BRAINBOXES

422/485 AUTOGATING Series

Including

Velocity 422/485 AUTOGATING

2.1 EDITION August 1998

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.

Due to Brain Boxes commitment to quality, software is subject to continuous improvements: upgrades can be obtained from www.brainboxes.com/software.

Serial Solutions are designed, manufactured, and supported by

BRAIN BOXES.

Unit 3C, Wavertree Boulevard South,
Wavertree Technology Park,
Liverpool, L7 9PF,
England.
Telephone: 0151-220 2500
Fax: 0151-252 0446
E-mail: help@brainboxes.com / sales@brainboxes.com
Web: www.brainboxes.com

ACKNOWLEDGEMENTS.

PCP TECHNOLOGY

422/485 AUTOGATING Series

Including

Velocity 422/485 AUTOGATING

2.1 EDITION August 1998

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.

Due to Brain Boxes commitment to quality, software is subject to continuous improvements: upgrades can be obtained from www.brainboxes.com/software.

Serial Solutions is supplied to you by

PCP Technology Ltd.

PCP House,
Albany Park,
Frimley Road,
Camberley,
Surrey, GU15 2PL.
Telephone: 01276-686688
Fax: 01276-686602

ACKNOWLEDGEMENTS.

.

HARDWARE MANUAL

422/485 AUTOGATING Series

Including

Velocity 422/485 AUTOGATING

2.1 EDITION August 1998

Guarantee.

FULL 36 MONTHS GUARANTEE.

We 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 your Dealer, with proof of purchase, who will determine whether to repair or replace this product with an equivalent unit.

COPYRIGHT.

Due to the Manufacturers commitment to quality, software is subject to continuous improvements: information regarding upgrades can be obtained from your supplier.

Serial Solutions is supplied to you by

ACKNOWLEDGEMENTS.

AT RS422/485 MANUAL

Outline Contents

THE LAYOUT OF THIS MANUAL

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

Chapter 2 - At Dual Port RS422/485 And AT Velocity RS422/485 Hardware Configuration shows you how to configure the settings of the Opto, AT Dual and Velocity RS422/485 cards, after a brief discussion of the RS232, RS422 and RS485 standard, serial port IRQ and address selection, RS485 gating, Autogating, multiplexing and sample wiring diagrams are given.

Chapter 3 - How to install your card into the PC, After configuration a step by step guide on how to open your PC and insert the card.

Chapter 4 - At Dual Port RS422/485 And AT Velocity RS422/485 Software Configuration, details on how to install your Opto, AT Dual and Velocity RS422/485 in DOS, Windows 3.x, 95, 98 and NT as well as OS/2.x and OS/2 Warp.

Chapter 5 - Serial Utility Disk and Programming Guide, describes the sample programs on the disk supplied with the card. The chapter concludes with a detailed discussion of how to program the 16450 serial port chip.

The index - covers the complete contents of this manual.

CONTENTS

CHAPTER 1

SERIAL SOLUTION SOFTWARE.

Introduction

This chapter is a brief description of 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
Serial Solutions for Windows NT

All the Serial Solutions drivers have the following features:-

Drivers for PC standard UARTs e.g. 8250 and 16450
Drivers for PC FIFO UARTs e.g. 16550 as well as the new improved 32 byte 16650 and 64 byte 16750 UARTs
Support for shared interrupt cards e.g. our Quad 232, Lynx 232 and Quad 422 cards.
Support for any mix of RS232, RS422 and RS485 handshake schemes.
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, supporting 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 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.

Terminal Emulation Program.

Two sets of terminal programs are included. 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.

The 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.

Each of the programs behave in exactly the same way. Once running the PC acts as a terminal; any characters typed at the keyboard are sent to the serial port, and any characters received from the serial port are displayed on the screen. Function keys are used to display the Help screen, use the set up menus or exit the program.

In addition the terminal programs recognise an impressive set of commands, that may be executed automatically or interactively.

The commands include features to enable Echoing of transmitted and received data, set up the serial port parameters, set substitution characters and string for both in coming and outgoing data, execute terminal batch files, transmit from and receive to disk, execute DOS commands from within the terminal program.

The input output translations offer the user a way of smoothing over the differences between different pieces of equipment, allowing conversion of data between otherwise incompatible systems.

Additional Sample 8 Port Software

As well as the EASY programs, there is MANYPORT.EXE and its source code MANYPORT.BAS. The .exe version has been compiled using Microsoft Quick Basic V4.5. Its purpose is to demonstrate how to program multiport cards from BASIC. It will work with Lynx 8 and Quad 4 port cards, DigiCHANNEL cards, Flytech Flynix-8 FAT-011 cards, as well with any mix of 2 port especially if each port has its own individual IRQ.

Termdef.txt

Termdef.txt is a data file of commands that the terminal emulation programs use to define the properties of the particular emulated terminal. By building up his own library of terminal definitions a user can quickly emulate either his favourite terminal or literally scores of different terminals on the one PC.

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. It 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 out 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.
EasyC32	C source code, project files and ready to run.exe program for an easy to understand Windows 95 terminal program. Learn how to write Windows comms apps correctly the easy way. Also written to work in windows NT.

Serial Solutions For Windows NT

Windows NT has an improved communication API and directly supports up to 255 ports. No extra driver is necessary for Windows NT to drive multiport cards. Serial Solutions for Windows NT consists of the following programs:-

Multiport.cpl	Control panel applet which allows the simple configuration of all ports on a multiport card simultaneously.
Mulport.hlp	Help file for the above control panel applet.

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 attractively laid out manual in an A5 binder with slip case, containing 100 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.

Cabling Requirements.

Try our popular RS232 D type cables, and RS422 & RS485 twisted pair cables! High quality and competitively priced.

CHAPTER 2

AT 1 PORT, DUAL PORT AND VELOCITY RS422/485 HARDWARE CONFIGURATION

Introduction

This chapter explains how to configure and install the Serial Solution AT 1 Port, Dual Port RS422/485, Opto Isolated AT Dual Port RS422/485 and AT Velocity RS422/485 card in an IBM PC computer. Detailed instructions are given how to set the address select & IRQ jumpers. RS485 users may need to alter the 485 Autogating and Multiplex jumpers.

The half size RS422/RS485 cards will fit into long or short slots, in standard 8 bit PC slots or in the longer 16 bit AT slots. The card works in PC, XT, AT, 386, 486 and Pentium PC compatibles.

AT Dual, Velocity and Opto RS422/485 Card Features.

The AT 1 Port, Dual Port and Velocity RS422/485 cards have the following features: -

(One or)Two ports independently selectable as RS422 or RS485.
Reliable communications up to 4000 feet, 1.2 Kilometres.
100% IBM Compatible serial port Texas Instruments 16C550. The Velocity 422/485 is fitted with a 16C750 UART, an enhanced but compatible chip to the 16C550.
Jumper selectable serial port address, COM1 -COM8.
TXD, RXD, RTS and CTS signals.
Two or four wire operation.
RS485 Gating jumper options:
TXD driver Gating: Always On, RTS enable or Autogated .
RXD receiver Gating: Always On, RTS disable or Autogated .

RS485 TXD/RXD multiplex selectable by jumpers: - either FULL DUPLEX or HALF-DUPLEX.

On board 120 Ohm terminating resistors.

On board Fail Safe open circuit and short circuit detection.

Fully double buffered for reliable asynchronous operation.

High-speed circuitry to ensure operation with fast PC's e.g. 450Mhz Pentium II machines WITHOUT use of extra wait states.

Jumper selectable interrupt level. IRQ 2-7, 10-12 & 15.

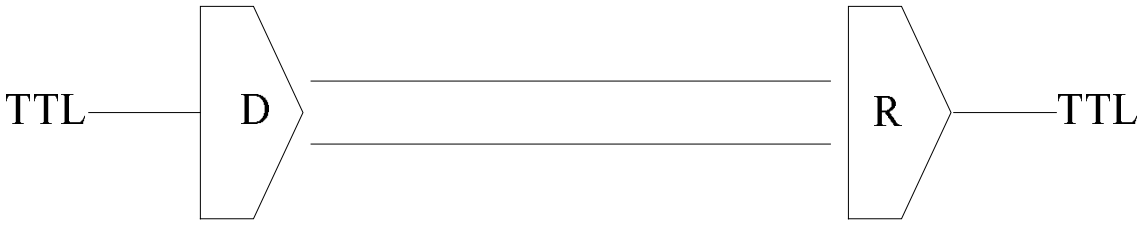
Serial Port Features: -

Baud Rate:	50 Baud to 115,200 Baud. 50 Baud to 921,600 Baud (Velocity RS422/485 only)
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. 14.7456MHz (Velocity RS422/485 only)
I/O Connection:	Dual 9 pin D Male type connectors.
Drivers:	SN75174
High Level Voltage:	3.7V typical at 33mA source
Low Level Voltage:	1.1V typical at 33mA source
High Level Current:	-60mA max
Low Level Current:	60mA max
Receivers:	SN75175
Difftial I/P threshold:	200mV max.
Hysteresis:	50mVolt typical
Input Impedance:	>12K Ohm without terminators.

The RS422 Standard.

The RS422 standard defines a serial communications standard. RS422 is a high speed and/or long distance data transmission. Each signal is carried by a pair of wires and is thus a differential data transmission system. Over distances up to 40 feet the maximum data rate is 10 Megabits per second, and for distances up to 4000 feet the maximum data rate is 100 Kilobytes per second. A 120-Ohm resistor should be used to terminate the receiving end of the line. It is generally used between one transmitter receiver pair to ONLY one other transmitter receiver pair, but each output can drive up to 10 receivers.

RS422 Standard 1 Driver up to 10 Receivers	
Line Length	Max Data Rate
40 Feet = 12m	10 Mbits/sec
400 Feet = 122m	1 Mbits/sec
4000 Feet = 1219m	100 Kbits/sec

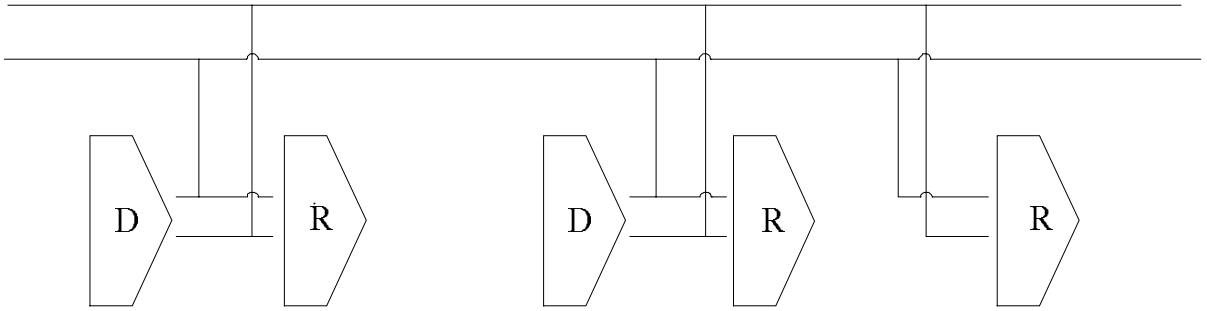


The RS485 Standard.

The RS485 standard is similar to the RS422 standard upon which it is based. The main difference is that up to 32 transmitter receiver pairs may be present on the line at one time. A 120-Ohm resistor should be used to terminate either end of the main line. If more than one device may transmit data, the RTS line is used as

transmit enable signal, so preventing contention between talkers.

RS485 Standard Up to 32 Driver/Receiver Pairs	
Line Length	Max Data Rate
40 Feet = 12m	10 Mbits/sec
400 Feet = 122m	1 Mbits/sec
4000 Feet = 1219m	100 Kbits/sec



Terminating Impedance's

RS422 and RS485 lines should be terminated at the end of the main branch of the RECEIVER, in the cables characteristic impedance. These terminating impedance's stop echoes caused by the serial data being reflected back at the cable ends. It is not necessary to terminate the transmitter end of the twisted pair.

The AT Dual Port RS422/485, Opto Isolated AT Dual Port RS422/485 and AT Velocity RS422/485 cards have the correct 120 Ohm terminating resistors for the RXD twisted pair line and the CTS twisted pair line fitted for both the serial ports on the card.

There is no need to add any more at the PC end.

The terminating impedance's shown later in the wiring diagrams of Figure 2-8, Figure 2-9, Figure 2-11 & Figure 2-13 are automatically provided by the on board resistors and do not have to be added by the user.

Fail Safe Open Circuit Detection.

Open circuit is when there are no drivers on the circuit. This

occurs by design in party line multi driver/receiver systems and unintentionally when the twisted pair line is accidentally cut or disconnected or the transmitting device fails. In RS485 party line systems there are extended periods of time when none of the many possible talkers are gated onto the bus. This is known as the line idle state and occurs when all the driver outputs are in the high impedance state. The lines float, perhaps being pulled to the high or low state by noise or other voltages on the line.

Without fail safe open circuit detection false start bits are detected by the receivers, either corrupting good communications or causing noise to masquerade as good data.

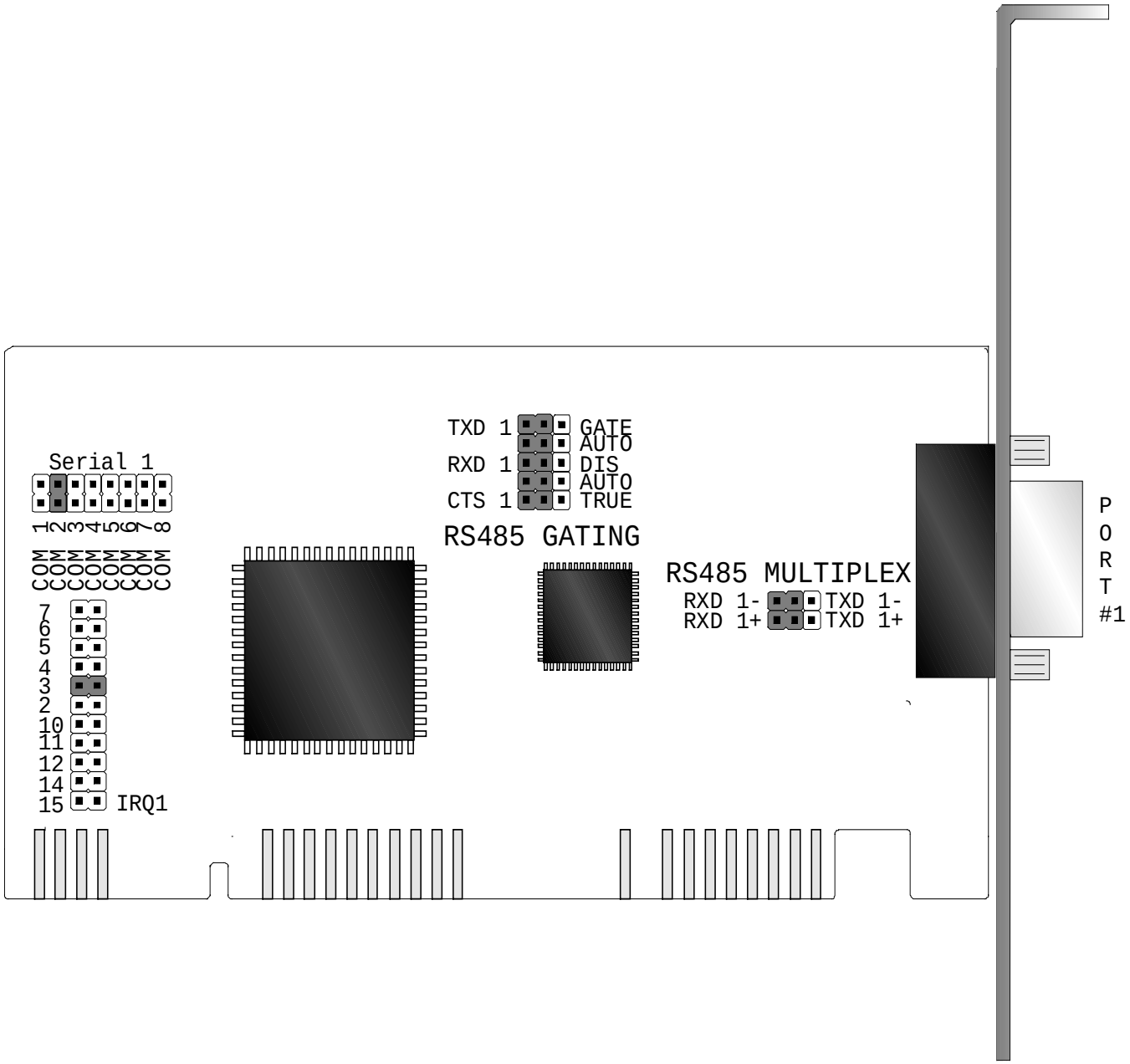
The on board fail safe open circuit detection causes the receiver to go to a known, pre-determined state and prevents false start bits and bad data being detected during open circuits.

