MI PORT™

Multi-Interface PCI Bus Serial Cards

(RS-232 / RS-422 / RS-485)

Models 3PCIU2, 3PCIU4, 3PCIU8

(Non-Isolated)

and

Models 3PCIOU1, 3PCIOU2, 3PCIOU4

(Isolated)

B&B ELECTRONICS

International Headquarters

B&B Electronics Mfg. Co. Inc.
707 Dayton Road
Ottawa, IL 61350 USA

Phone (815) 433-5100 -- **General Fax** (815) 433-5105
Website: **www.bb-elec.com**

European Headquarters

B&B Electronics
Westlink Commercial Park
Oranmore, Co. Galway, Ireland

Phone +353 91-792444 -- **Fax** +353 91-792445
Website: **www.bb-europe.com**

CAUTION:
This is an Electrostatic Sensitive Device. Use ESD precautions for safe handling.

Before removing the card from the anti-static protective packaging:

- Discharge any static electricity buildup on your body by touching a large grounded metal surface or the metal chassis on equipment connected to earth ground by a 3-wire power cord. Use of a grounding wrist strap is recommended.

- Avoid touching the gold connectors or other parts on the card except when necessary to set the configuration DIP switches.

- Remove AC power from the computer before inserting the card

B&B Electronics shall not be liable for incidental or consequential damages resulting from the furnishing, performance, or use of this manual.

All brand names used in this manual are the registered trademarks of their respective owners. The use of trademarks or other designations in this publication is for reference purposes only and does not constitute an endorsement by the trademark holder.

Table of Contents

Chapter 1: General Information

Introduction

MIport PCI serial interface cards allow you to add RS-232, RS-422 and RS-485 interfaces to Windows based computers equipped with a PCI bus. Depending on your choice of card one, two or four optically isolated serial ports, or two, four or eight non-isolated serial ports, can be added. MIport PCI serial cards are Plug and Play compatible, which allows the Windows Operating System and driver to set the addresses and IRQ used by the card.

Figure 1. A 3PCIOU2 Optically Isolated Two-Port PCI Serial Interface Card

Features

- Optically isolated or non-isolated models available
- Isolated models available in 1, 2 or 4 ports
- Non-isolated models available in 2, 4 or 8 ports
- 4 and 8 port models include fanout cables
- Plug & Play compatible – Windows sets addresses and IRQ used
- 5 volt and 3.3 volt PCI bus compatible
- PCI-X compatible
- Conform to the PCI V2.3 Universal PCI specification
- RS-232/RS-422/RS-485 interfaces
- 2-wire or 4-wire RS-485 operation (half or full-duplex)
- Automatic Send Data Control for RS-485 operation
- Buffered high speed XR17D15x PCI Bus UARTs (16C550 compatible) with 64 byte FIFOs for input/output with programmable trigger thresholds
- Advanced driver function for COM port rename (Refer to Chapter 4)
- Supports baud rates up to 460.8 kbps
- Windows XP, Windows 2008 Server (32/64 bit), Windows Vista (32/64 bit), Windows 7 (32/64 bit), Linux Kernel 2.6.x is also supported in the following distributions: Ubuntu 10.04 Desktop Edition (32/64 bit), Ubuntu 10.04 Server Edition (32/64 bit)

MIport Models and Features

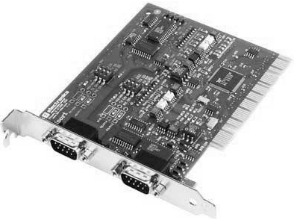
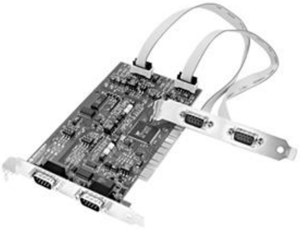
Model Number	Ports	Interface	Connectors
3PCIOU1	1	RS-232/422/485	DB-9 male
3PCIOU2	2	RS-232/422/485	DB-9 male
			
3PCIOU4	4	2 RS-232/422/485 ports 2 RS-422/485 ports	2 x DB-9 male 2 x DB-9 male use 2nd expansion slot
			

Figure 2. MIport Optically Isolated PCI Cards

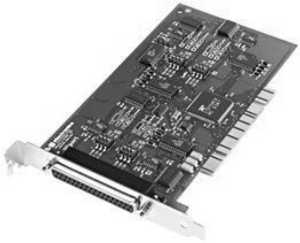
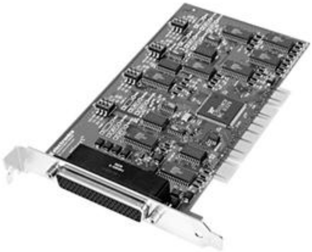
Model Number	Ports	Interface	Connectors
3PCIU2	2	RS-232/422/485	2 x DB-9 male
3PCIU4	4	RS-232/422/485	DB-37 female (plus DB-37 to 4x DB-9 male cable)
			
3PCIU8	8	4 RS-232/422/485 ports 4 RS-232 ports	DB-78 female (plus DB-78 to 8x DB-9 male cable)
			

