
  6036 IF ((INP(S + 6) AND HSIN) <> HSIN) THEN 6035:
```

In the sample program the THRE, transmitter holding register empty check is performed AFTER each data byte has been sent, it could be checked before sending the next byte.

Since the sample program is not interrupt driven the transmit routine has several lines to constantly poll the DR, Data Ready, bit of the LSR, that indicates whether an incoming byte has been received and is in the RBR.

```
5070 HSIN = 0: REM NO Input Handshake see line 6036

6000 REM SEND A$ TO THE SERIAL PORT IF HSIN IS TRUE
6010 A = LEN(A$): REM NUMBER OF BYTES TO SEND
6012 IF ((INP(S + 5) AND 1) = 1) THEN PRINT
     CHR$(INP(S));
6015 :
6020 FOR X = 1 TO A
6030 OP = ASC(MID$(A$, X, 1))
6035 IF ((INP(S + 5) AND 1) = 1) THEN PRINT
     CHR$(INP(S));
6036 IF ((INP(S + 6) AND HSIN) <> HSIN) THEN 6035:
                         REM Test Handshake In line
6037 REM And Wait Till Other Device is Ready To
     Receive
6040 OUT S, OP:          REM SEND DATA TO TXD REG
6050 IF ((INP(S + 5) AND 32) <> 32) THEN 6050:
     REM WAIT TILL HOLD REG EMPTY
6055                     REM BEFORE SENDING NEXT BYTE
6056 :
6057 REM READ ANY INPUT DATA HERE AS WELL JUST TO
     PREVENT OVERFLOW ERRORS
6058 IF ((INP(S + 5) AND 1) = 1) THEN PRINT
     CHR$(INP(S));
6060 NEXT
6070 :
6100 PRINT A$; : REM COPY TO SCREEN
6200 GOSUB 8000
6400 RETURN
```

PC Serial Port Chips.

The original IBM PC had a serial port based upon the Intel 8250 Asynchronous Communications Element. The 8250 occupies 7 i/o locations in the PC, all serial port communications parameters are programmed by setting bits in the 8250 registers.

The introduction of the IBM AT saw the use of an enhanced 8250 called the NS16450 ACE. This chip has faster access time to allow the 4.77Mhz PC bus be upgraded to the 8 MHz AT bus and has an 8th register, the scratch register. This extra register does NOT change how the ACE works but simply gives the PC programmer one extra location for storing data temporarily. Apart from the scratch register the 16450 behaves, from a programming point of view, identically to the 8250.

Figure 3-3. Comparison Of PC Serial Port Chips.

Chip	Read-Write Cycle Time	No. Regs	Used In	Notes
8250	755ns	7	Original IBM PC	
16450	360ns	8	IBM AT, 286, 386 & 486	Scratch Reg.
16550	280ns	8	IBM PS/2 Models 50-90	FIFO
TL16C452	175ns	8	AT Dual RS232 Card AT Dual RS422/485 Card	2 Serial + 1 printer
			AT Lynx 8 Port RS232	8 Serial
			PS Dual RS232 Card	2 Serial
			PS Dual RS422/485 Card	2 Serial
TL16C552	280ns	8	FIFO versions of all above cards	FIFO
All our range of cards can be fitted with the 16C552 as an option giving 16 bytes of input buffer & 16 bytes of output buffer per port				

In April 1987, IBM introduced the PS/2 range which featured the Micro Channel Architecture, PS/2 computers use an upgraded 16450, the 16550. The reason for using the 16550 was

its faster access time, the 16550 has a two major new features, two 16 byte FIFOs, first in first out buffers, and the possibility of DMA access. However, the IBM Technical References states that the FIFOs should not be used as this 'may result in non detectable data errors.' The way the 16550 is wired up precludes the DMA features being used. The 16550, on power up, behaves identically to the 16450.

The Texas Instruments TL16C452 is an enhanced 16450, featuring high speed twin 16450 serial ports and a complete Centronics Printer Interface on one chip, the TL16C452 is now used in all our PC serial port cards. The equivalent buffered part, 16C552 can be fitted on request.

Figure 3-4. 8250 & 16450 Register Map.

Reg Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
(DLAB=0) RECEIVER BUFFER READ ONLY	RXD DATA BIT7	RXD DATA BIT6	RXD DATA BIT5	RXD DATA BIT4	RXD DATA BIT3	RXD DATA BIT2	RXD DATA BIT1	RXD DATA BIT0
(DLAB=0) TRANSMIT HOLD REG WRITE ONLY	TXD DATA BIT7	TXD DATA BIT6	TXD DATA BIT5	TXD DATA BIT4	TXD DATA BIT3	TXD DATA BIT2	TXD DATA BIT1	TXD DATA BIT0
(DLAB=0) INTERRUPT ENABLE REG	0	0	0	0	MODEM STATE	RCVR LINE STATE	TXD REG EMPTY	RXD DATA FULL
INTERRUPT IDENT REG READ ONLY	0	0	0	0	0	INT ID BIT 1	INT ID BIT 0	0 IF INT TRUE