Fail Safe Short Circuit Protection.

Short circuits are when the two lines of a twisted pair are connected together. This occurs due to either accidental damage to the cable or due to failure of one or more transmitter/receivers on the line. The short circuit condition is dangerous since damage to the receiver may occur and communication may be corrupted or prevented.

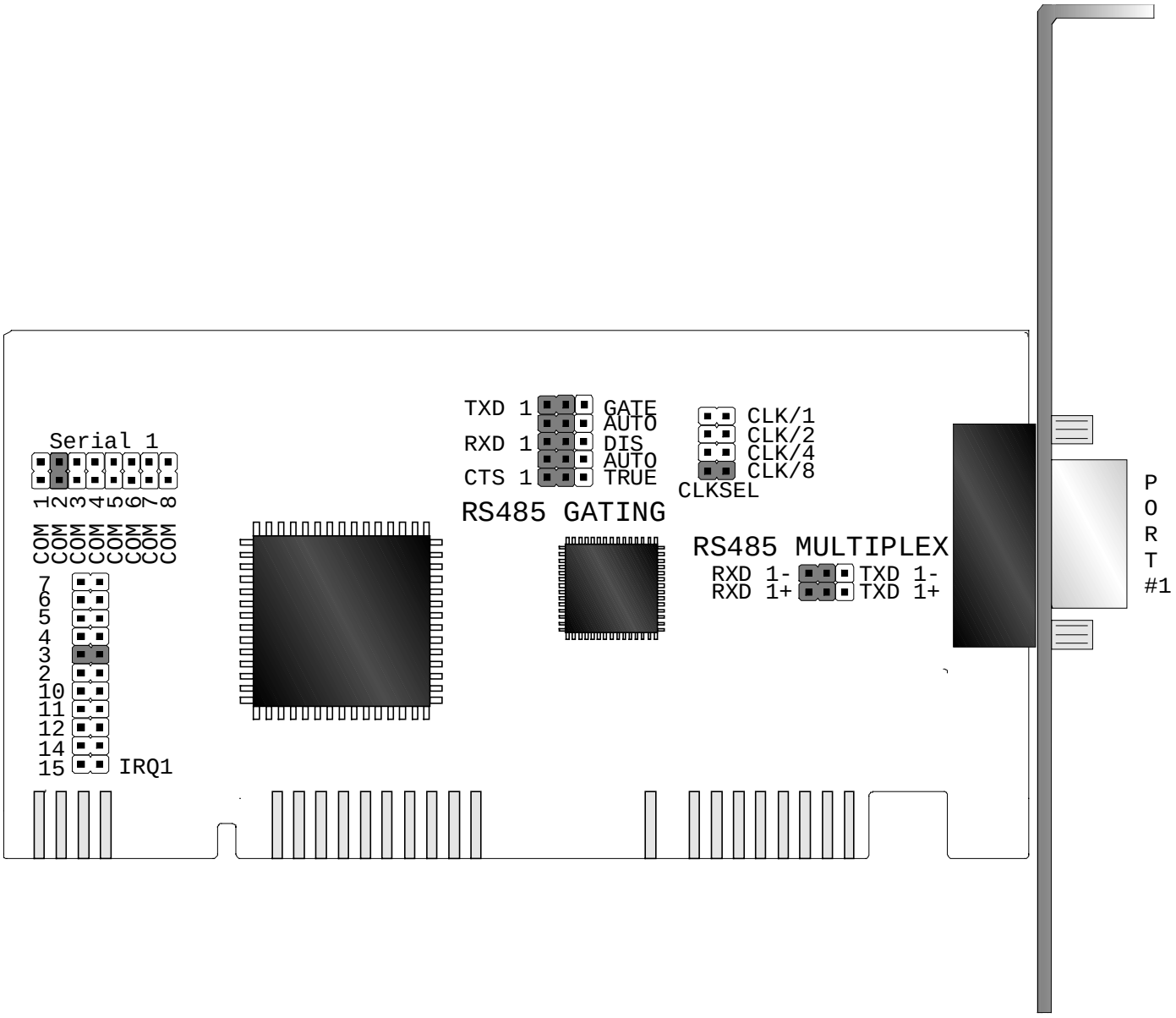
The on board fail safe short circuit detection prevents the line impedance from going to zero and thus protects the inputs of receivers and the outputs of drivers.

Figure 2-1a). AT 1 Port RS422/485 Card Layout.



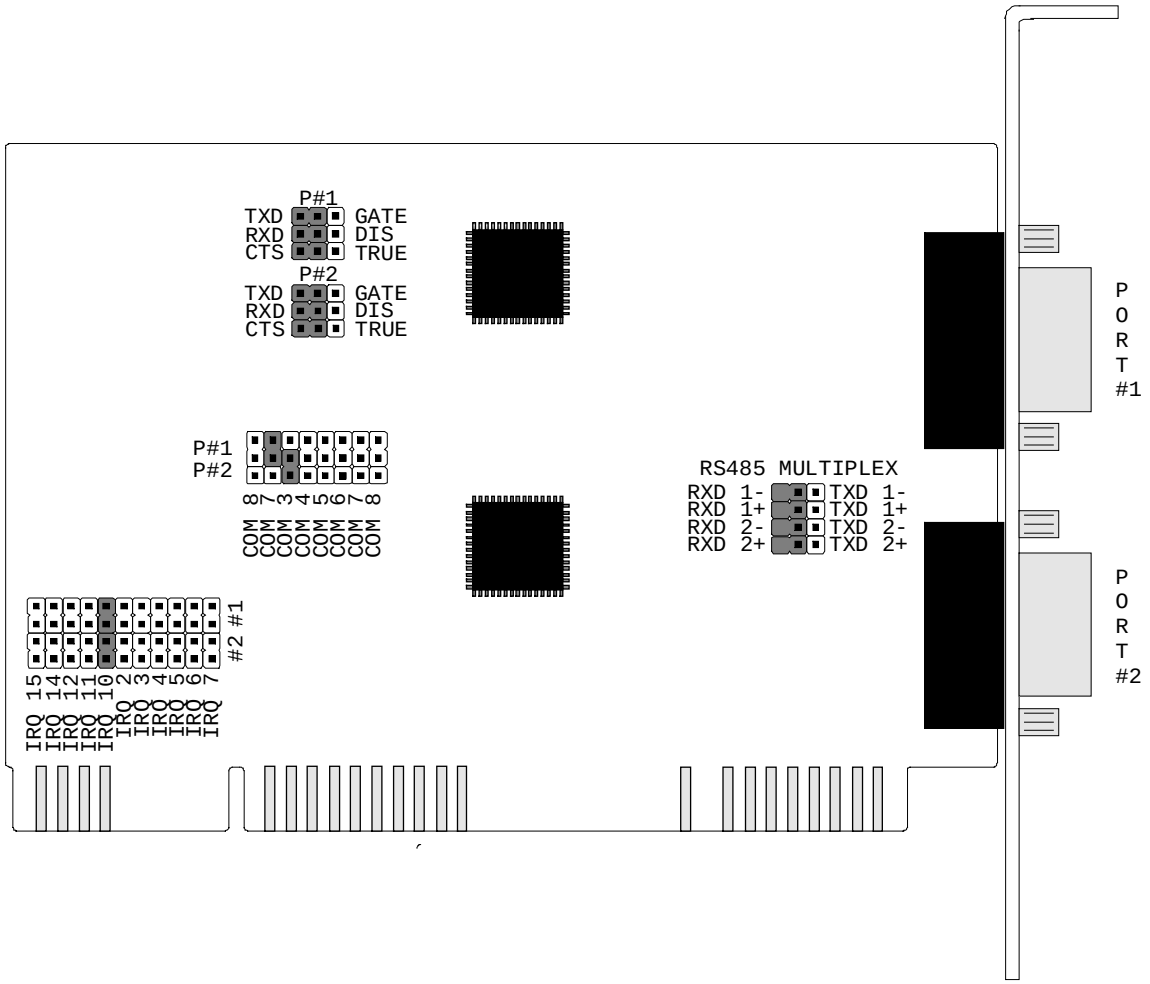
AT One Port RS422/485 Dimensions: 13.5 x 7.0 cm
5.31 x 2.75 in

Figure 2-1b). AT 1 Port Velocity RS422/485 Card Layout.



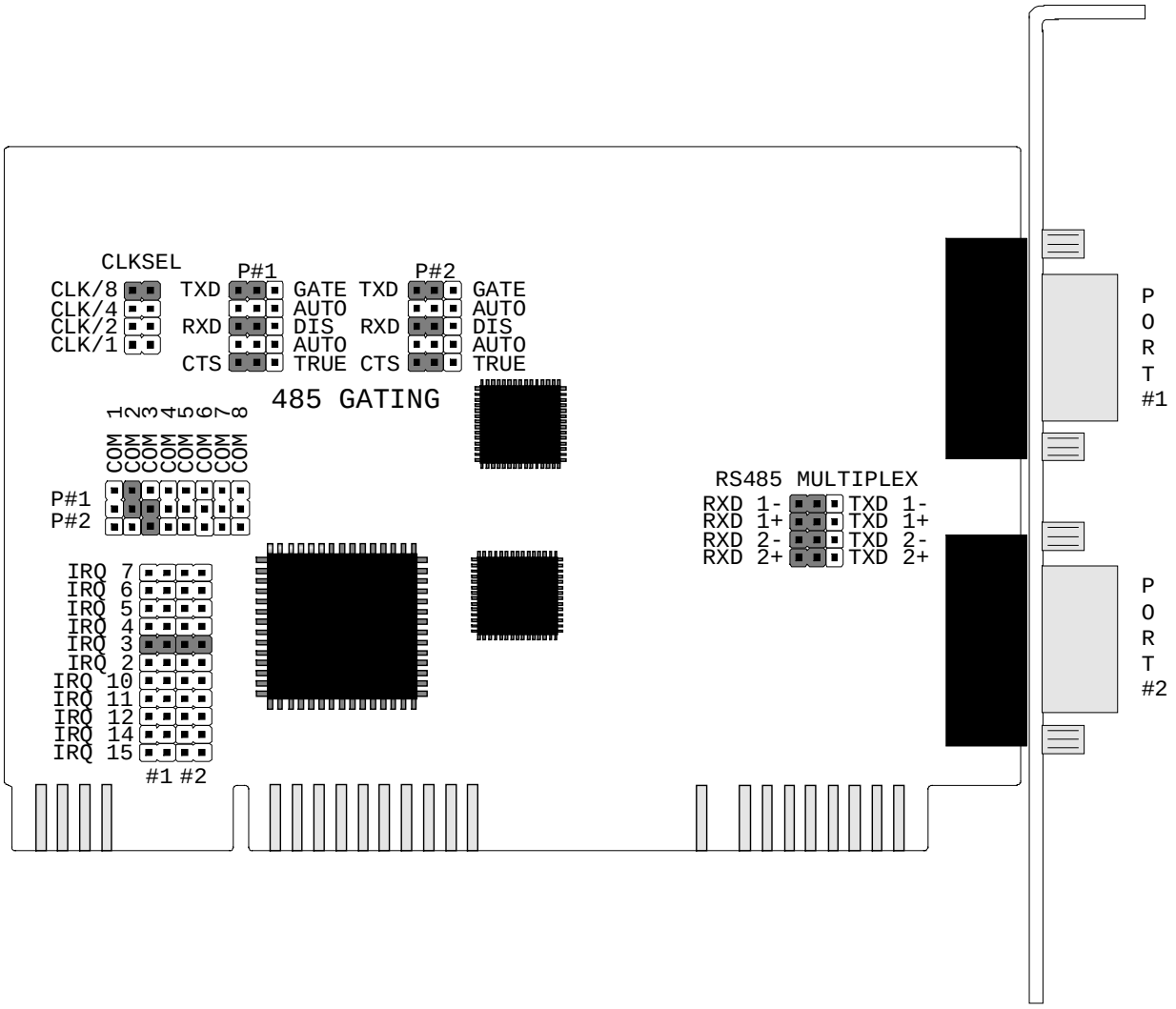
AT One Port Velocity RS422/485 Dimensions: 13.5 x 7.0 cm
5.31 x 2.75 in

Figure 2-1c). AT Dual Port RS422/485 Card Layout.