Figure 3. MIport Non-Isolated PCI Cards

Specifications

OS Supported	Windows XP, Windows 2008 Server (32/64 bit), Windows Vista (32/64 bit), Windows 7 (32/64 bit), Linux Kernel 2.6.x is also supported in the following distributions: Ubuntu 10.04 Desktop Edition (32/64 bit) and Ubuntu Server Edition (32/64 bit).	
Bus	PCI bus (33MHz/32-bit) PCI Bus specification	
Slot	Requires one PCI slot (3.3V or 5V signaling)	
Baud Rates	**Maximum:**	Up to 460.8 kbps (RS-232/422/485)
	Typical:	75, 110, 134, 150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14.4k, 19.2k, 38.4k, 57.6k, 115.2k, 230.4k, 460.8k
UARTs	XR17D15x (16C550 compatible) with 64 byte FIFO buffers	
Character Length	5, 6, 7 or 8 bits	
Parity	Even, odd, none, space or mark	
Stop Bits	1, 1.5 or 2	
Optical Isolation	2000 VDC minimum on all lines	
(3PCIOU1, 3PCIOU2, 3PCIOU4 only)	Ports are isolated from the PC power and ground, as well as other ports on the same card.	
Connectors	**3PCIOU1:**	1 – DB-9 male
	3PCIOU2:	2 – DB-9 male
	3PCIOU4:	2 – DB-9 male (RS-232/422/485 ports) 2 – DB-9 male via ribbon cable on 2nd expansion slot bracket (RS-422/485 ports)
	3PCIU2:	2 – DB-9 male
	3PCIU4:	1 – DB-37 female with DB-37 to 4 x DB-9 male cable
	3PCIU8:	1 - DB-78 female with DB-78 to 8 x DB-9 male cable (4 x RS-232/422/485 and 4 x RS-232)
Data Signals	**RS-232:**	TD, RD, RTS, CTS, DTR, DSR, DCD, RI and GND (TD, RD, RTS, CTS, GND only on 3PCIOU4)
	RS-422:	TD(A)-, TD(B)+, RD(A)-, RD(B)+ and GND
	RS-485:	Data(A)-, Data(B)+ and GND
Environmental	Operating temperature range: 0º to 50ºC minimum Operating humidity: 5% to 95%, non-condensing	
Dimensions	4.8 x 3.8 in (12.2 x 9.6 cm) card edge (Mounting bracket, 1.2 x 12.1 x 0.9 cm)	
Accessories	**Software:**	Driver CD-ROM disc for Windows 98, ME, NT, 4.0, 2000, 2003 Server, XP, Vista, and Linux 2.4
	Manual:	Instruction Manual Contained on CD ROM

Figure 4. MIport Serial Card Specifications

Chapter 2: Serial Card Setup

The following Serial Card Setup section applies to the following PCI cards:

- 3PCIOU1 one port optically isolated PCI serial card
- 3PCIOU2 two port optically isolated PCI serial card
- 3PCIOU4 four port optically isolated PCI serial card
- 3PCIU2 two port non-isolated PCI serial card
- 3PCIU4 four port non-isolated PCI serial card
- 3PCIU8 eight port non-isolated PCI serial card

Any deviations from the procedure for specific models are noted.

Pre-Setup Steps

Your serial card has been tested for proper operation before packaging and shipping. It should be in perfect mechanical and electrical condition upon receipt.

ESD Precautions

To ensure a successful installation and setup it is important that you follow the standard ESD precautions outlined below:

CAUTION:
This is an Electrostatic Sensitive Device. Use ESD precautions for safe handling.

Before removing the card from the anti-static protective packaging:

- *Discharge any static electricity buildup on your body by touching a large grounded metal surface or the metal chassis on equipment connected to earth ground by a 3-wire power cord. Use of a grounding wrist strap is recommended.*
- *Avoid touching the gold connectors or other parts on the card except when necessary to set the configuration DIP switches.*
- *Remove AC power from the computer before inserting the card*

Initial Configuration

The ports in your MIport card are normally pre-configured for RS-232 operation. To ensure the card is configured correctly for your desired operating mode, you will have to check and/or set the three operating mode DIP switches on the card. If you are configuring for RS-485 Mode you also may have to set up the RTS Control parameter in the device driver.

Note: *Refer to Chapter 4 for information on Setting Driver Options.*

Overview of Operational Modes

RS-232 Mode

In RS-232 Mode MIport serial ports function as buffered standard PC serial ports and operate as DTEs (Data Terminal Equipment). RS-232 interfaces are commonly used for communications with modems, serial printers, and computer-controlled devices such as security equipment, bar code scanners and point-of-sale devices.

For most MIport models, RS-232 Mode supports eight single-ended signal lines and signal ground (GND) including transmit (TD), receive (RD) and six hardware handshake lines (DTR, DSR, RTS, CTS, DCD, RI). The only exception to this is the MIport Model 3PCIOU4 card, which supports TD, RD, RTS, CTS and GND.

RS-422 Mode

In RS-422 mode MIport serial ports provide two sets of differential signal pairs (TD and RD) and signal ground for each port. The RS-422 standard uses balanced differential drivers and receivers for each signal. This facilitates greater communication distances than unbalanced systems such as RS-232. In RS-422 mode the transmitter and receiver are always enabled (TX ON, RX ON).

RS-422 operation is suitable for interconnecting a computer and one device for full duplex (point-to-point) bi-directional communication, or a computer and several devices for unidirectional (point-to-multipoints) communication. RS-422 interfaces are commonly used for video editing/control, camera control, electronic signage, television studio/satellite dish control, performance lighting and audio equipment control.

RS-485 Mode

In RS-485 Mode MIport cards provide RS-485 interfaces which operate with the same signals and signal levels as RS-422. RS-485 interfaces differ from RS-422 in that they allow multiple devices to share the same communication link using half duplex (2-wire) or full duplex (4-wire) connections. Since it is possible to have more than one transmitter connected to the media, transmitters must be enabled only while sending data, and tri-stated at all other times so other devices can use the wire pair. MIport cards automatically enable the transmitter at the appropriate time using Automatic Send Data Control, based on the contents of the output buffer. When the buffer has data to send, the transmitter is enabled (TX SD). When all data in the buffer has been sent, the transmitter is disabled and tri-stated to a high impedance state.

In half-duplex operation, the receiver is disabled during transmit (RX $\overline{\text{SD}}$), and enabled when not transmitting. In full-duplex operation the receiver is always enabled (RX ON). Since RS-485 transmitters are tri-stated when not transmitting, the receive inputs must be biased to ensure the media floats in the Mark state so that the first Space state is detected correctly at the start of the next transmission.

These serial cards incorporate the necessary biasing to accommodate up to 32 standard nodes. (Typical input resistance (Rin) for each load is 12kΩ). Provisions are made for custom biasing and/or termination.

Note: *For more information on RS-485 Mode refer to Chapter 6*

Operating Mode Selection

The hardware address and IRQ for the serial card is set by the Windows Operating System using driver information files and the Plug and Play OS.

The Operating Mode is set using DIP switches, Device Manager Driver Settings and by your cable connections and software. Each port on a MIport card has an associated DIP switch to set its operating mode. The port number associated with the DIP switch is clearly silk screened on the printed circuit board.