LINE CONTROL REGISTER	DLAB	SET BREAK	STICK PARITY SELECT	EVEN PARITY SELECT	PARITY ENABLE	NO. STOP BITS	WORD LEN.BIT1	WORD LEN. BIT 0
MODEM CONTROL	0	0	0	LOOP	OUT2	OUT1	RTS	DTR
LINE STATUS REGISTER	0	TXD EMPTY	TXD HOLD EMPTY	BREAK INT	FRAME ERROR	PARITY ERROR	OVERRUN ERROR	RXD DATA READY
MODEM STATUS REGISTER	DCD	RI	DSR	CTS	DELTA DCD	DELTA RI	DELTA DSR	DELTA CTS
SCRATCH REGISTER	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
(DLAB=1) DIVISOR LATCH LSB	DATA BIT7	DATA BIT6	DATA BIT5	DATA BIT4	DATA BIT3	DATA BIT2	DATA BIT1	DATA BIT0
(DLAB=1) DIVISOR LATCH MSB	DATA BIT8	DATA BIT9	DATA BIT10	DATA BIT11	DATA BIT12	DATA BIT13	DATA BIT14	DATA BIT15

NOTE : OUT2 must be true (=1) to enable the interrupt circuitry in all PCs and PS/2s.

CHAPTER 4
RS232 PORT CABLING.

Introduction.

This chapter gives details of the 9 pin RS232 pin outs, cabling and connections, with information on how to connect the serial ports of two PCs and make a selftest loop back connector.

The RS232 Standard.

The RS232 standard is ancient in computer industry terms. Introduced in 1962, it is now widely established. RS232 is a slow speed, short distance, single ended transmission system (i.e. only one wire per signal). Typical RS232 maximum cable length is 50 feet with a maximum data rate of 20K bits per second.

Figure 4-1. RS232 Point To Point Connection.
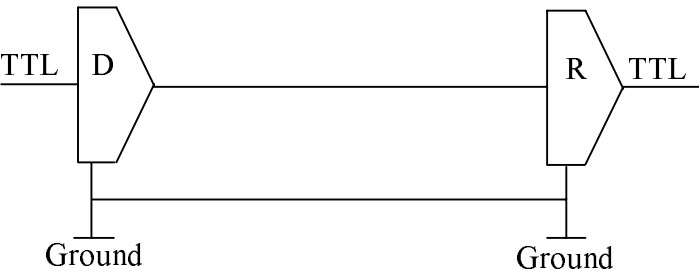
RS232C Standard 1 Driver 1 Receiver	
Line Length	Max Data Rate
50 Feet = 15m	20 Kbits/sec

Pin D Serial Port Pin Outs.

The pinouts of the 9 pin Male D connector is given below.
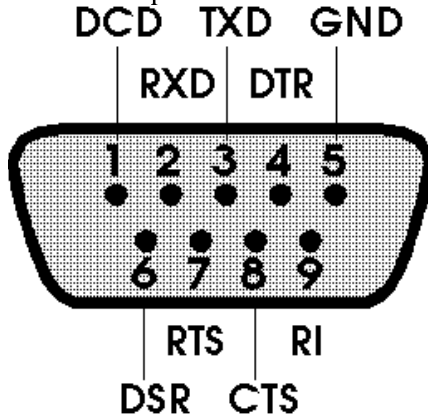
Figure 4-2. 9 Pin D Serial Port RS232 Pin Outs.

PIN	SIGNAL	DIRECTION
1	DCD DATA CARRIER DETECT	INPUT TO PC
2	RXD RECEIVED DATA	INPUT TO PC
3	TXD TRANSMITTED DATA	OUTPUT FROM PC
4	DTR DATA TERMINAL READY	OUTPUT FROM PC
5	GROUND	N/A
6	DSR DATA SET READY	INPUT TO PC
7	RTS REQUEST TO SEND	OUTPUT FROM PC
8	CTS CLEAR TO SEND	INPUT TO PC
9	RI RING INDICATOR	INPUT TO PC.

Pin D Serial Port RS232 Cables.

To connect to the AT style RS232 Serial Port you will need a cable terminating in a 9 way female D connector. It is sound practice to use cables with screws fitted that will allow you to fasten the cable securely to the PC card.

In general, you will need to make up a "cross over" cable to correctly interface the PC to the RS232 port of another computer

or device. Traditionally, making up the cross over cable has been considered a black art. However, provided you have the pin outs and handshake requirements of both sides of your RS232 connection, the cross over cable becomes a matter of common sense. The cross over cable is simply to ensure that the right signals going out of one RS232 port go into the appropriate lines of the other RS232 port.
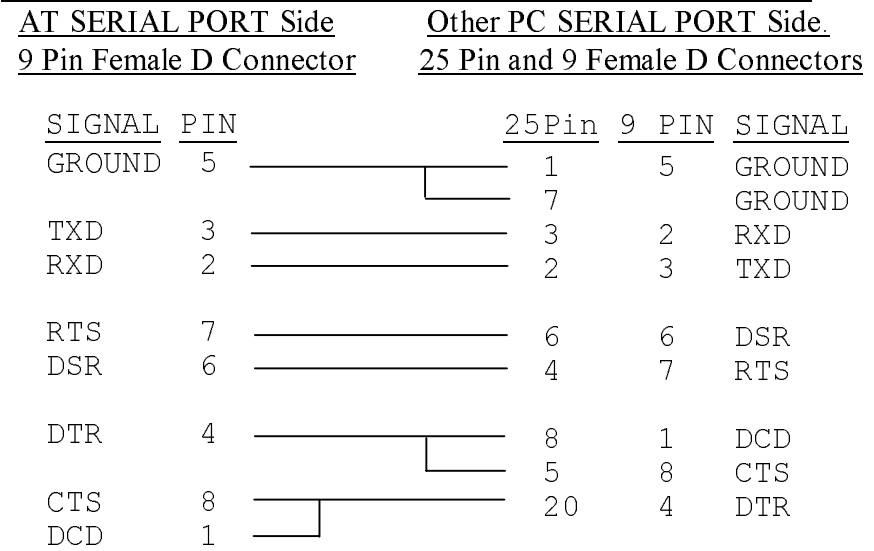
Pin D Serial Port Connection To Another PC.

Suppose we want to connect the AT style 9 pin D Serial Port to the serial port of another IBM PC. See Figure 4-2.

1) Connect up the earth lines.
Line 5 of Serial Port 2 to lines 1 & 7 of the other PC.
This gives the two devices a common earth level.

2) Connect the Transmit and Receive lines together.
Line 3, TXD, Port 2 goes to line 3, RXD, of the other PC.
Line 2, RXD, Port 2 goes to line 2, TXD, of the other PC.
This allows each to receive the data transmitted by the other.

3) Connect up the Port 2 DTR line, pin 4 to the other PC DCD, pin 8 and CTS, pin 5, lines.
Also, connect up the other PC DTR line, pin 20 to the Port 2 DCD, pin 1 and CTS, pin 8, lines.
This allows the receiving device to signal when it can no longer accept data. The receiving device sets DTR false when it is unable to receive any more data. The sending device reads DTR on its CTS and DCD pins. It should stop sending when CTS goes false.

4) Connect up the Port 2 RTS line, pin 7, to the other PC DSR line, pin 6.
Also, connect up the other PC RTS line, pin 4, to the Port

2 DSR line, pin 6.
This RTS line is used to let the other device know that it is ready for data exchange.

## Figure 4-3. 9 Pin D Serial Port To Other PC Cable.

AT SERIAL PORT Side   Other PC SERIAL PORT Side.
9 Pin Female D Connector   25 Pin and 9 Female D Connectors