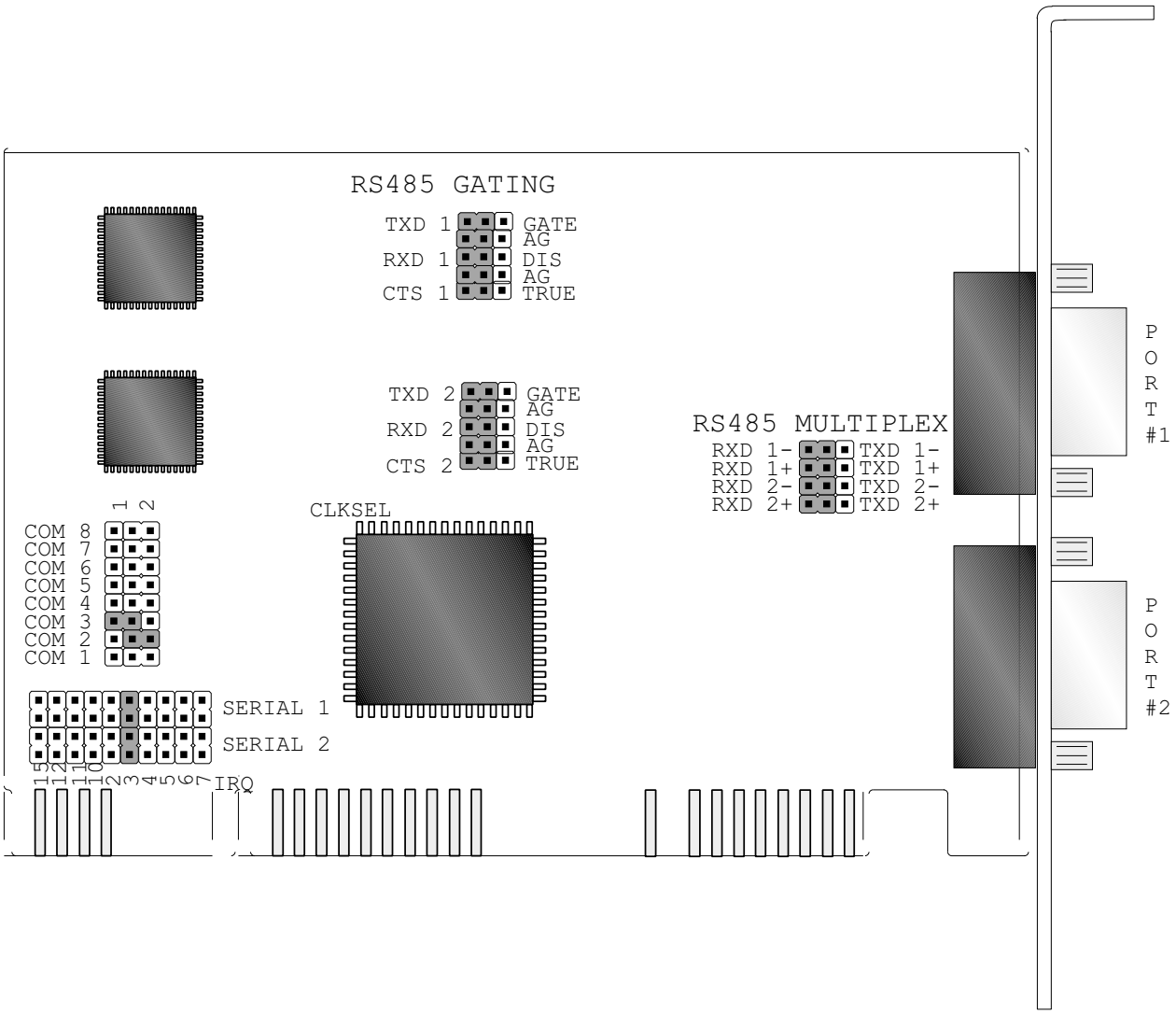
AT Dual Port RS422/485 Dimensions: 13.0 x 8.5 cm
5.40 x 4.20 in

Figure 2.1d) - AT Velocity RS422/485 Card Layout.



AT Velocity RS422/485 Dimensions: 13.5 x 8.5 cm
5.2 x 3.2 in

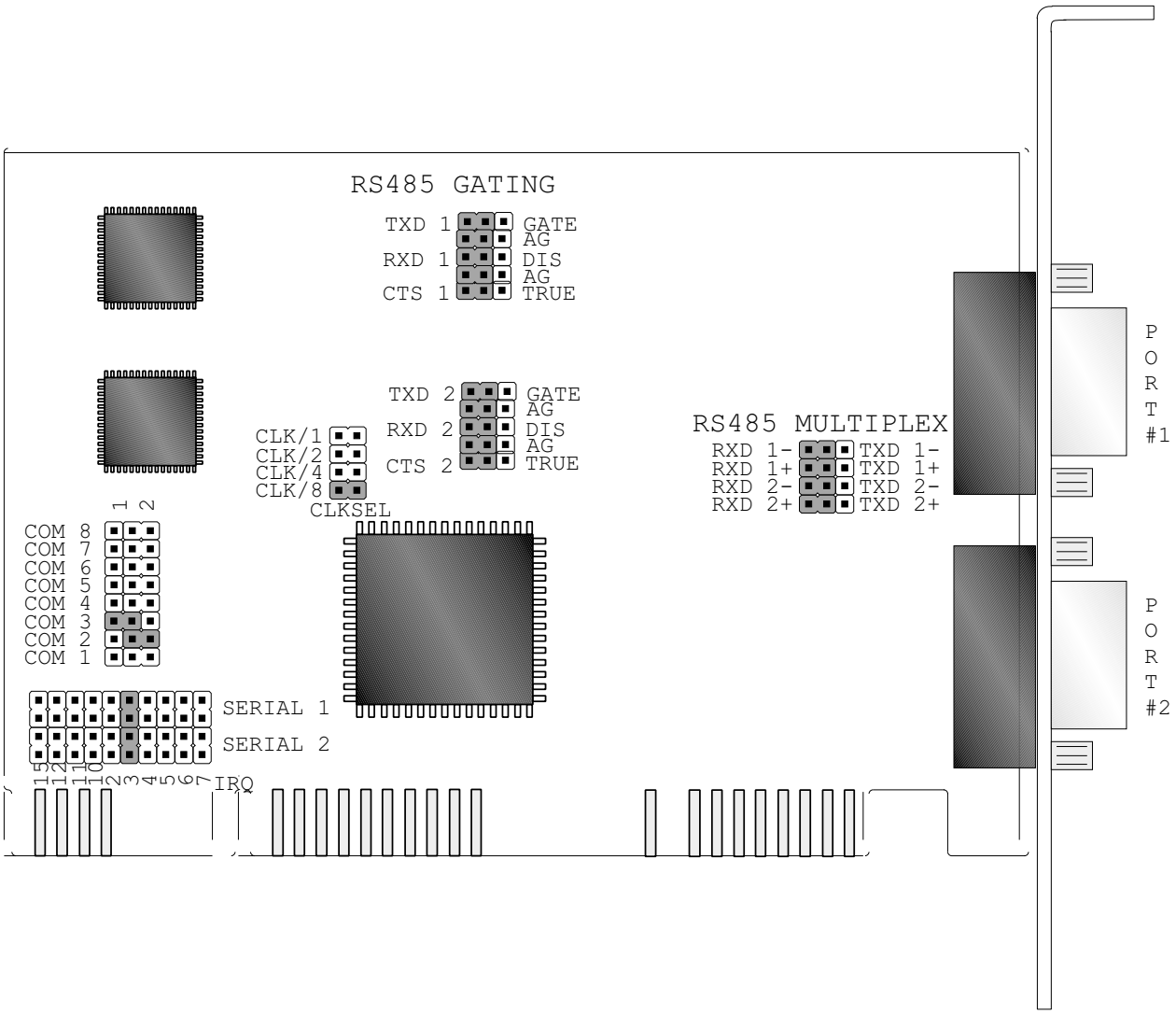
Figure 2-1e). Opto Dual Port RS422/485 Card Layout



Opto Isolated RS422/485 Dimensions: 10.8 x 15.7 cm
4.2 x 6.2 in

Opto RS422/485 Optical Isolation:- +/- 1500 Volts DC
1000 Volts AC

Figure 2-1e). Opto Velocity RS422/485 Card Layout



Opto Velocity RS422/485 Dimensions: 10.8 x 15.7 cm
4.2 x 6.2 in

Opto Velocity Optical Isolation:- +/- 1500 Volts DC
1000 Volts AC

Configuring the RS422 Card.

In the state it leaves our factory, the AT 1 Port, 1 Port Velocity, Dual Port, Dual Port Velocity, Opto Isolated Dual Port and Opto Isolated Velocity RS422/485 cards are ready to plug straight into an IBM compatible PC computer. All cards have been configured for the RS422 operation.

Therefore, unless you have GOOD REASON, you do not need to alter its default setting.

However, due to the presence of other serial port cards in the PC, your SERIAL SOLUTIONS card may need configuring to suit your set-up. The layout of the cards is given in Figures 2-1a), 2-1b) 2-1c), 2-1 d), 2-1 e) and 2-1 f) above.

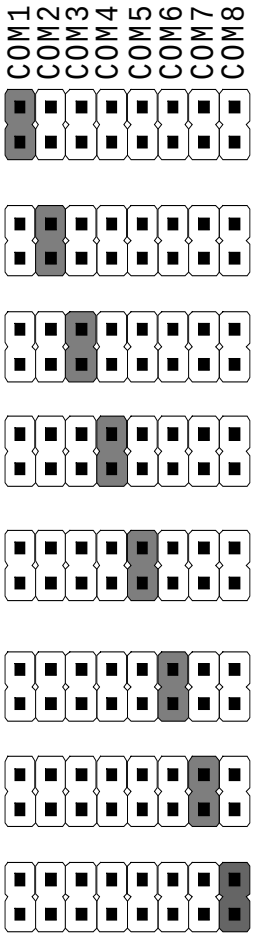
Serial Port Jumpers.

Serial port 1 drives the UPPER 9 pin D connector and serial port 2 drives the LOWER 9 pin D connector. There are 4 sets of jumpers associated with the operation of the two ports; also on the Velocity RS422/485 cards there is an extra jumper block, the "clock select", which is associated with port speed. Their locations on each card are show in Table 2.1, below:

Table 2-1. Jumper and Header Locations

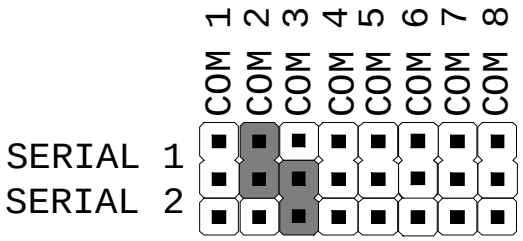
	AT 1 Port RS422/485	**AT Dual RS422/485**	**Opto Iso RS422/4855**
Com Port Select	Top left	Middle left	Middle left
IRQ Select	Bottom left	Bottom left	Bottom Left
RS485 Gating	Top centre	Left of centre	Top centre & centre
RS485 Multiplexing	Middle right	Middle right	Middle right
Clock Select (Velocity Models)	Top right	Top left	Left of centre

Figure 2-2. Port Address Jumper Block.



Serial Port Address Selection

The movable jumper on the address jumper block, (shown below with default settings), is used to specify the I/O address of the first serial port. This determines whether the card is configured as COM1, COM2, COM3, COM4 or COM5- COM8.



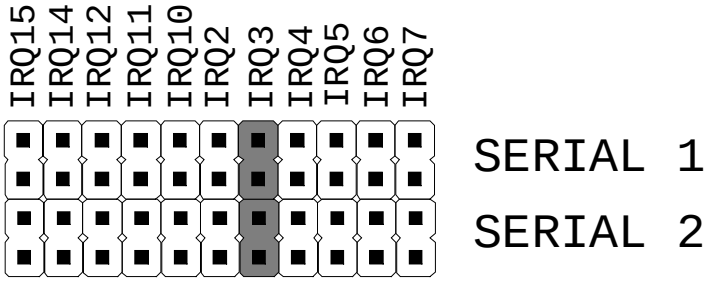
NOTE: On the Opto Dual AT RS422/485 the order of the COM numbers is reversed - COM8 is on the left hand side and COM1 is on the right hand side.

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

The COM5 to COM8 addresses, 01A0 hex to 01B8 hex will not work on the genuine original 8088 IBM PC but are valid for most 8088 PC's and all machines of Pentium Pro standard and below. Below is shown a representation of the IRQ Port jumper block on the AT Dual RS422/48, with factory defaults.

Serial Port IRQ Interrupt Jumper Selection

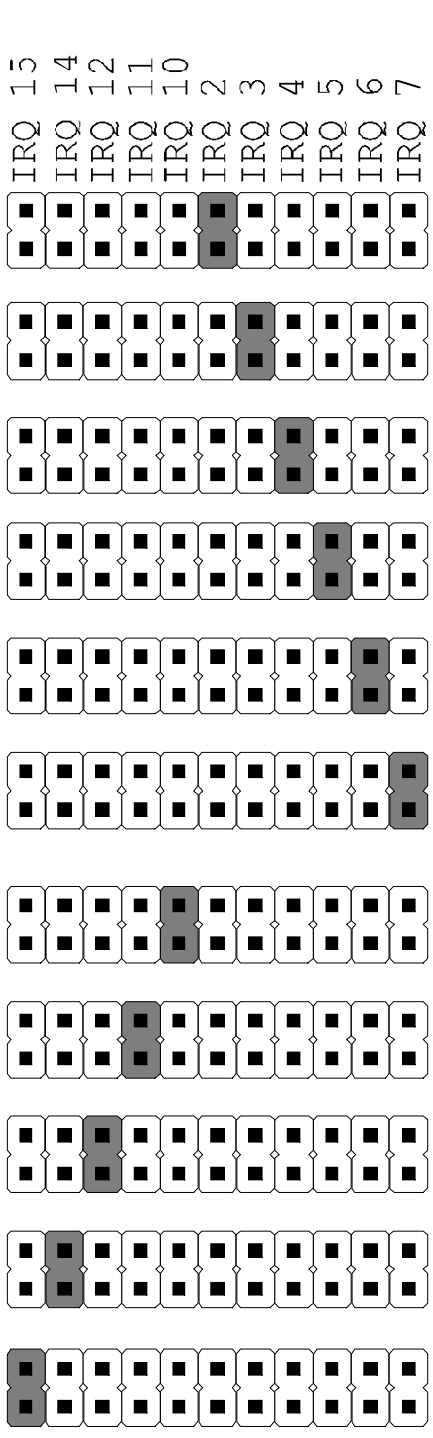
By referring to Figures 2-1 locate, the Serial Port IRQ interrupt jumper block (pictured below with default settings) at the lower left of the board. The movable jumper on the jumper block is used to specify which hardware interrupt is to be generated by the PC serial port.



Two rows of jumpers are visible, the top row corresponding to serial #1 and the bottom row to serial #2. By placing a jumper over two pins in the same row, an interrupt vector may be assigned to a serial port.

Most users will require that the IRQ be set to correspond to serial port address selection. However, 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.

Figure 2-3. COM Port IRQ Jumper Block.



NOTE:

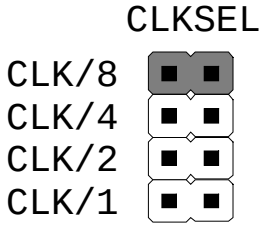
IRQ 0 & 8 - Timer & Clock Interrupts, not on expansion bus
IRQ 1 - Keyboard interrupt, not on expansion bus
IRQ 9 - Best left Unexplained, not on expansion bus
IRQ 13 - Maths coprocessor interrupt, not on expansion bus

Velocity RS422/485 specific Features

Serial Port Clock Speed Jumper Selection

This jumper determines the frequency of the reference clock that is supplied to the serial port and thus controls the maximum available data rate from that port.

Figure 2.4 Serial Port Clock Speed select.



The factory shipped default setting of CLK/8 provides a maximum baud rate of 115.2 kbaud, thus providing standard performance out of the box. At this setting the Velocity RS422/485 card will work in systems designed around the standard RS422/485 cards, however, when set to values higher than this, it is unlikely to work.

The following table gives the maximum baud rates available when the CLK jumper is set to its other positions:

CLK	Baud Rate
CLK/1	921,600 Baud
CLK/2	460,800 Baud
CLK/4	230,400 Baud
CLK/8	115,200 Baud

Unless your software has been written specifically for the velocity RS422/485 card, it is unlikely that the application will report the actual baud rate of the card, if the CLK jumper is not in the /8 position. The baud rate reported by software is actually only a logical assignment to a divisor value. This divisor is programmed into the serial port to divide the input frequency of the reference clock to

achieve a calculated baud rate, this obviously assumes a fixed reference clock. The table below lists a few division values for standard baud rates:

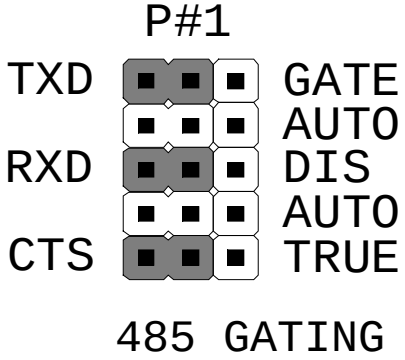
Baud	Divisor
115.2 k	1
57.6 k	2
38.4 k	3
28.8 k	4

Thus, when a baudrate of 28,800 is set from software the actual baudrate is the maximum baud rate (baudmax) / 4. Consequently, if the CLK jumper is at CLK / 1 then setting a baudrate of 28,800 in software actually leads to a real baudrate of baudmax / 4, which is equal to 921,600 / 4, which is 230,400.