Note: *Refer to Appendix x for DIP switch locations on various MIport cards.*

Setting the DIP Switches on RS-232/422/485 Ports

Set the DIP switches to configure the desired operating mode as follows:

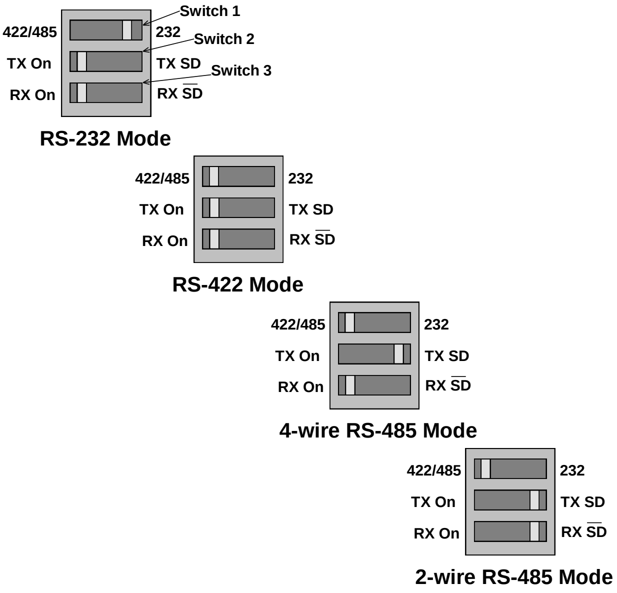


Figure 5. RS-232/422/485 DIP Switch Settings.

DIP Switch 1 (RS-232/422/485 ports)

The top DIP switch (1) configures the port for RS-232 or RS-422/485 operation. This switch is the only one that is required to be set for RS-232 operation. The positions of switches 2 and 3 do not matter when switch 1 is set for RS-232 operation.

DIP Switch 2 (RS-232/422/485 ports)

The middle DIP switch (2) configures the port for RS-485 or RS-422 operation. For RS-422 operation (which uses two wire pairs and sends point-to-point or point-to-multipoints) the transmitter can be enabled all the time. Placing the middle DIP switch in the TX ON position accomplishes this.

For RS-485 operation the middle DIP switch is placed in the TX SD position. In this position the transmitter is only enabled when data is being sent. The transmitter is tri-stated when not sending data, allowing other transmitters on the communications line to transmit without interference.

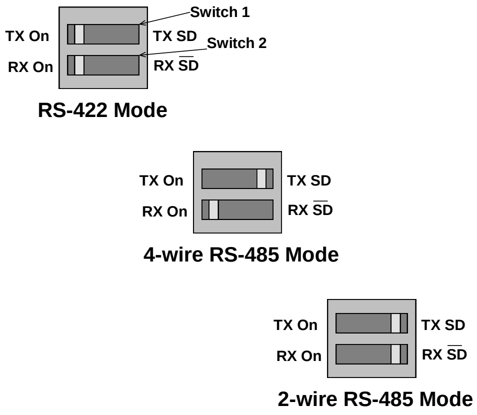
DIP Switch 3 (RS-232/422/485 ports)

The bottom DIP switch (3) configures the port for half-duplex (two-wire) RS-485 operation or full-duplex (four wire) RS-422/RS-485 operation. Placing the bottom DIP switch in the RX ON position configures the port for four wire operation. In this mode the receiver is continuously enabled, allowing it to receive all data on the communications line. Since the transmitter sends data on the other wire pair the port does not receive its own transmissions.

Placing the bottom DIP switch in the RX $\overline{SD}$ position configures the port for two wire operation. In this mode the transmitter and receiver are connected to the same wire pair. The receiver is disabled when its transmitter is sending, preventing the port from receiving its own data.

Setting the DIP Switches on RS-422/485 Only Ports

The 3PCIOU4 MIport card provides a combination of RS-232/422/485 and RS-422/485 only ports. Ports that do not include RS-232 operation use double DIP switches rather than triple DIP switches. These DIP switches operate the same as the two bottom DIP switches in the RS232/422/485 ports.



RS-422 Mode

4-wire RS-485 Mode

2-wire RS-485 Mode

Figure 6. RS-422/485 only DIP Switch Settings

DIP Switch 1 (RS-422/485 only)

The top DIP switch (1) configures the port for RS-485 or RS-422 operation. For RS-422 operation (which uses two wire pairs and sends point-to-point or point-to-multipoints) the transmitter can be enabled all the time. Placing the middle DIP switch in the TX ON position accomplishes this.

For RS-485 operation the middle DIP switch is placed in the TX SD position. In this position the transmitter is only enabled when data is being sent. The transmitter is tri-stated when not sending data, allowing other transmitters on the communications line to transmit without interference.

DIP Switch 2 (RS-422/485 only)

The bottom DIP switch (2) configures the port for half-duplex (two-wire) RS-485 operation or full-duplex (four wire) RS-422/RS-485 operation. Placing the bottom DIP switch in the RX ON position configures the port for four wire operation. In this mode the port’s receiver is continuously enabled, allowing it to receive all data on the communications line. Since the port’s transmitter sends data on the other wire pair the port does not receive its own transmissions.

Placing the bottom DIP switch in the RX $\overline{SD}$ position configures the port for two wire operation. In this mode the port’s transmitter and receiver are connected to the same wire pair. The receiver is disabled when its transmitter is sending, preventing the port from receiving its own data.

Installing the Serial Card

CAUTION:
This is an Electrostatic Sensitive Device. Use ESD precautions for safe handling.

Before removing the card from the anti-static protective packaging:

- *Discharge any static electricity buildup on your body by touching a large grounded metal surface or the metal chassis on equipment connected to earth ground by a 3-wire power cord. Use of a grounding wrist strap is recommended.*
- *Avoid touching the gold connectors or other parts on the card except when necessary to set the configuration DIP switches.*
- *Remove AC power from the computer before inserting the card*

1. Shut down your computer.
2. Unplug the power cord to remove power to prevent accidentally turning on the computer during installation.
3. Remove the cover of the computer.
4. Locate an empty PCI expansion slot.
5. Remove the expansion slot cover. Save the retaining screw.
6. Ground yourself to the computer chassis before and while inserting the card.
7. Install the card into the unused slot. Be certain that the card is inserted completely (fully seated) in the slot.
8. Secure the card with the mounting screw from Step 5.
9. Replace the cover; plug in the power cord.
10. Connect your cables.
11. Power up the system.
12. Install the drivers as described in Chapter 3.