```
SIGNAL  PIN                    25Pin  9 PIN  SIGNAL
GROUND   5  ─────────────┐       1      5    GROUND
                         └────    7           GROUND
TXD      3  ──────────────────    3      2    RXD
RXD      2  ──────────────────    2      3    TXD

RTS      7  ──────────────────    6      6    DSR
DSR      6  ──────────────────    4      7    RTS

DTR      4  ─────────────┐        8      1    DCD
                         └────    5      8    CTS
CTS      8  ──────┐            20      4    DTR
DCD      1  ──────┘
```

9 Pin D Serial Port To A Modem.

If you are connecting a MODEM to a 9 pin D Serial Port then you will NOT need a cross over cable and a straight through cable connected as the 9 to 25 pin adapter given in Figure 4-3.

Pin D Serial Port Loop Back Connector.

A loop back connector can be used to echo RS232 data transmitted by a serial port back into its own RS232 receiver. In this way the function of the serial port can be tested.

For an AT style Serial Port use the a female 9 way connector wired as in Figure 4-4

Figure 4-4. 9 Pin D Serial Loop Back Connector.

```
     PIN   SIGNAL                          DIRECTION

       3     TxD TRANSMITTED DATA      OUTPUT FROM PC
       2     RxD RECEIVED DATA         INPUT TO PC

       7     RTS REQUEST TO SEND       OUTPUT FROM PC
       8     CTS CLEAR TO SEND         INPUT TO PC


       6     DSR DATA SET READY        INPUT TO PC
       1     DCD DATA CARRIER DETECT   INPUT TO PC
       4     DTR DATA TERMINAL READY   OUTPUT FROM PC
```

Figure 4-5. 9 To 25 Way Adapter.

  This adapter cable makes the AT style 9 pin serial port, look like
the standard PC 25 pin serial port. It is NOT a cross over cable!

9 Pin AT SERIAL PORT   25 Pin PC SERIAL PORT
9 Pin Female D Connector   25 Pin Male D Connector

```
   SIGNAL    PIN                       PIN    SIGNAL
   GROUND     5  ─────────────────      1     GROUND
                                        7     GROUND

   DCD        1  ─────────────────      8     DCD
   RXD        2  ─────────────────      3     RXD
   TXD        3  ─────────────────      2     TXD
   DTR        4  ─────────────────     20     DTR
   DSR        6  ─────────────────      6     DSR
   RTS        7  ─────────────────      4     RTS
   CTS        8  ─────────────────      5     CTS

   RI         9  ─────────────────     22     RI
```

A small adapter with this connection is available from us or your
dealer, at a nominal charge.

Index

Index