Another, quick way of calculating the real baud rate is to multiply the baud rate you have selected by (8/CLK). For example if CLK =2 and baud rate selected is 115,200, actual baud rate is 115,200 x (8 / 2), which is 460,800.

Serial Port 485 Gating Jumper Block.

By referring to figure 2-1, locate the Serial Port Gating jumper block at the left of the upper centre of the board. This block (pictured below with default settings) differs slightly on the AT OPTO DUAL RS422/485, it does not have the AUTOGATING settings, but other than that, functions in the same way. N.B. The worked example given in figure 2-5 below, is for Port 1 gating, but also applies to Port 2.



The 5 movable jumpers are (see Figure 2-5, below): -

i.) TXD GATE / TXD AUTO GATING - For RS485 Use.

For use in RS485 Half Duplex systems only, where only one twisted pair connects both receive and transmit lines of all serial devices together. The RTS line is used as a control to allow the outgoing TXD data to reach the 9-pin connector. When RTS is false, the TXD lines are tristated, the PC cannot send data but other devices may send data, which the PC receives. When the RTS line is true, the PC can send data down the TXD line to be received by other devices. Note the PC will hear its own transmitted data echoed back on the RXD lines unless the RXD DISABLE jumper, below, is set.

If the base address of serial port 1 is BASE, and since RTS is bit, decimal value 2 then: -

```
OUT BASE+4,3+8  REM RTS True TXD Enabled
OUT BASE+4,1+8  REM RTS False TXD Disabled
```

Note: The 1+8 keep DTR and OUT2 true.

If the AUTO (**Autogating**) jumper is set the card itself will automatically enable / disable the transmitter, thus removing the responsibility from the software and guaranteeing a rapid “Receive turnaround” especially in multitasking operating systems. N.B. In this setting the card will autogate regardless of whether or not the users software is toggling the RTS line.

ii) RXD DISABLE / RXD AUTO GATING - For RS485 Use.

This jumper is the compliment of the above RXD GATE jumper. For use in RS485 Half Duplex systems only, where only one twisted pair connects both receive and transmit lines of all serial devices together. This jumper is used to stop the PC hearing its own transmitted data. When the RTS line is true, no received data, RXD, can be received. When the RTS line is false all data sent by the other serial devices is heard.

If the base address of the serial port 1 is BASE, and since RTS is bit1 decimal value 2 then:-

```
OUT BASE+4,3+8   REM RTS True RXD Disabled
OUT BASE+4,1+8   REM RTS False RXD Enabled
```

Note the 1+8 keep DTR and OUT2 true.

If the AUTO (**Autogating**) jumper is set the card will automatically disable / enable the receiver, thus removing the responsibility from the software and guaranteeing a rapid “Receive turnaround” especially in multitasking operating systems.

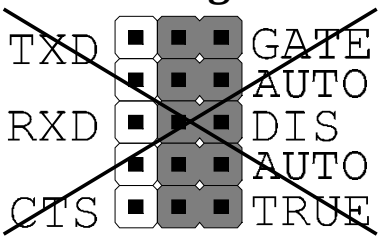
iii) CTS FORCE TRUE - Usually For RS485 Use.

This jumper determines whether the CTS handshake signal is INPUT from the external serial device or is permanently forced true on the card. The serial ports own RTS OUTPUT handshake signals is always taken out to the serial port connector.

NOTE: Though there are 5 option pin settings, on each 485 port gating block, there are only 3 jumpers provided - 1 for each of the TXD, RXD and CTS settings. I.e. You should only set TXD to GATE OR TXD to AUTO and NOT both, the same applies to RXD Gating.

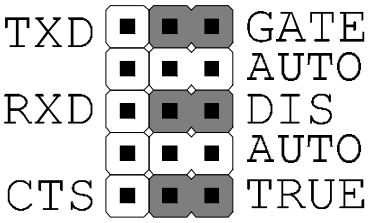
For example, in the diagram below, fig 1, is not acceptable; figs 2 and 3 are.

Fig 1



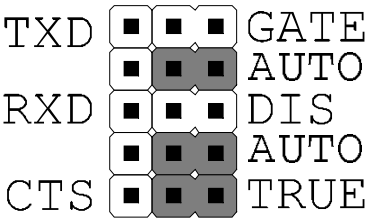
Not Valid

Fig 2



S/W gated

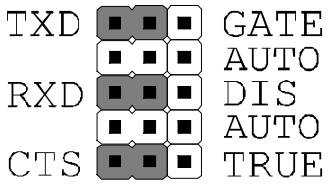
Fig 3



Autogated

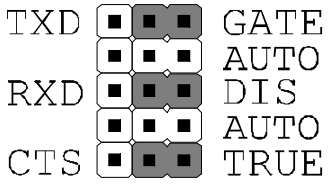
Figure 2-5. Port 1 Gating Jumper Block.

FACTORY DEFAULT SETTING:
ALL JUMPERS DISABLED (RS422 MODE)

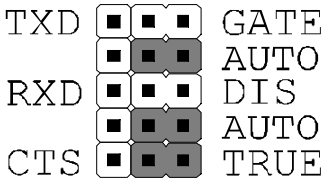


RECOMMENDED ALTERNATE SETTINGS:

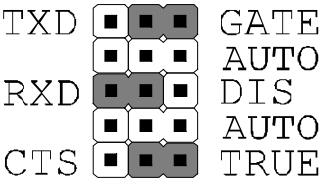
1. RS485 HALF DUPLEX OPERATION



2. RS485 HALF DUPLEX AUTOGATED



3. RS485 LISTEN TO EVERYTHING



Serial Port Multiplex Jumper Block.

By referring to the figure 2-1, locate the Serial Port RS485 Multiplex jumper block at the right of the middle side of the board. Shown below with default settings, this block consists of 4 rows of 3 pins, with 2 two of these rows controlling a port, the top two control port 1 and the bottom two control port 2. **N.B.** The worked example given below, in Figure 2-6, is for Port 1 multiplex settings, but also applies to Port 2.

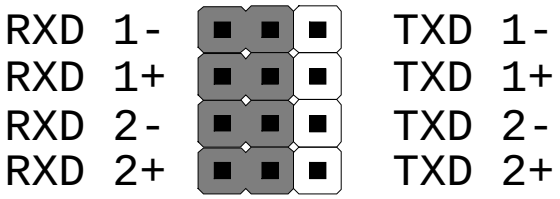
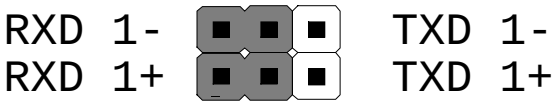
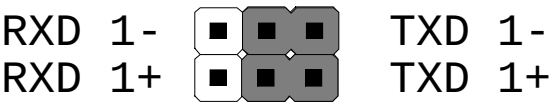


Figure 2-6. Port Multiplex Jumper Block.

FACTORY DEFAULT SETTING:
NOT MULTIPLEXED

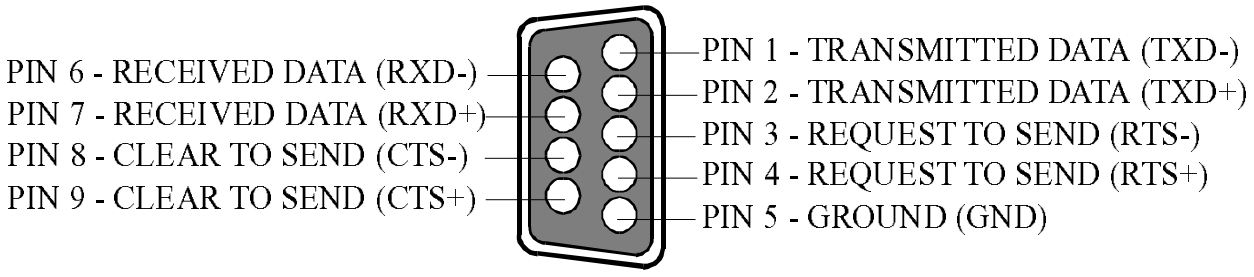


ALTERNATE SETTINGS:
RS485 HALF DUPLEX OPERATION



When the Multiplex jumpers are set to the right then the RXD- & TXD- signals and the RXD+ & TXD+ signals are shorted together at the 9 pin connector. The multiplex jumpers should only be set for RS485 Half Duplex operation, when one twisted pair is used to interconnect both transmit and receive lines.

Figure 2-7. - Serial Port Pin Outs.



RS422 Operation.

RS422 Handshake.

Generally, in RS422 systems all 8 signal lines from the 9 pin D connector participate in the data transfer sequence, thus 4 twisted pair cables are used. One twisted pair carries the TXD data outwards, one pair brings the RXD data inward, another pair carries the RTS handshake outwards and the fourth pair brings the CTS handshake inwards. There is no need to carry the ground from one device to another.

This RS422 arrangement allows data to be transmitted and received simultaneously since each signal has its own data cable pair. In addition, the receiver can set RTS true so telling the transmitter on its CTS input that the receiver is ready to accept data. In this way, no data will ever be transmitted when the receiver is unable to accept it, due to a full input buffer etc. And so no data will be lost.

The Serial Solutions for DOS software is set up to this option with the following line in the CONFIG.SYS file.

DEVICE=NEWCOM.SYS /H2,1 COM2 RS422

Where **H2** means we are setting the Handshake for COM2, and **1** sets handshake type 1 i.e. the RS422 RTS/CTS handshake. See the Serial Solutions manual for more detail.

Alternatively, the data could be transmitted using the XON/XOFF handshake and so remove the need for the RTS and CTS lines and the extra two twisted pair cables.

The Serial Solutions software is set up to this option with the following line in the CONFIG.SYS file.

DEVICE=NEWCOM.SYS /X2 COM2 XON/XOFF

RS422 Serial Port Cables.

Use screened twisted pair Belden cable 9729 and 9829, L type 2493 and 2919 or IBM Part No 4716748 cable to make the RS422 connection. Unscreened Belden type 8795 may also be used in less noisy environments.

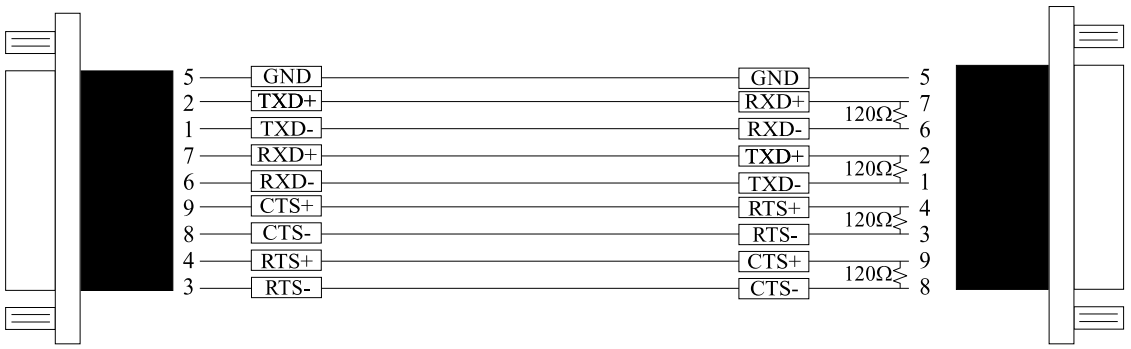
The on board resistor networks terminate the receiving end of the twisted pair cable in its characteristic impedance.

Figure 2-8. Serial Port 1 To Other PC Cable.

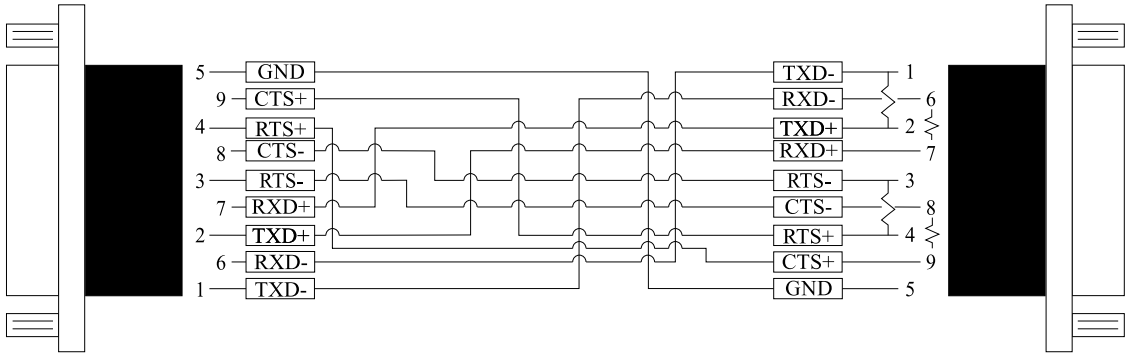
SERIAL PORT 1 Side
9 Pin Female D Connector

Other PC SERIAL PORT Side.
9 Pin Female D Connector

SCHEMATIC REPRESENTATION:



ACTUAL REPRESENTATION:



Note:

Receiver ends terminated in characteristic impedance ONBOARD resistor networks.

USE BELDEN TYPE 9729 etc. see above.

RS485 Gating & Multiplex Jumpers as Factory Set.

RS485 Operation.

The RS485 standard is intended for up to 32 driver receiver pairs on the bus. The line drivers used in the Serial Solutions RS422/485 card are designed to work correctly in both RS422 and RS485 systems. The main difference therefore is in how the system is implemented. Though the card uses a 9 pin D connector, in general, not all the lines are used for RS485 systems. The RTS+/- and CTS+/- lines, though driven by the card, are usually not connected. In two wire, Half-Duplex configurations the TXD+ line is connected to the RXD+ whilst the TXD- line is connected to the RXD-, only one pair of twisted wire cable is used in RS485 Half Duplex communications.

The hardware handshaking performed by the CTS+/- and RTS+/- lines in RS422 systems are handled by a software protocol in RS485 systems. In situations where more than one device may transmit data on the shared data line, each cards RTS line is used as a gating signal to enable the TXD driver only when that card needs to transmit data, i.e. set TXD GATE or AUTO jumper. This mechanism prevents bus contention caused by multiple transmitters holding the line in opposing states. Revision 3 and higher versions of this card have a facility which automatically “gates” the RTS line, thus enabling the transmitter independently of any software. This “Auto gating” is described in more detail in the previous 485 half duplex section

The three wiring schemes given described below are: -

i.) RS485 One Talker Many Listeners (HALF DUPLEX)
ii.) RS485 Many Talkers Many Listeners (FULL DUPLEX.)
iii.) RS485 Many Talkers Many Listeners (HALF DUPLEX.)

RS485 Cable.