Chapter 3: Driver Software Installation

Installing Windows 7 Driver Software

Windows 7 does not have a “Found New Hardware Wizard.” Therefore the MIPort Driver must be installed manually.

1. Install the MIPort Card and place the CD in the CD ROM Drive
2. Open Device Manager
3. PCI Serial Port will be listed under “Other Devices.”
4. Select PCI Serial Port, right click, and select “Update Driver.”
5. Select “brows my computer” and locate the drivers on the CD ROM.
6. A windows security notice will be displayed. Select “install.” This notice will be displayed again. Select “install.”
7. “Windows successfully updated driver software” will be displayed. Select close.
8. In device manager, the MIPort Card will be listed under Multiport Serial Adapter. Additional COM Ports will be listed under “Ports (COM & LPT).”

Installing Windows XP Driver Software

Installation of the MIport driver software on Windows XP is a three-step process:

1. Windows XP searches and identifies new hardware that has been installed.
2. You use the Found New Hardware Wizard to install the driver software for the card.
3. You use the Found New Hardware Wizard to install the software for each port on the card.

There are several possible methods for installing the software. The procedure outlined here is recommended for most situations.

Note: *If at some point in the future, you want to update these drivers, remove the old drivers before installing the new version. Refer to Chapter 6 for driver removal procedures.*

Pre-Installation Steps

1. Configure the port(s) on the card for the desired mode (RS-232, RS-422 or RS-485) using the three DIP switches on the card.
2. If configuring for RS-422 or RS-485 Modes, and bias or termination resistors are needed, add them at this time.

> ***Note:*** *Refer to Chapter 2 of information on DIP switch settings and bias/termination resistors.*

3. Install the card in the slot. Use appropriate ESD handling precautions.
4. Power up the computer
5. Insert your driver disc in the CD-ROM drive.

Using the Found New Hardware Wizard

Windows will detect the PCI card and start the **Found New Hardware Wizard** to begin the driver installation. Driver software (on CD-ROM) is provided with your MIport card. Do not connect to Windows Update to search for software.

6. Select **No, not at this time** and click **Next.**

PCI Card Software Installation

Once the new hardware has been detected, the wizard will proceed to install the software for the card. The following dialog box will appear:

7. To begin the installation of the software for the PCI card, click **Install the software automatically.** Click **Next.**

   Windows will find the appropriate files on the CD, then display a dialog box concerning Window Logo testing for Vista. This feature of Vista simply indicates that these drivers have not yet undergone the Microsoft testing procedure required to use the Windows Vista Logo on the packaging. Diver compatibility is not affected.

8. Click **Continue Anyway.**

   A dialog box will appear indicating the software installation is proceeding.

9. When the **Completing the Found New Hardware Wizard** dialog appears, click **Finish.**

Port Driver Installation

The Welcome to the Found New Hardware Wizard will appear again, indicating it has detected a port on the PCI card. Repeat the steps above to install the port driver software.

If the cared you are installing has more than one port, Windows Vista will find each port in sequence and re-launch the **Found New Hardware Wizard** for each port. Repeat the previous steps for each port.

Checking the Driver Installation

You may want to check to verify that the new B&B COM ports are now available.

1. From the Widows Desktop, click **Start → Control Panel →System and Maintenance → Device Manager**
2. In the **Device Manager**, click **Multi-port serial adapters.** All serial adapter cards should appear in the list. Additional information about the cards can be obtained by double-clicking the name of the card.
3. Click **Ports (COM & LPT)** All installed ports should appear in the list. The COM port number assigned to each port will be shown.

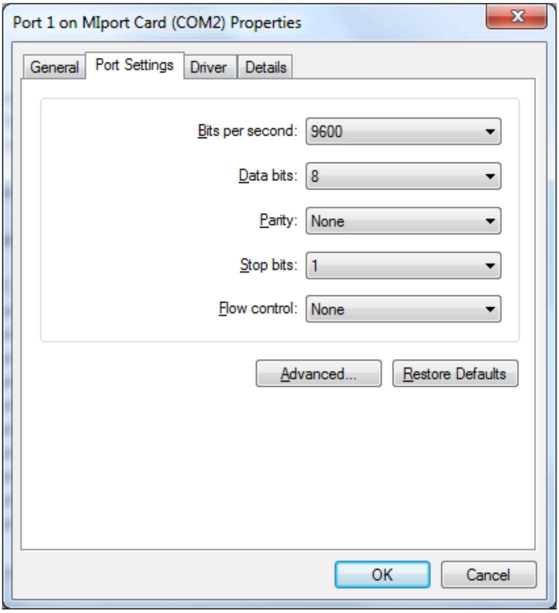
Chapter 4: Setting Driver Options

Configuring Port Settings

By entering the Properties dialog a variety of information can be obtained and several port parameters can be configured.

1. On the **Ports (COM & LPT)** list, double-click the name of the port to be configured.
2. On the **Port Properties** dialog, click the Port **Settings** tab.

   The dialog will display the current settings for Bits per second, Data bits, Parity, Stop bits and Flow control. If necessary, change these settings to the required values.



Port 1 on MIport Card (COM2) Properties

Bits per second: 9600

Data bits: 8

Parity: None

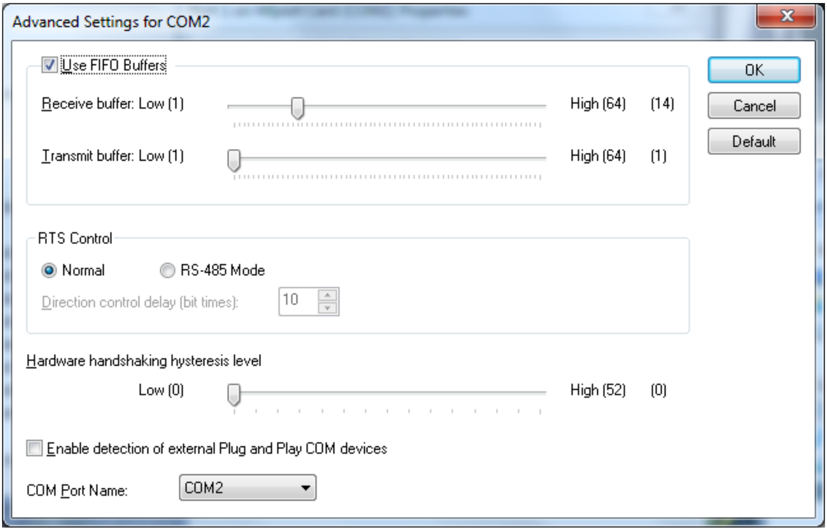
Stop bits: 1

Flow control: None

Figure 7. The Port Settings Dialog

3. Click **Advanced…**

   The **Advanced Port Settings** dialog will appear. This screen allows you to set the Receive and Transmit FIFO buffer thresholds, RTS Control parameters, Hardware handshaking hysteresis level and the COM port name.



Advanced Settings for COM2

☑ Use FIFO Buffers

Receive buffer: Low (1) High (64) (14)

Transmit buffer: Low (1) High (64) (1)

RTS Control

● Normal ○ RS-485 Mode

Direction control delay (bit times): 10

Hardware handshaking hysteresis level

Low (0) High (52) (0)

□ Enable detection of external Plug and Play COM devices

COM Port Name: COM2

Figure 8. Advanced Port Settings

Setting the FIFO Buffers

MIport cards use UARTs that contain 64-byte transmit and receive FIFO (first in, first out) buffer registers. The transmit buffer holds the data to be sent; the receive buffer holds the data received.

The **Transmit buffer: Low** parameter in the **Advanced Port Settings** dialog allows you to set the minimum value at which the UART will request more data from the computer. As the MIport card is transmitting data, the number of bytes left in the transmit buffer will decrease. When the buffer reaches the value set, the UART will request more data from program memory. This is intended to optimize the throughput of the data. Typically this parameter is set at zero and usually does not have to be changed.

The **Receive buffer: Low** parameter in the Advance Port Settings dialog allows you to set the value at which the UART will request that the computer read the data that has been received. As the MIport card is receiving data the contents of the receive buffer increases, when it reaches the value set, the UART will request that the computer transfer the contents of the buffer to program memory. This is intended to optimize the throughput of the data. The default value is 14. Usually this parameter does not have to be changed.

Setting the RTS Control Parameter

1. Set **RTS Control** to **Normal** for RS-232; set **RTS Control** to **RS-485 Mode** for 2-wire or 4-wire RS-485 operation. Either setting can be used for RS-422.

Typically the COM parameters set in this dialog box are overridden by the software application.

Note: *For proper operation the DIP switches on the PCI card must be set for the same mode. Refer to Chapter 2 for more information on configuring the DIP switches.*

2. The **Direction control delay (bit times)** box allows you to set the length of time the Transmit driver continues to be enabled after the last bit of data in the transmit register has been sent. The purpose of this setting is to maintain the transmission line in a know (idle) state until the data has reached its destination. If the line is released too soon it can pick up noise that could cause problems on the communications system. This parameter is preset to a value of 10 bit times and typically will not have to be changed. For longer transmission lines this value may have to be increased.

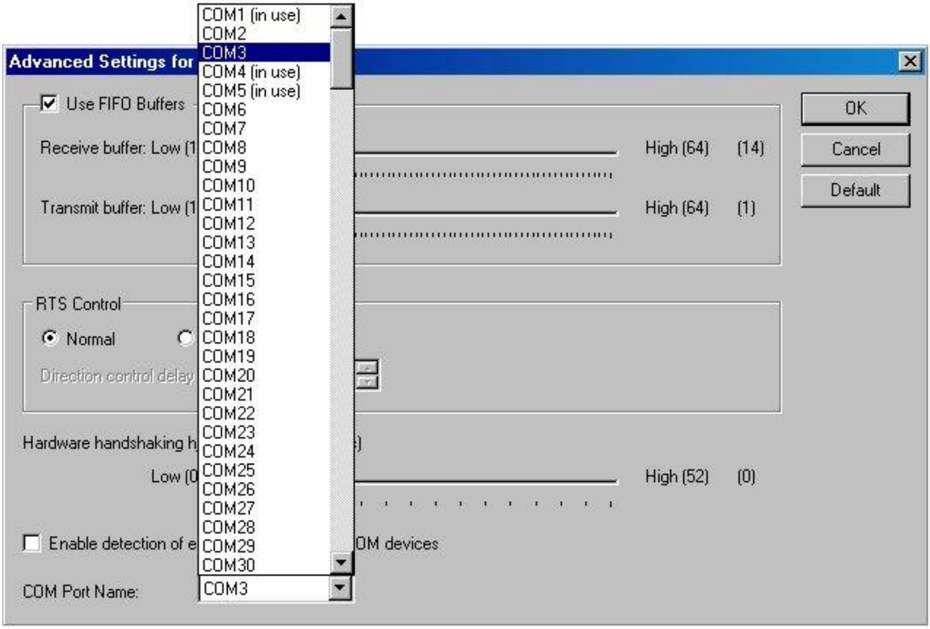
Setting the Hardware Handshaking Level

The **Hardware handshaking Hysteresis level (characters)** parameter allows you to set how long the local UART will continue to accept data after its receive buffer has reached the point at which it asks the computer to transfer the data to program memory. When this point is reached the UART de-asserts the RTS hardware handshake line (which is connected to the remote UART's CTS line), stopping the remote device from sending data until its CTS line is re-asserted. This will occur when the local receive FIFO buffer has been cleared. The default value for this parameter is 0. Typically this value will not need to be changed.

Changing the COM Port Name/Number

1. To change the COM port number assigned to the port, click the **COM Port Name** arrow and select the new **COM Port Name** from the list.

   Available names for COM numbers are shown. Select a new number from those not "in use". COM numbers from COM1 to COM256 may be available. COM numbers "in use" may be used by motherboard ports, modems, virtual COM ports for network serial server devices or FAX modems. Formerly installed USB to serial adaptors, PCI cards or other hardware may have reserved a COM number. The device may need to be added back to the system, then the software removed. Advanced editing of the registry may required to clean up the problem. Special permissions are required with 2000 or XP.