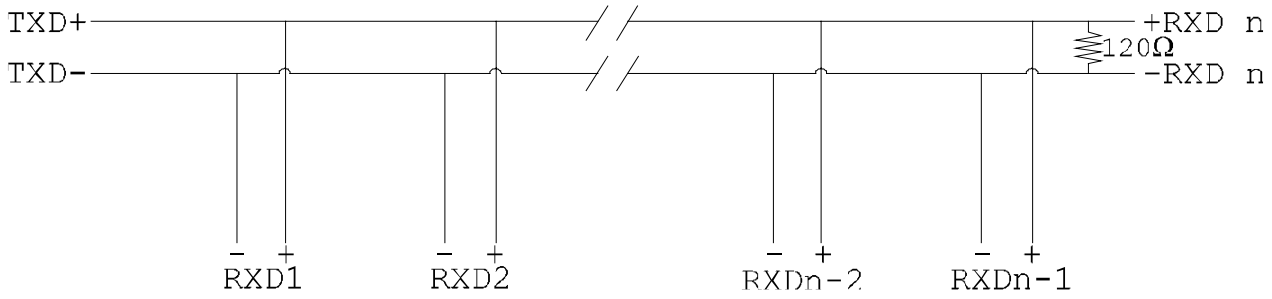
For best noise immunity use twisted pair cables to make the RS485 connection. In Half Duplex wiring only 1 twisted cable pair is needed. Two twisted pair cables are needed for Full Duplex communications.

Use screened twisted pair Belden cable 9729 and 9829, UL type 2493 and 2919 or IBM Part No 4716748 cable to make the RS485 connection. Terminate the twisted pair cable at either end in its characteristic impedance, which for the Belden 9729 cable is 120 Ohms. Unscreened Belden type 8795 may also be used in less noisy environments.

RS485 One Talker - Many Listeners.

There are several schemes for connecting RS485 devices depending on the characteristics of the system. In many cases there will be only one device, which can transmit, data and all the others simply listen to it. This scheme is used for theatrical lighting intensity control in the DMX512 standard. This is shown in Figure 2-9, below. For the talker the RS485 TXD GATE jumper should remain in the factory set position, i.e. transmitter is always enabled. There is NO multiplexing of the TXD and RXD lines. Data is only flowing one way, from PC outwards, and is thus a Half-Duplex configuration, only one twisted pair cable is needed.

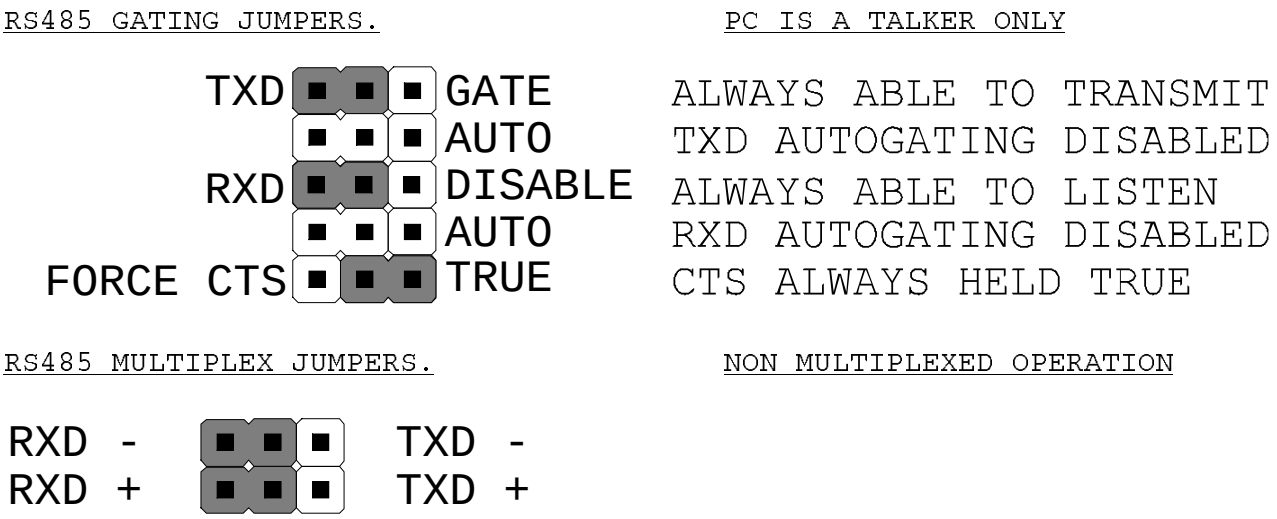
Figure 2-9. RS485 1 Talker Many Listeners.



Note: The Receiver end of MAIN line terminated in characteristic impedance by ONBOARD resistor networks, stubs off the main not terminated.

In the above scheme one RS485 device is talk only, it transmits data but it does not receive any. The other RS485 devices are receive only, they do not transmit any data at all. Figure 2-10 gives the RS485 Gating jumper settings.

Figure 2-10. RS485 Jumpers, 1 Talker Many Listeners.



RS485 Many Talkers- Many Listeners, Full Duplex.

The RS485 many talkers, many listeners, Full Duplex system can be used when all the RS485 devices have separate Transmit and Receive channels. There is NO multiplexing of the TXD and RXD signals on the same device. This system is especially useful when there is no flow control available on the PC, usually due to the use of a third party communications program that prevents the use of the RTS signal as a "transmit enable" control, via the TXD GATE jumper. It can be used in the following situations:-

a) The PC is connected to only ONE RS485 device.

b) The PC is communicating with several RS485 devices that are each able to recognise and respond to their own unique address.

The RS485 devices only drive their TXD lines when they are responding to requests from the PC to send data. In effect, the RS485 device's address and the command it receives is used to control access to the devices TXD channel.

This is a Full Duplex system. Two twisted pair cables are required. One twisted pair, is the PC's TXD channel, it carries the data sent from the PC's TXD outputs to the RXD inputs of each of the RS485 devices. The second twisted pair, is the Devices TXD channel, it carries the data sent from each of the devices' TXD

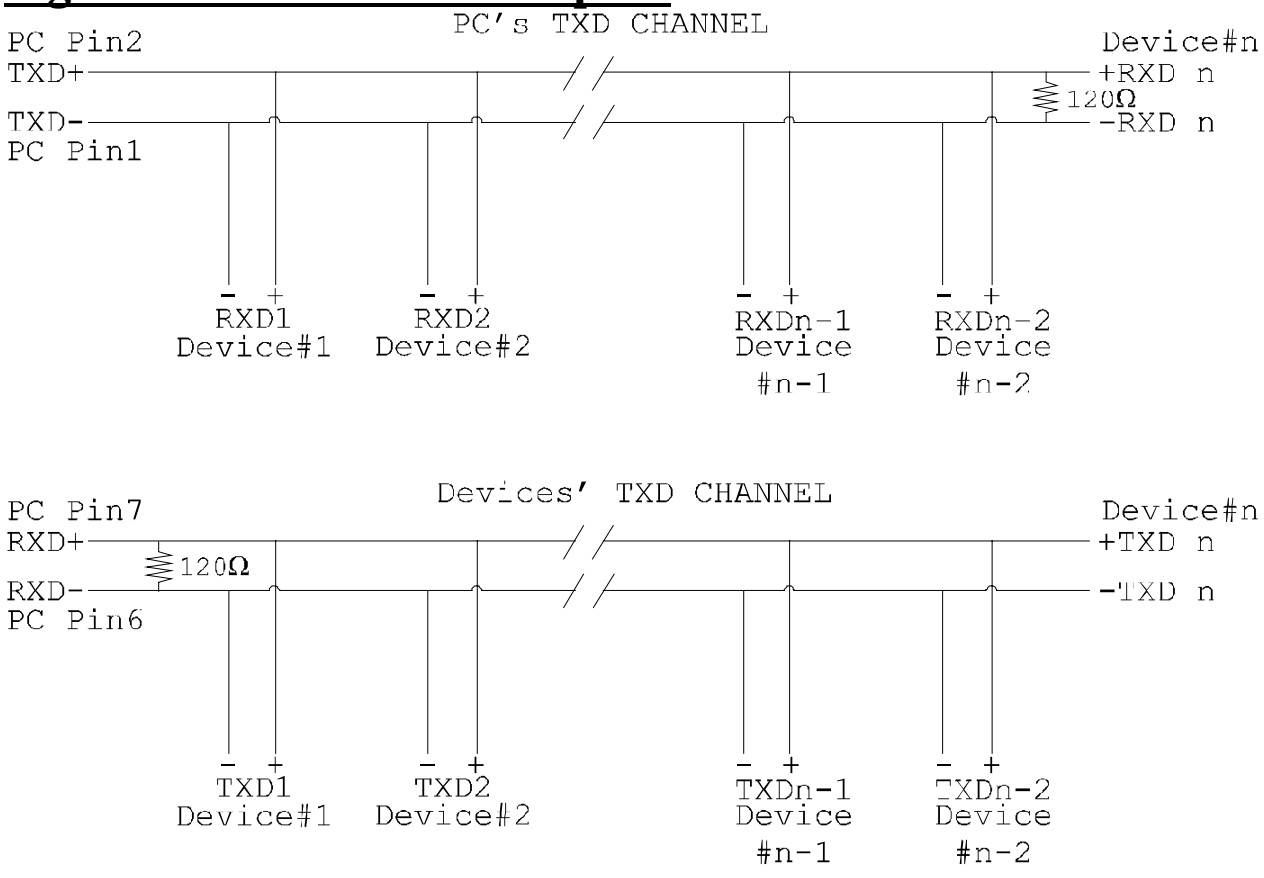
outputs to the RXD inputs of the PC.

The advantages of this system are great, since no new communications software is needed, and the PC can talk and listen at the same time. In effect the handshaking is performed by the intelligence of the RS485 devices attached to the PC.

When wired as in Figure 2-11 below, the PC can transmit data at any time and all the RS485 devices #1 to #n simultaneously receives it.

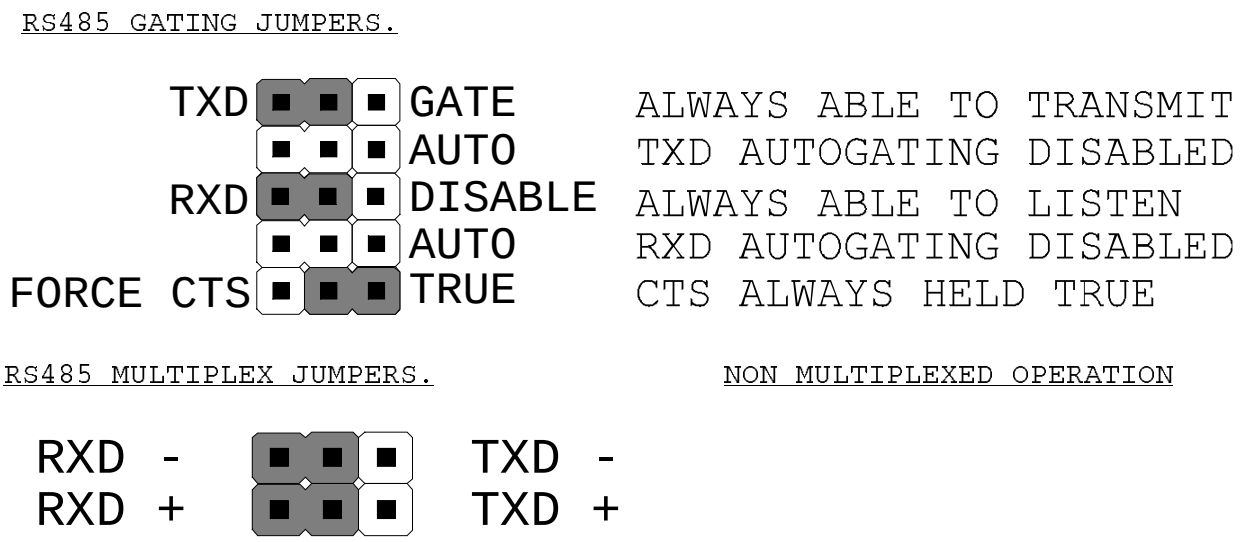
Only one of the RS485 devices may talk, i.e. transmit data, at any one time. Each RS485 device recognises commands and data addressed to it, it only talks when the PC commands it to do so. When the RS485 device receives the command to talk from the PC, it gates its TXD drivers on, sends the data down the device TXD channel, and disables its TXD drivers. The other RS485 devices remain in the receive only mode when they are not being addressed, they do not transmit any data at all. Figure 2-12 gives the jumper settings for the serial card in the RS485 Full Duplex mode.

Figure 2-11. RS485 Full Duplex.



Note: The receiver end of MAIN line terminated in characteristic impedance, stubs off the main not terminated.

Figure 2-12. RS485 Jumpers, Full Duplex.



RS485 Many Talkers- Many Listeners, Half Duplex.

Another popular RS485 layout is for multiple talkers and multiple listeners. This is shown in Figure 2-13, below. This is also known as "party line" transmission. It is imperative to have some method of preventing two devices trying to drive the data lines at the same time. The normal method is to use the RTS line as a talk enable. The RTS line should go true immediately prior to the data transmission and go false immediately after the last byte in the stream is sent. See Figure 2-13 for jumper settings.

The Serial Solutions software is set up to this option with the following line in the CONFIG.SYS file.

DEVICE=NEWCOM.SYS /H2, 2

Where **H2** means we are setting the Handshake for COM2, and **2** sets handshake type 2 i.e. the RS485 RTS enable handshake. See the Serial Solutions manual for more details.

If you have a revision 3 or higher card, then you can avoid

this software overhead by enabling the Auto gating option, see jumper settings below. This is ideal in Multitasking operating systems such as Windows NT and OS/2.

The Serial Solutions for DOS software is set up with the following line in the config.sys file.

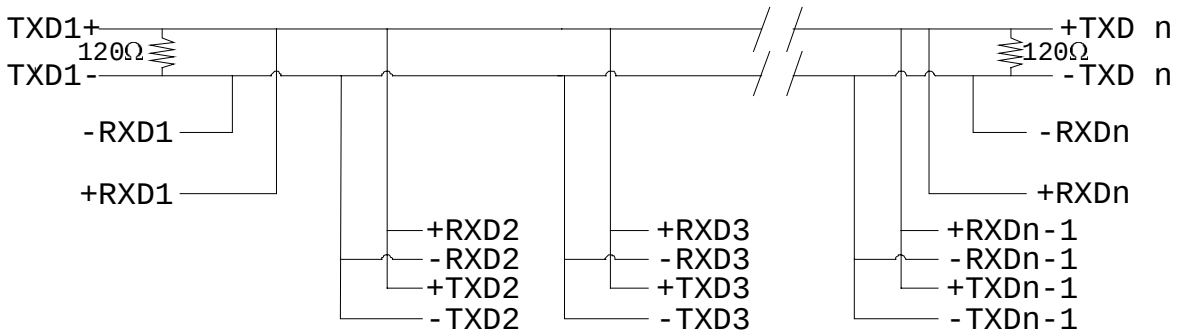
DEVICE=NEWCOM.SYS /H2 , 4

Where **H2** means COM2 and **4** sets handshake type 4 i.e. 3 wires, essentially no handshake. If you wish to ensure no data loss XON XOFF handshaking should be used but only if your external devices support it. XON XOFF handshaking is enabled in DOS for serial solutions by using:

DEVICE=NEWCOM.SYS /X*n*

Where ***n*** is the RS422/485 COM port to which the handshaking is to be applied. For more information on Serial Solutions for DOS commands etc., refer to Chapter 1 "**Introducing Serial Solutions**" or contact your product dealer.