Advanced Settings for

☑ Use FIFO Buffers

Receive buffer: Low (1 High (64) (14)

Transmit buffer: Low (1 High (64) (1)

RTS Control

● Normal ○

Direction control delay

Hardware handshaking h)

Low (0 High (52) (0)

□ Enable detection of e OM devices

COM Port Name: COM3

Figure 9. Changing the COM Port Name/Number

2. After selecting a new COM name/number, Click **OK**, then click **OK** again on the **Settings** page.

After returning to the **Device Manager** screen, it should refresh automatically. The name/number of the port in the Device Manager list should now show the change.

Chapter 5: Installing Linux Driver Software

Installing MIport Driver on Ubuntu 10.04

Ubuntu 10.04 Desktop (32-bit and 64-bit)

1. Install the MIport serial card in the computer.

2. Open a terminal window. This can be found on the desktop menu under Application | Accessories | Terminal.

D-3. **[optional]** Start recording everything entered into, and output to, the terminal. This step is only required if there is a problem installing the driver.

```
username@computer$ script -t 2> ~/MIport.time -a ~/MIport.script ↲
```

```
Script started on Tue 31 Aug 2010 03:56:11 AM CDT
```

4. Extract the archive that stores the MIport driver.

```
username@computer$ tar -zxvf /media/MIport\
3.0/linux/drivers/2.6/MIport/MIport-3.0.0.tar.gz ↲
```

```
MIport-3.0.0/
MIport-3.0.0/rc.local
MIport-3.0.0/bbmknod.sh
MIport-3.0.0/README
MIport-3.0.0/README.pdf
MIport-3.0.0/Makefile
MIport-3.0.0/MIport.c
Miport-3.0.0/rc.MIport
```

5. Change the current working directory to where the MIport driver source code is located.

```
username@computer$ cd MIport-3.0.0 ↲
```

6. Compile the MIport device driver.

```
username@computer$ make ↲
```

```
make -C /lib/modules/2.6.32-24-generic/build M=/home/username/MIport-
3.0.0 modules
make[1]: Entering directory `/usr/src/linux-headers-2.6.32-24-generic'
  CC [M]  /home/username/MIport-3.0.0/MIport.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/username/MIport-3.0.0/MIport.mod.o
  LD [M]  /home/username/MIport-3.0.0/MIport.ko
make[1]: Leaving directory `/usr/src/linux-headers-2.6.32-24-generic'
```

7. Determine the present working directory. The output may be different than shown below.

```
username@computer$ pwd ↲
```

```
/home/username/MIport-3.0.0
```

8. Install the MIport device driver. The part after “PWD=” of the command below should match the output from the command above.

```
username@computer$ sudo make install PWD=/home/username/MIport-3.0.0 ↲
```

```
make -C /lib/modules/2.6.32-24-generic/build M=/home/username/MIport-
3.0.0 modules_install
make[1]: Entering directory `/usr/src/linux-headers-2.6.32-24-generic'
  INSTALL /home/username/MIport-3.0.0/MIport.ko
  DEPMOD  2.6.32-24-generic
make[1]: Leaving directory `/usr/src/linux-headers-2.6.32-24-generic'
cp ./rc.MIport /etc/rc.MIport
chmod 755 /etc/rc.MIport
```

9. Start the MIport driver.

```
username@computer$ sudo /etc/rc.MIport ↲
```

```
Searching for MIport driver in: ...
The MIport driver was found in: ...
The MIport driver is loaded.
Making special nodes for MIport.

Making /dev/ttyM0 /dev/cum0
Making /dev/ttyM1 /dev/cum1
Making /dev/ttyM2 /dev/cum2
Making /dev/ttyM3 /dev/cum3
Making /dev/ttyM4 /dev/cum4
Making /dev/ttyM5 /dev/cum5
Making /dev/ttyM6 /dev/cum6
Making /dev/ttyM7 /dev/cum7
```

10. Verify that the device driver is running. If the output is similar to what is shown below, then the MIport driver is running. If the output is blank, then the MIport driver is not running.

```
username@computer$ dmesg | grep MIport ↲
```

```
[  271.828182] B&B Electronics PCI MIport serial driver Revision: 4.5
[  314.159265] MIport 0000:07:05:05.0: PCI INT A -> GSI 26 (level, low)
-> IRQ 26
```

11. **[optional]** If recording was started in Step 2, stop recording. The file ~/MIport.script contains all output recorded from the terminal.

```
username@computer$ exit ↲
```

```
exit

Script done on Tue 31 Aug 2010 03:58:48 AM CDT
```

12. Test serial communications using gtkterm. Change the serial port under Configration | Ports and set the port to /dev/ttyM0 (for each serial port, increment the number at the end of the name; e.g., /dev/ttyM1).

```
username@computer$ gtkterm & ↲
```

13. If the driver works correctly, modify /etc/rc.local to automatically load the driver when the operating system starts.

```
username@computer$ sudo gedit /etc/rc.local
```

Add the following code to the file (before the "exit 0" statement):

```
if [ -f /etc/rc.MIport ] ; then
   exec /etc/rc.MIport
fi
```

An example of the code to add to /etc/rc.local is in the MIport driver folder in the file rc.local.

14. Restart the computer, and then test serial communications again with gtkterm.

Ubuntu 10.04 Server (32-bit and 64-bit)

1. Determine which version of GCC built the current Linux kernel.

```
username@computer$ cat /proc/version ↲
```

```
Linux version 2.6.32-21-server (buildd@yellow) (gcc version 4.4.3
(Ubuntu 4.4.3-4ubuntu5) ) #32-Ubuntu SMP Fri Apr 16 09:17:34 UTC 2010
```

The current Linux kernel was built using GCC v4.4.3.

2. Install the version of GCC that built the current Linux kernel.

```
username@computer$ sudo apt-get update ↲
```

```
:
:
Fetched 12.0MB in 50s (237kB/s)
Reading package lists... Done
```

```
username@computer$ sudo apt-get install gcc-4.4 gcc ↲
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  binutils gcc-4.4 libc-bin libc-dev-bin libc6 libc6-dev libgomp1 linux-
libc-dev manpages-dev
:
:
After this operation, 35.3MB of additional disk space will be used.
Do you want to continue [Y/n]? y ↲
:
:
Setting up binutils (2.20.1-3ubuntu7) ...
Setting up libgomp1 (4.4.3-4ubuntu5) ...
Setting up gcc-4.4 (4.4.3-4ubuntu5) ...
Setting up gcc (4:$.4.3-1ubuntu1) ...
Setting up libcdev-bin (2.11.1-0ubuntu7.5) ...
Setting up linux-libc-dev (2.6.32-25.45) ...
Setting up libc6-dev (2.11.1-0ubuntu7.5) ...
```

3. Get the version of the Linux kernel.

```
username@computer$ uname -r ↲
```

```
2.6.32-21-server
```

4. Install the Linux kernel headers for the current Linux kernel. In the command below, the part after “linux-headers-” should match the output of the command above.

```
username@computer$ sudo apt-get install linux-headers-2.6.32-21-server ↲
```

```
:
:
Setting up linux-headers-2.6.32-25 (2.6.32-25.45)
Setting up linux-headers-2.6.32-25-server (2.6.32-25.45)
```

5. Create a folder for mounting the CD-ROM.

```
username@computer$ sudo mkdir -p /media/cdrom ↲
```

6. Check that the folder was created for mounting the CD-ROM.

```
username@computer$ ls /media ↲
```

```
cdrom  floppy  floppy0
```

If cdrom is not listed above, then the folder was not create correctly.

7. Insert the CD-ROM into the CD-ROM drive and mount the CD-ROM.

```
username@computer$ sudo mount -t iso9660 -r /dev/cdrom /media/cdrom ↲
```

8. Check that the CD-ROM was mounted.

```
username@computer$ ls /media/cdrom ↲
```

```
autorun.inf  linux  manuals  MIport.ico  windows
```

If the files listed above are not shown, then the CD-ROM was not mounted.

9. Extract the archive that stores the MIport driver.

```
username@computer$ tar -zxvf
/media/cdrom/linux/drivers/2.6/MIport/MIport-3.0.0.tar.gz ↲
```

```
MIport-3.0.0/
MIport-3.0.0/rc.local
MIport-3.0.0/bbmknod.sh
MIport-3.0.0/README
MIport-3.0.0/README.pdf
MIport-3.0.0/Makefile
MIport-3.0.0/MIport.c
Miport-3.0.0/rc.MIport
```

10. Unmount the CD-ROM.

```
username@computer$ sudo umount /media/cdrom ↲
```

11. Change the current working directory to where the MIport driver source code is located.

```
username@computer$ cd MIport-3.0.0 ↲
```

12. Compile the MIport device driver.

```
username@computer$ make ↲
```

```
make -C /lib/modules/2.6.32-24-generic/build M=/home/username/MIport-
3.0.0 modules
make[1]: Entering directory `/usr/src/linux-headers-2.6.32-24-generic'
  CC [M]  /home/username/MIport-3.0.0/MIport.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/username/MIport-3.0.0/MIport.mod.o
  LD [M]  /home/username/MIport-3.0.0/MIport.ko
make[1]: Leaving directory `/usr/src/linux-headers-2.6.32-24-generic'
```

13. Determine the present working directory. The output may be different than shown below.

```
username@computer$ pwd ↲
```

```
/home/username/MIport-3.0.0
```

14. Install the MIport device driver. The part after “PWD=” of the command below should match the output from the command above.

```
username@computer$ sudo make install PWD=/home/username/MIport-3.0.0 ↲
```

```
make -C /lib/modules/2.6.32-24-generic/build M=/home/username/MIport-
3.0.0 modules_install
make[1]: Entering directory `/usr/src/linux-headers-2.6.32-24-generic'
  INSTALL /home/username/MIport-3.0.0/MIport.ko
  DEPMOD  2.6.32-24-generic
make[1]: Leaving directory `/usr/src/linux-headers-2.6.32-24-generic'
cp ./rc.MIport /etc/rc.MIport
chmod 755 /etc/rc.MIport
```

15. Start the MIport driver.

```
username@computer$ sudo /etc/rc.MIport ↲
```

```
Searching for MIport driver in: ...
The MIport driver was found in: ...
The MIport driver is loaded.
Making special nodes for MIport.

Making /dev/ttyM0 /dev/cum0
Making /dev/ttyM1 /dev/cum1
Making /dev/ttyM2 /dev/cum2
Making /dev/ttyM3 /dev/cum3
Making /dev/ttyM4 /dev/cum4
Making /dev/ttyM5 /dev/cum5
Making /dev/ttyM6 /dev/cum6
Making /dev/ttyM7 /dev/cum7
```

16. Verify that the device driver is running. If the output is similar to what is shown below, then the MIport driver is running. If the output is blank, then the MIport driver is not running.

```
username@computer$ dmesg | grep MIport ↲
```

```
[  271.828182] B&B Electronics PCI MIport serial driver Revision: 4.5
[  314.159265] MIport 0000:07:05:05.0: PCI INT A -> GSI 26 (level, low)
-> IRQ 26
```

17. Use a serial communications program to verify that the driver works. The serial ports will be “/dev/ttyM0” through “/dev/ttyM7”.

18. If the driver works correctly, modify /etc/rc.local to automatically load the driver when the operating system starts.

```
username@computer$ sudo vim /etc/rc.local ↲
```

Add the following code to the file (before the "exit 0" statement):

```
     if [ -f /etc/rc.MIport ] ; then
         exec /etc/rc.MIport
     fi
```

An example of the code to add to /etc/rc.local is in the MIport driver folder in the file rc.local.

19. Restart the computer, and then test serial communications again.

Chapter 6: Removing Drivers, Ports and Cards

Windows 7

Uninstalling the MIport Card

1. In the **Device Manager**, under **Multi-port serial adapters**, right-click the card to be uninstalled. **NOTE:** The card must still be physically installed in the computer, otherwise it will not show up in the Device Manager.
2. In the pop-up menu that appears, click **Uninstall**.
3. On the **Confirm Device Removal** dialog that appears, click **OK**.
4. After a few seconds the dialog will disappear. In the **Device Manager** the listings for the card (under **Multi-port serial adapters)** and all associated COM ports (under **Ports (COM & LPT)**) will be gone.
5. Physically remove the MIport card from the computer.