Figure 2-13. RS485 Many Talkers & Listeners. Half Duplex



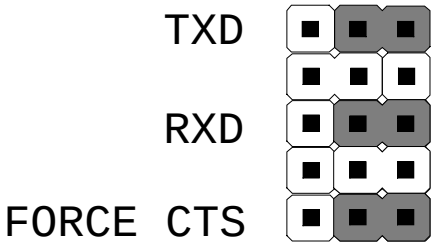
Note: BOTH ends of MAIN line terminated in characteristic impedance, stubs off main line not impedance, since both ends receive.

Note: The twisted pair ends are wired to both RXD+ & TXD+ and

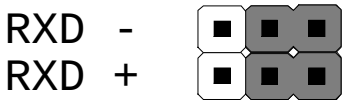
RXD- & TXD- at each RS485 device!

Figure 2-14. RS485 Half-Duplex Jumper Block.

RS485 GATING JUMPERS.



RS485 MULTIPLEX JUMPERS. MULTIPLEXED OPERATION



To program the RTS line true and false, if the base address of serial port 2 is BASE, and since RTS is bit1, decimal value 2 then:-

```
OUT BASE+4,3+8 REM RTS True TXD Enabled
OUT BASE+4,1+8 REM RTS False TXD Disabled
```

Note: The 1+8 keep DTR and OUT2 true.

Programs to take advantage of this scheme are given later in Chapter 5 - “Serial Utility Disk and Programming Guide. ”

CHAPTER 3

INSTALLING THE PC CARD IN THE COMPUTER.

Serial Card Installation.

Once the card has been correctly configured then it can be installed in the PC. For the ISA card it is best to make a note of the serial port I/O address selection and if appropriate IRQ jumper settings for later use.

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, everything 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.!

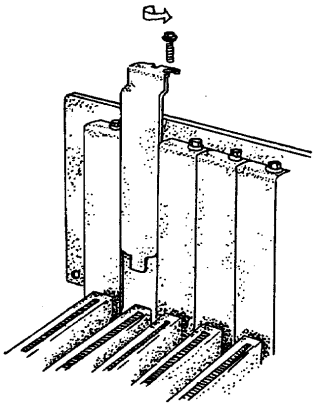
NOTE: Always turn the computer OFF before installing or removing any interface board..!!!

After having made sure that the I/O address and if appropriate 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!**

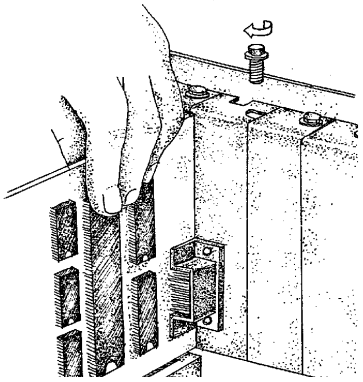
STEP 2: Remove the case.

Figure 3-1. Removing Blanking Cover



STEP 3: Choose an empty appropriate expansion slot. Remove the blanking cover protecting the slot on the PC back panel. KEEP the blanking cover screw safely for later (Figure 2-14).

Figure 3-2. Inserting The PC Serial Card.



STEP 4: Now insert the PC Serial card in the available 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 (Figure 2-15).

STEP 5: The 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 6: Now replace the system units cover by carefully sliding it down and back over the system unit. Replace the cover mounting screws.

STEP 7: 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!

■ 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

CHAPTER 4

AT DUAL RS422/485 AND AT VELOCITY RS422/485 SOFTWARE CONFIGURATION

Introduction

This section contains the installation procedures for the AT Dual RS422/485, AT Velocity RS422/485 and AT Opto Dual RS422/485 cards, for the operating systems DOS, Windows 3.x, Windows 95, Windows 98, Windows NT, OS/2 version 2.x and OS/2 Warp

In each section, as an example, the AT Dual RS422/485 with its factory default address selection has been used in the listed set-up procedure. The IRQ selections have been changed for indication only. You should change to these settings ONLY if your PC configuration allows this, refer to previous chapters for details on changing these settings. The set-up procedures in this chapter also assume that your PC has only one serial port present. For more information on configuring any of the Dual RS422/485 cards refer to Chapter 2 "**AT Dual RS422/485 and AT Velocity RS422/485 Hardware Installation Guide**". If you have changed the default settings of the cards in any way, then substitute the appropriate values in the relevant sections.

Informing The PC Of The Port Address.

The BIOS of most new PC's automatically detects whether serial ports COM1 - COM4 addresses are present in a machine. Other, older PC's are capable of detecting serial ports COM1 - COM2, but the problem comes with serial ports COM3 and COM4. PC's have a table of information that stores the set-up of the PC, this is the BIOS data area. It has space for 4 serial port addresses. The

trick, for the older PC's, is to get the right addresses into the third and fourth port areas. However, PC's are not capable of recognising ports above COM4 - specific device drivers are responsible for this.

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 are only allowed access to COM1 and COM2, though some manufacturers have 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 and Programming Guide**" 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 an AT Dual RS232 series 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.

Microsoft Windows Version 3.x.

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

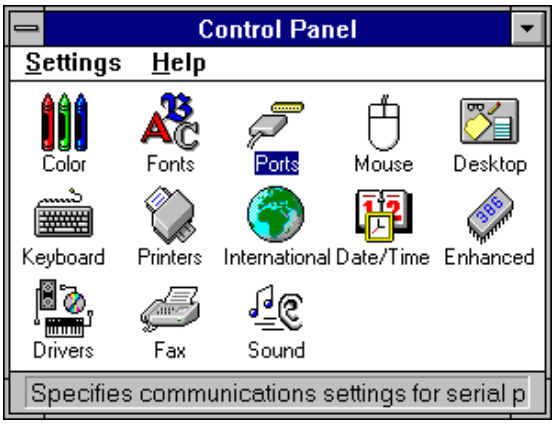
If more than 2 serial ports are installed then the default assignments assumed by Windows need to be changed. Windows assumes that both COM1 and COM3 are on IRQ4 and that both COM2 and COM4 are on IRQ3. Since the PC bus does not allow interrupt sharing, these pairs of ports cannot be used simultaneously,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

To obtain trouble free mix and match of the COM ports,

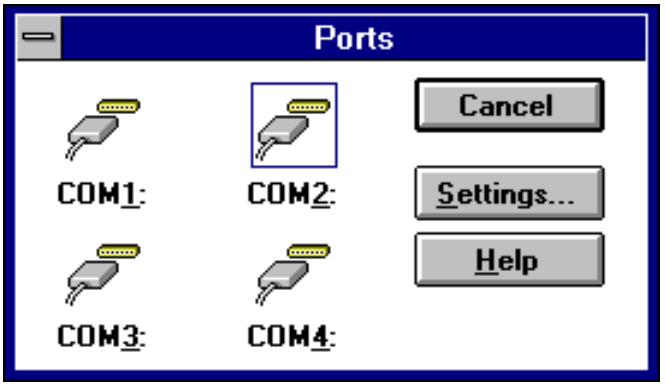
● Set COM3 and COM4 interrupts to ones not used by other devices on your system, say IRQ10 and IRQ11.

● From the Windows Program Manager select **Main**. Then double click the **Control Panel** icon.

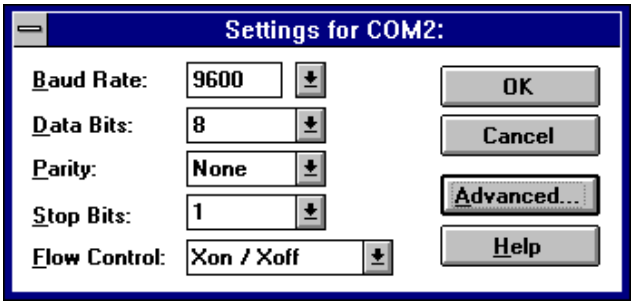
● From the Control Panel window select **Ports.**



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



● From the COM2 window click the **Advanced** box.

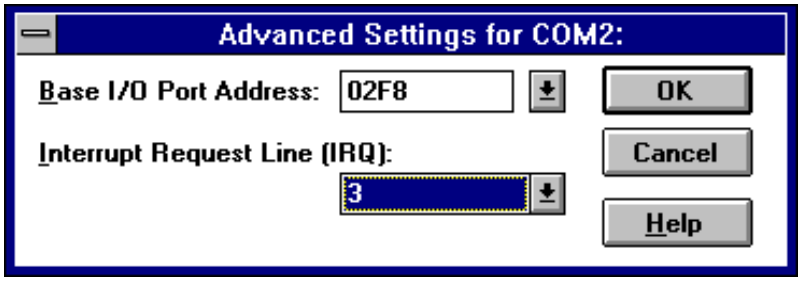


● From the Advanced Settings for COM2: window select **Interrupt Request Line**(IRQ):

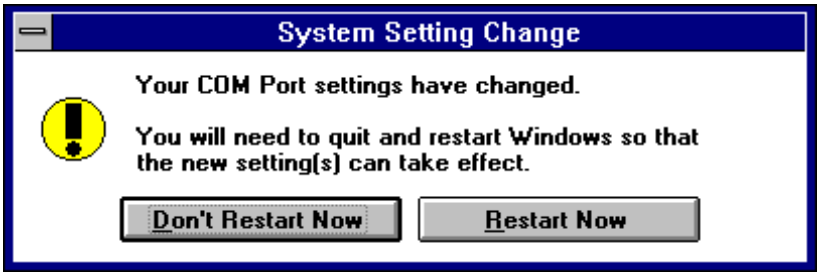
● Scroll through the list of interrupts and select IRQ3 by clicking on 3.

● 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.



● Click on Restart Now and let windows restart itself so that it is then able to run with the new configuration.

OR

● Click on Don't Restart Now and step back though the various windows and 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.

Microsoft Windows 95 & 98.

NOTE: All mention to Windows 95 infers reference to Windows 98 also. Unless it is noted within this section, the installation screen shots for Windows 95 and 98 are identical.

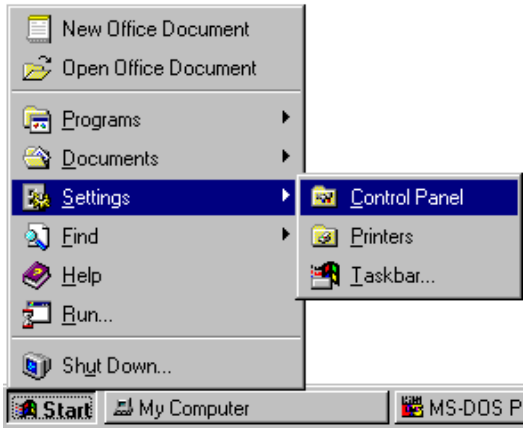
The Windows 95 environment now supports up to 255 standard serial ports, RS232, RS422 and RS485 etc.

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 got 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.

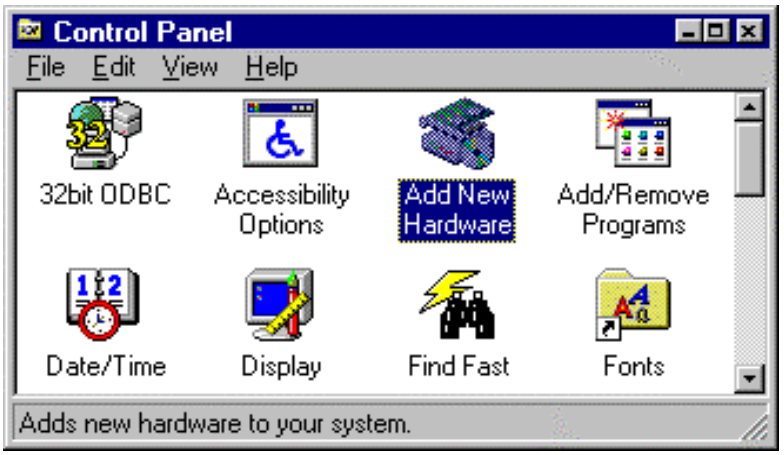
To obtain trouble free mix and match of the COM ports, assuming that only 1 Serial Port already exists on your PC,

- Set COM2, and COM3 interrupts to ones not used by other devices in your system, say IRQ3, and IRQ10 respectively. Plug in the card.

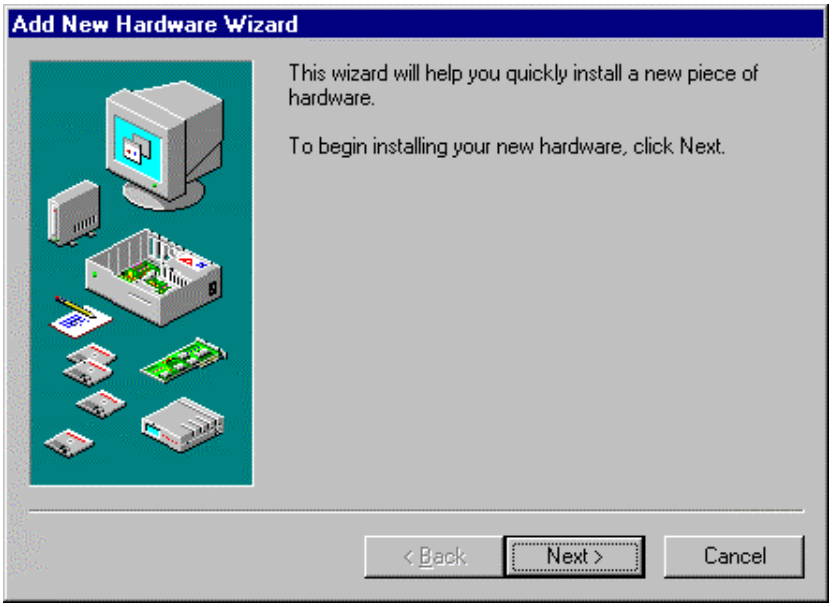
- Click the **Start** button and select **Settings**, then **Control Panel.**



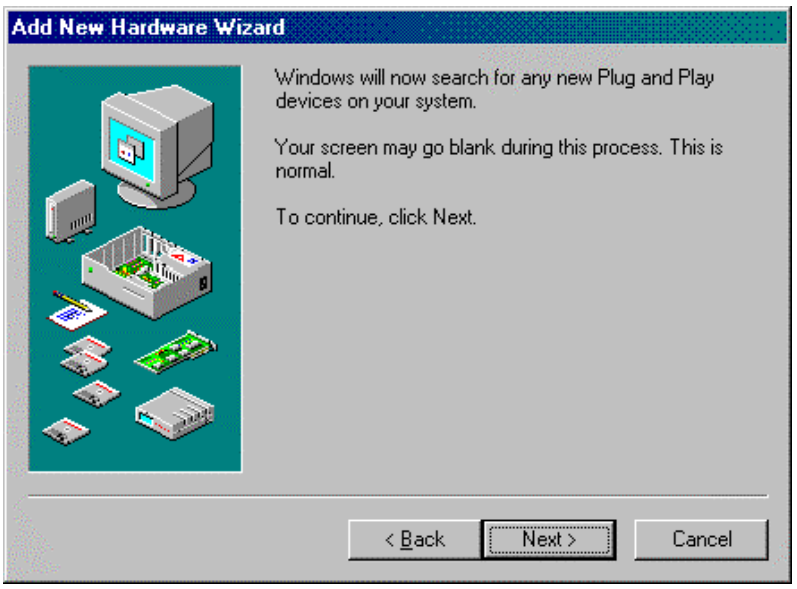
● Select **Add New Hardware.**



● Click **Next.**



● Click **Next** (Windows 98 only):



● Click **Next** (Windows 98 only):

If any of the AT OPTO, DUAL or VELOCITY RS422/485 ports are to be set to values of COM4 or greater then step to the next section entitled "**Installing as COM5 and higher.**"

Installing as COM4 and lower

If you are installing the AT OPTO, DUAL or VELOCITY RS422/485 ports as COM4 or lower, then continue.

● Windows 95 will respond with the following - click **Next**.

● Windows 98 will respond with the following. Click **Next.**

● Windows will prompt you with the following dialogue box, from which you should click **next**.

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

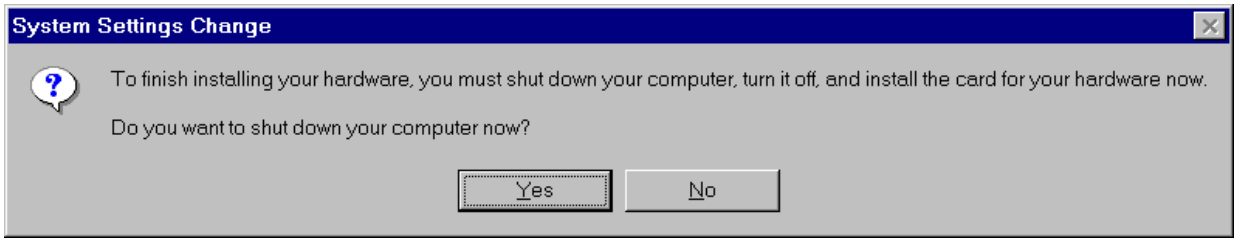
When Detection has finished, widows will inform you that it has finished detecting your new hardware.

● Click on **Details** to see what Windows has detected:

If you have set up the card as COM2, and COM3 then the view of detected ports should look somewhat like the following:

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

Windows will then ask you if you wish to restart.



If you have finished installing all ports, and you are happy with their configuration, then:

● Click **YES**

Otherwise:

● Click **NO** and either install the next port by following the procedure above, or reconfigure the port you have just installed by following the procedure below in the section "**changing port settings in Windows 95/98."**

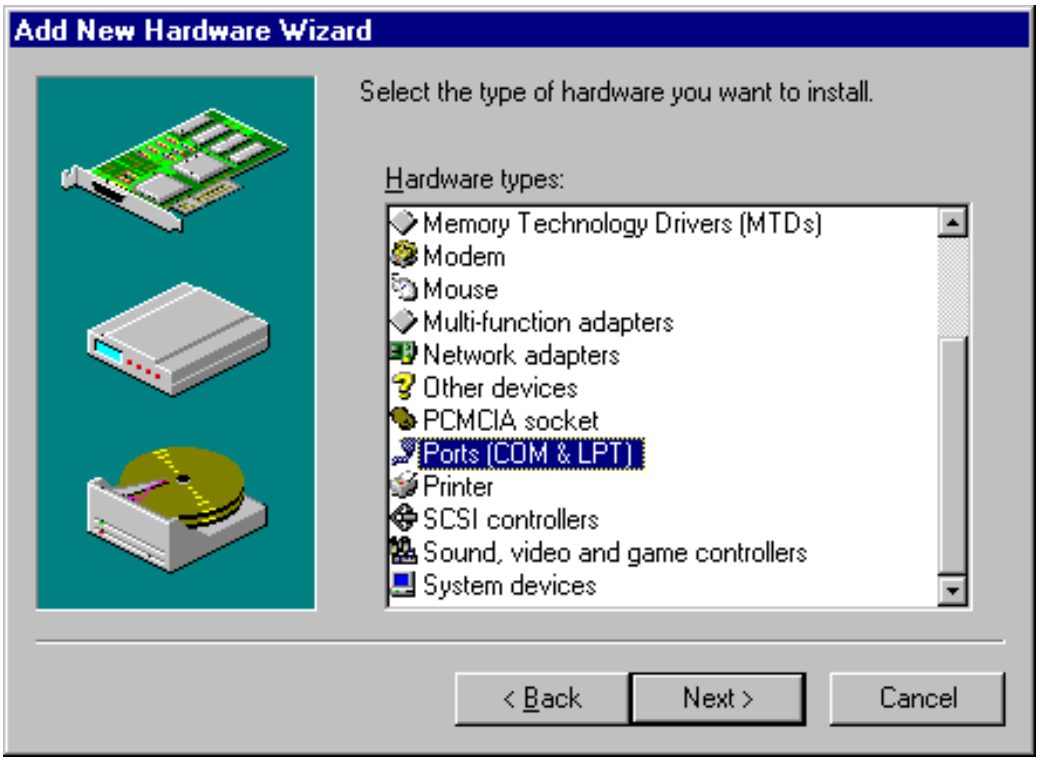
Installing as COM5 and Higher

If you are installing the AT OPTO, DUAL or VELOCITY RS422/485 as COM5 or higher, then on the second screen of the Add New Hardware Wizard, you should:

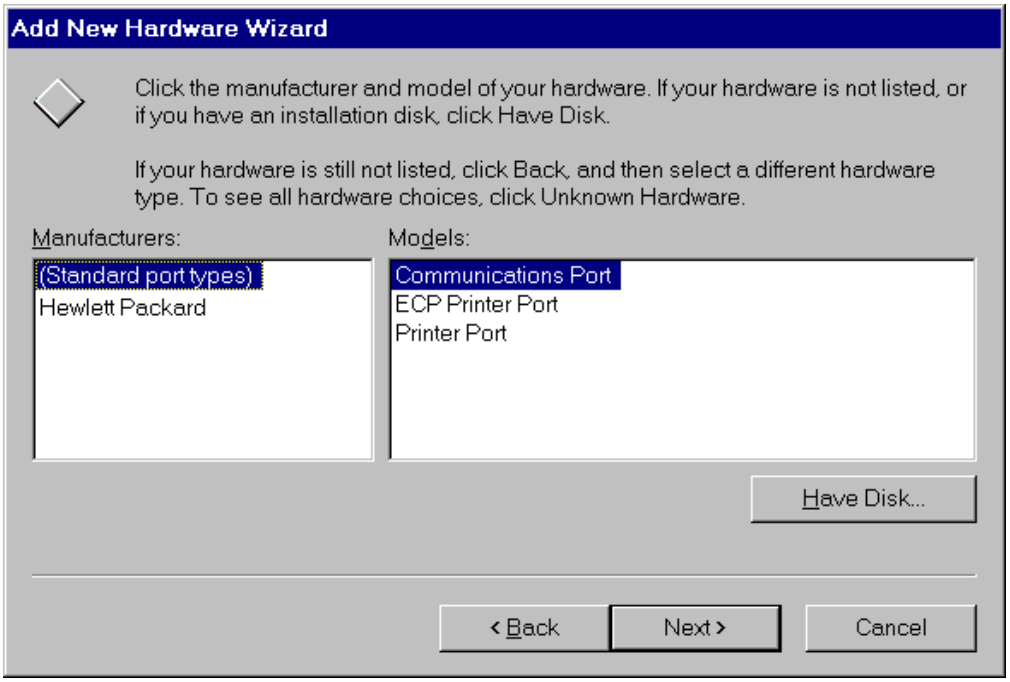
● Click **Next**. Do Not let Windows Detect your new serial ports. It will take forever and is not guaranteed to detect the correct settings. Windows 95 will display the following:

● Windows 98 will produce dialogue with the same information as above, select "**NO**", click **Next**.

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



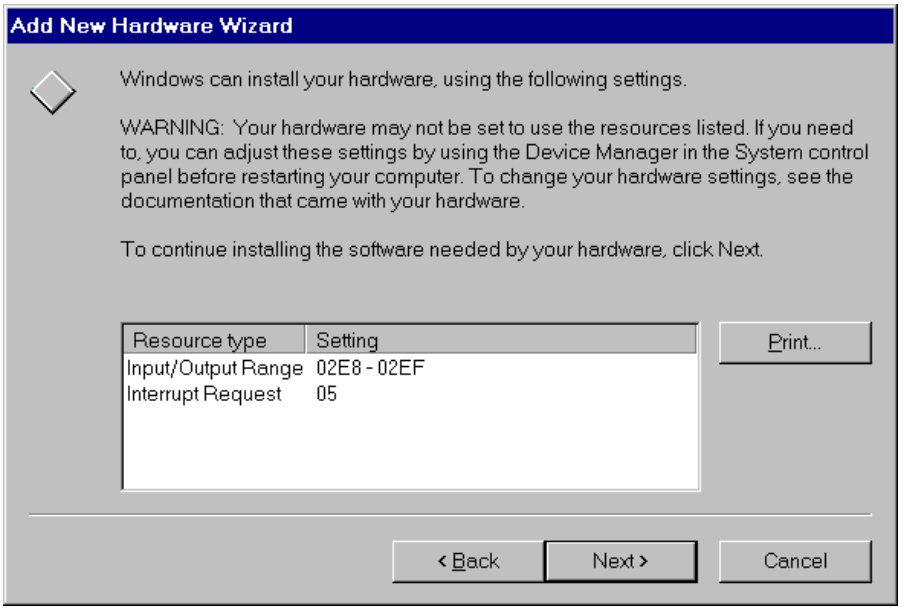
● Highlight **Standard Port Manufacturers** in the left window, and **Communications Port** in the right one, Click **Next.**



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 titled

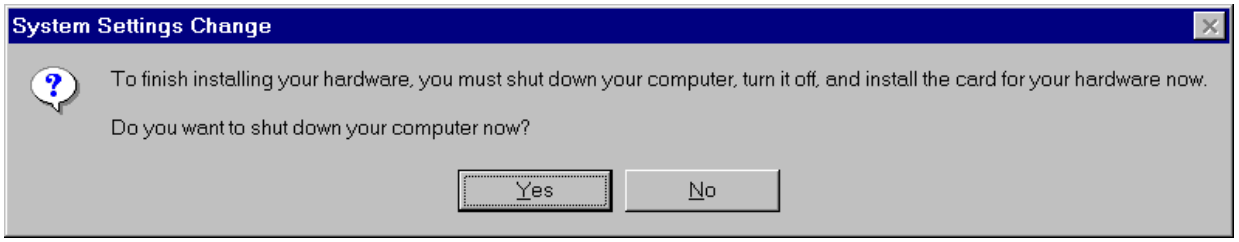
"**changing port settings in Windows 95/98."**

● Click **next**.



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

Windows will then ask you if you wish to restart.



If you have finished installing all ports, and you are happy with their configuration, then:

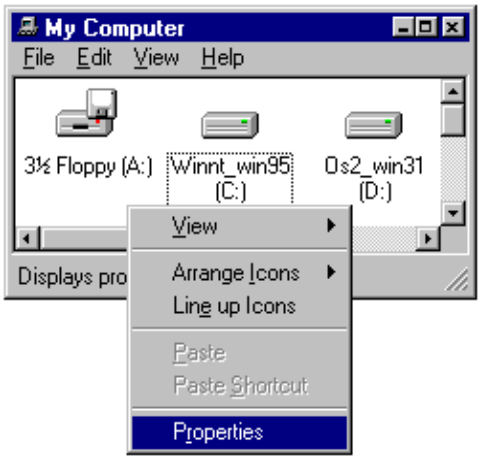
● Click **YES**

Otherwise:

● Click **NO** and either install the next port by following the procedure above, or reconfigure the port you have just installed by following the procedure below in the section "**changing port settings in Windows 95/98."**

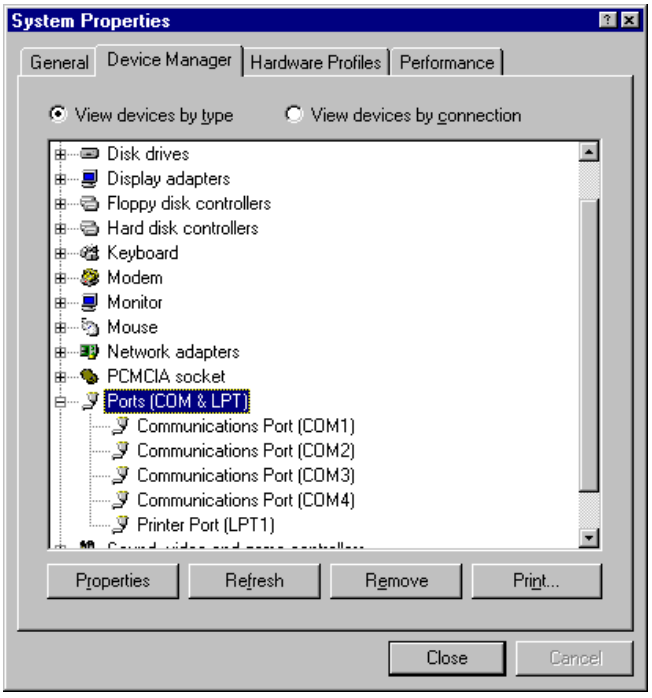
Changing Port Settings in Windows 95/98

All port settings are changed via the Windows 95/98 Device manager. This is most easily reached by opening My Computer, Right clicking in white space and selecting properties from the drop-down list.

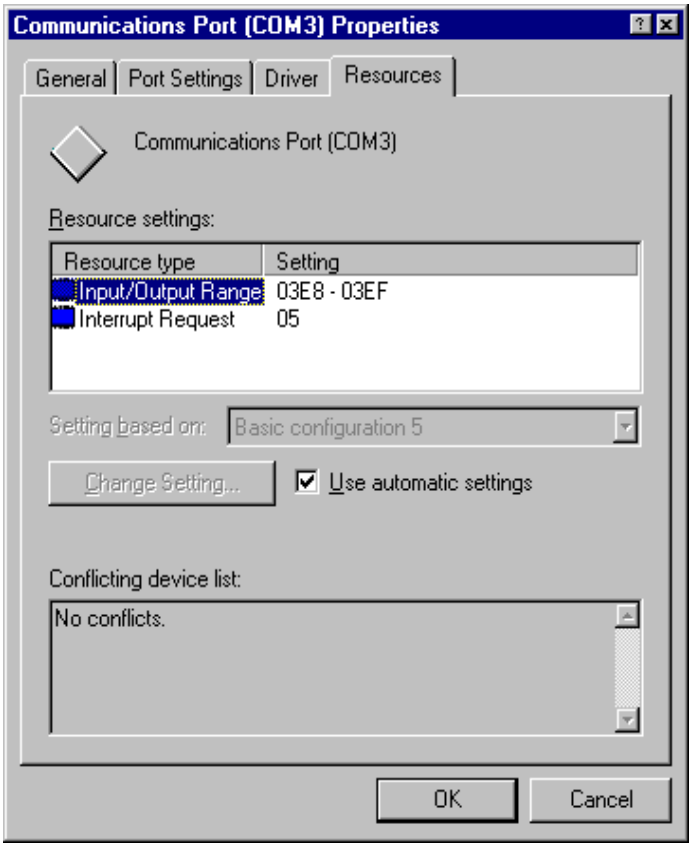


Select the **Device Manager** Tab, then click the **View devices by type** button, then:

● Double click **Ports (COM & LPT).**

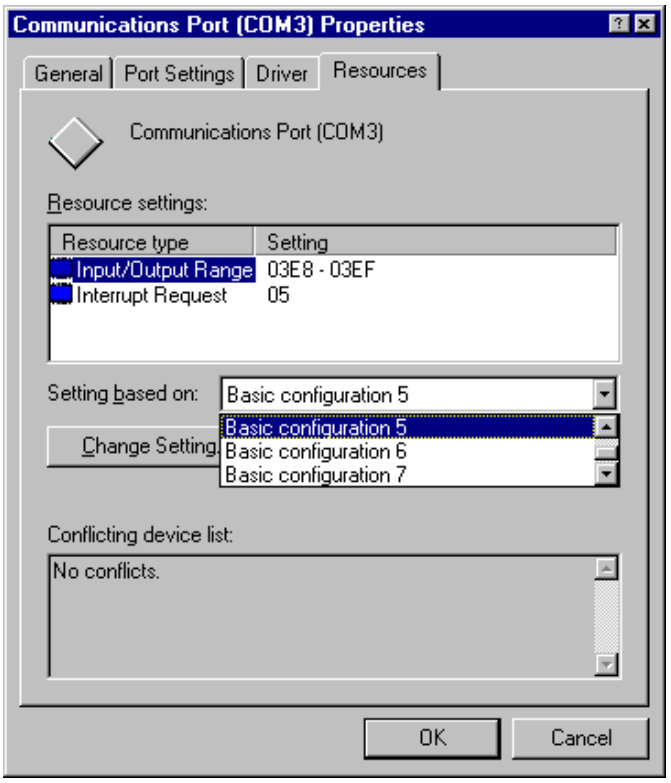


● Double Click the port whose settings you wish to change, then select the **Resources** tab.

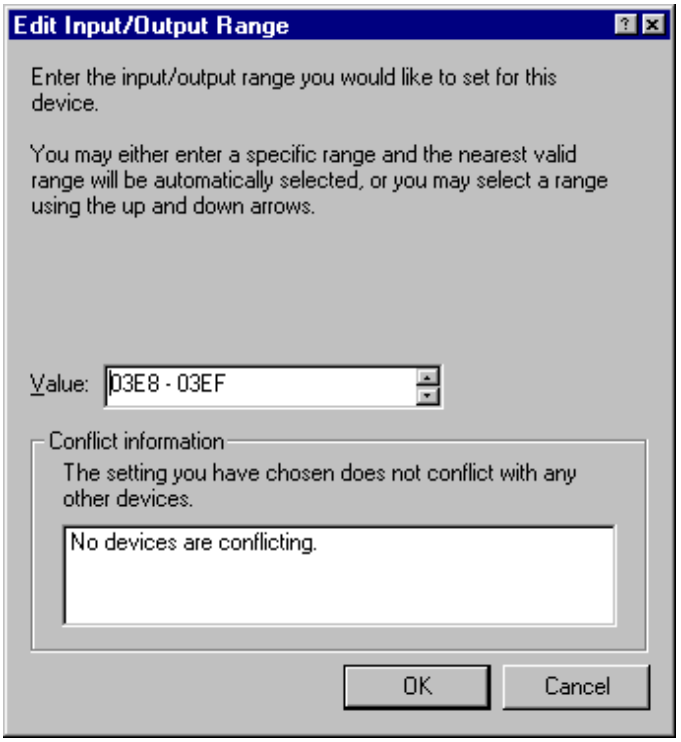


● Un-check the Use automatic settings box, and double click on the resource that you wish to modify. In some circumstances, windows

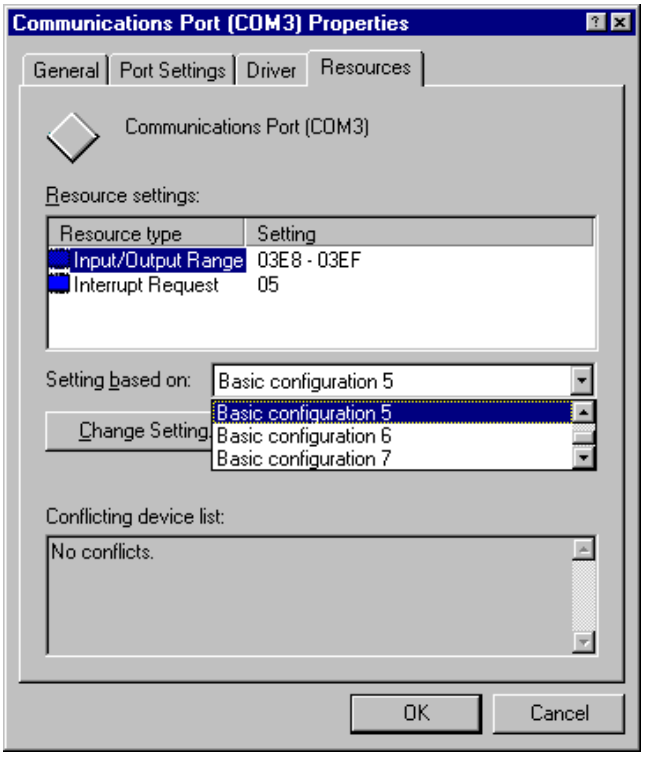
will not let you change these settings. This is easy to work around. Simply click the downward pointing arrow on the side of the box labelled **Setting based on**. Change the configuration to a higher number until Windows will let you change the resource setting that you want.



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



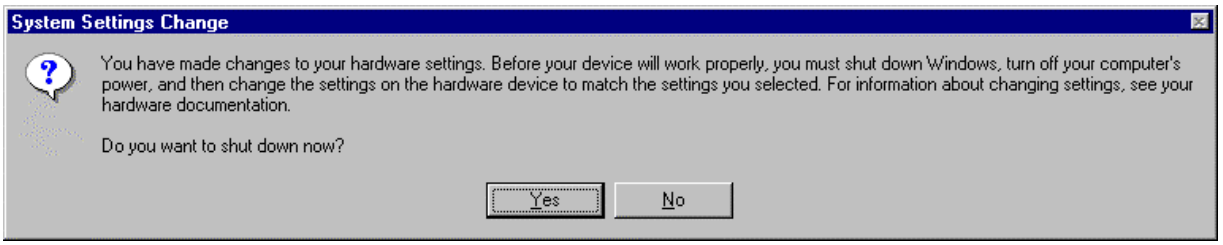
● When you have finished changing the settings for that particular port, click **OK**. If you have made changes, Windows will ask if you

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.

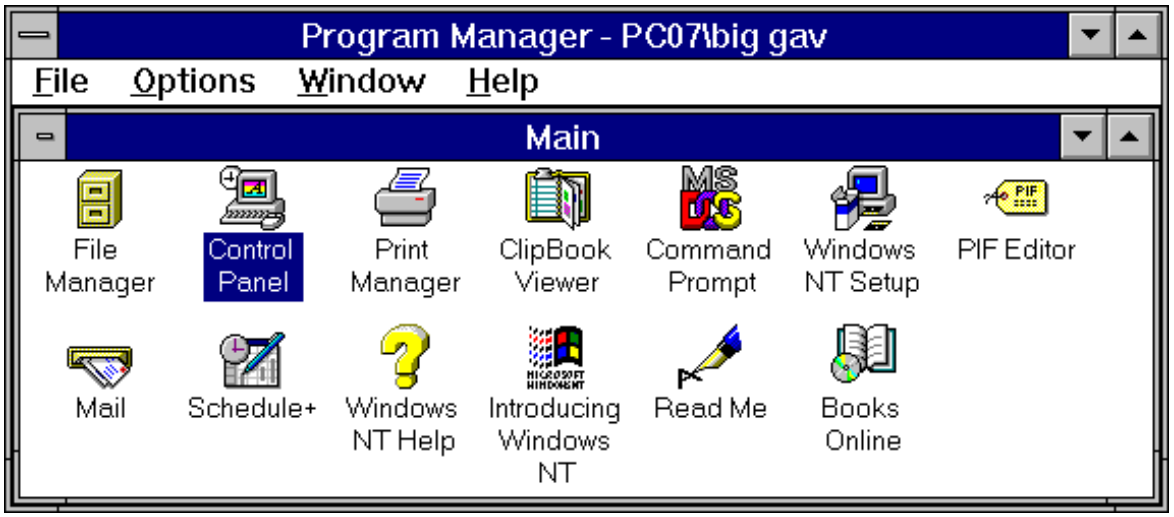
To obtain trouble free mix and match of the COM ports, assuming that only 1 Serial Port already exists on your PC,

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

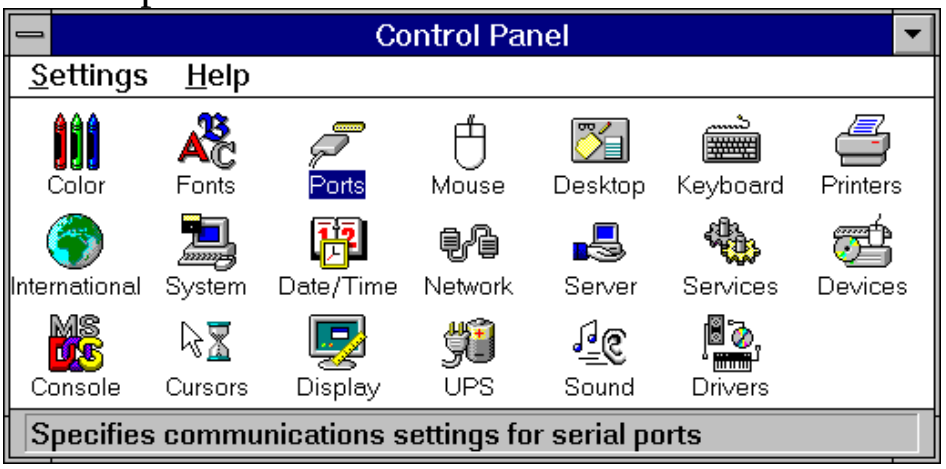
Version 4.0: ● Open 'My Computer' and double click the **Control Panel** icon or click the start button, select **Settings** and then **Control Panel**

● Set COM2, and COM3 interrupts to ones not used by other devices in your system, say IRQ3, and IRQ10 respectively. Plug in the card.

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



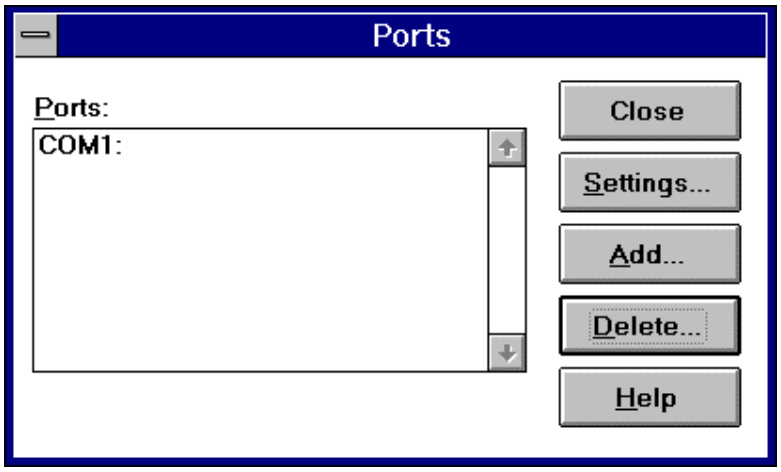
● From control panel Double click the **Ports** icon.



If you only have one existing port in your PC then your ports applet will look something like the following figure.

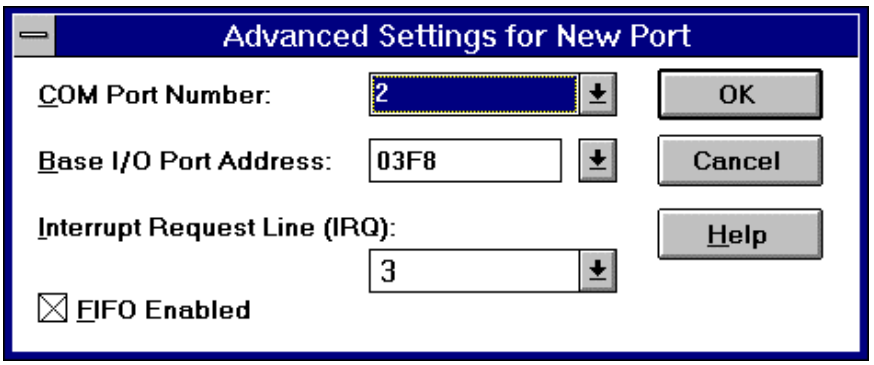
Your two new ports will need to be added one at a time. Each ports installation procedure is equivalent to the other from this point.

● 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**.

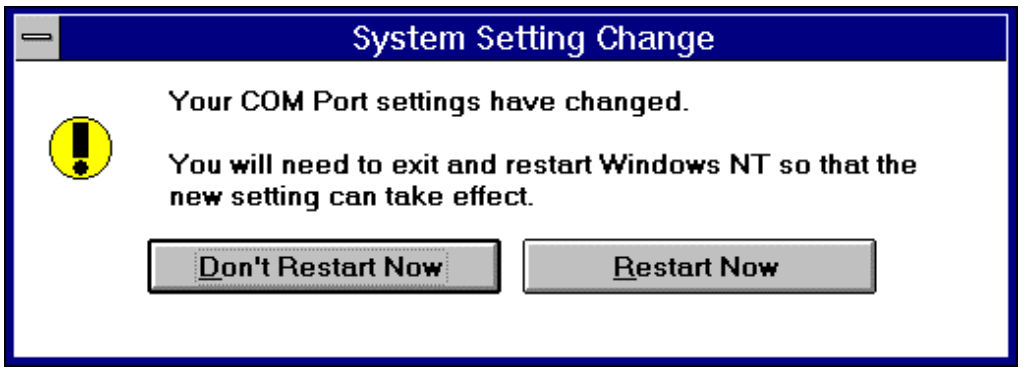


Windows will now tell you that for your settings to take effect, you will need to restart your machine. If you have more ports to install:

● 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 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 both COM3 using IRQ10 and COM4 using IRQ11 the CONFIG.SYS file entry should be:-

DEVICE=COM.SYS (3,3E8,10) (4,2E8,11)

Ensure that 1 space character is between the closing bracket ,10) of COM3 setting and the opening bracket (4, of COM4 setting.

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.

CHAPTER 5

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.

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 set-up 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 ADDRCOM3/4.

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 5-1 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 5-1. 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 5-1, 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 5-2. Parameters within square brackets [ ] are optional.

Figure 5-2. 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:-
Setting the baud rate.
Setting data word length, parity and stop bits.
Setting output lines RTS and DTR.
Reading the state of input lines CTS, DSR and DCD.
Detecting incoming data, RXD.
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
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 evice
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 5-3. 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	2 Serial
			AT Dual RS422/485 Card	+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
TL16C750	87ns	8	AT Velocity RS232	64 byte FIFO
			AT Velocity RS422/485	1MBaud
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 except for the velocity range. The equivalent buffered part, 16C552 can be fitted on request.

The Velocity range of cards uses a further enhanced version of the 16450, the 16750, which has a MASSIVE 64 byte FIFO buffer and is capable of transmission speeds of up to 1MBaud, whilst still remaining backwards compatible with the 16450.

Note: The 16C750 chips do not have a Centronics Printer interface. Thus the Velocity cards do not have a printer port option on them.

Figure 5-4. 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.

Index