Uninstalling the COM ports

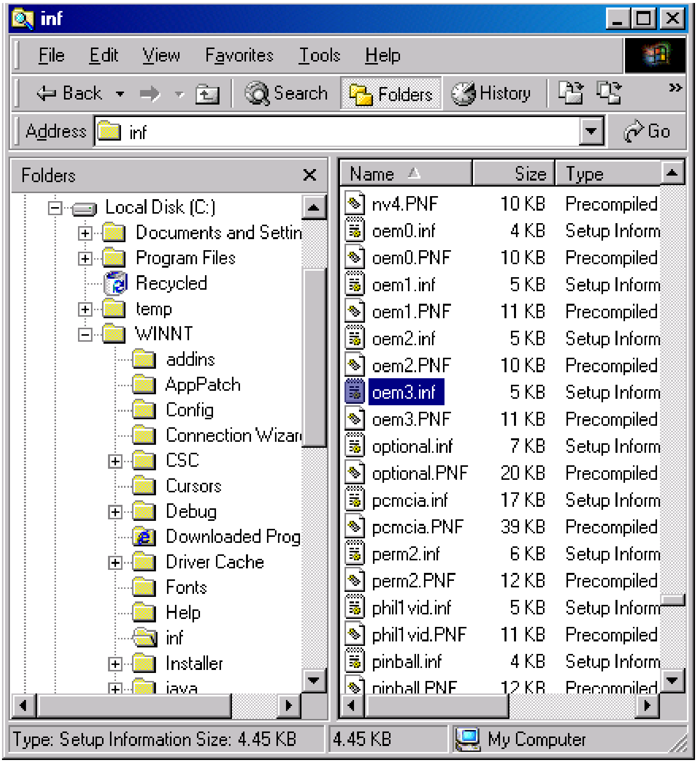
1. In the **Device Manager**, under **Ports (COM & LPT),** right-click the COM port to be uninstalled**.**
2. On the **Confirm Device Removal** dialog that appears, click **OK**.
3. After a few seconds the dialog will disappear. In the **Device Manager** the listings for the port (under **Ports (COM & LPT)**) will be gone.

This procedure will not affect other ports or the card itself. To uninstall the card and all its ports use the preceding procedure.

Removing INF and PNF Driver Files

1. Open Windows Explorer as follows: From the **Windows Desktop**, click **Programs → Accessories → Windows Explorer**
2. Under the Windows directory expand the inf sub-directory and find the oem*X*.inf and oem*X*.PNF files (where *X* represents the number of the file).

The operating system names these files during the installation process. To locate the correct INF files, search for files in **C:\Windows\Inf** and its subdirectories for a file named ***.INF** that contains the text “**Disk #6404**”. There will be two files, unless the installation failed, then there might only be one file. Then, the customer can double-check that it is in fact a B&B Electronics file.



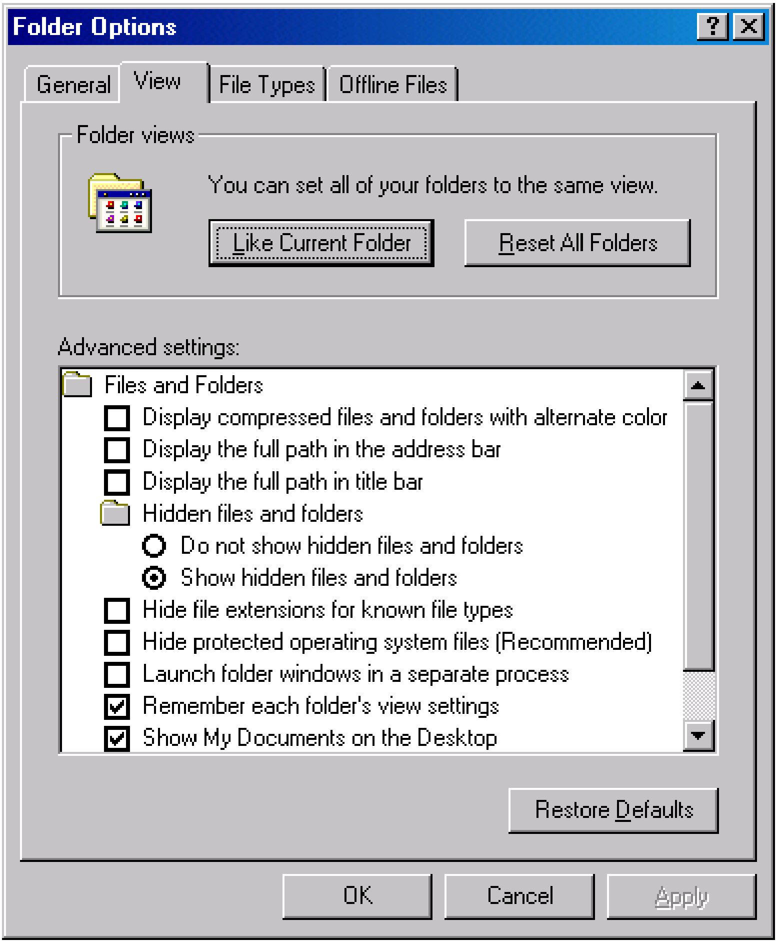
- Local Disk (C:)
  - Documents and Settin
  - Program Files
  - Recycled
  - temp
  - WINNT
    - addins
    - AppPatch
    - Config
    - Connection Wizar
    - CSC
    - Cursors
    - Debug
    - Downloaded Prog
    - Driver Cache
    - Fonts
    - Help
    - inf
    - Installer
    - java

Name	Size	Type
nv4.PNF	10 KB	Precompiled
oem0.inf	4 KB	Setup Inform
oem0.PNF	10 KB	Precompiled
oem1.inf	5 KB	Setup Inform
oem1.PNF	11 KB	Precompiled
oem2.inf	5 KB	Setup Inform
oem2.PNF	10 KB	Precompiled
oem3.inf	5 KB	Setup Inform
oem3.PNF	11 KB	Precompiled
optional.inf	7 KB	Setup Inform
optional.PNF	20 KB	Precompiled
pcmcia.inf	17 KB	Setup Inform
pcmcia.PNF	39 KB	Precompiled
perm2.inf	6 KB	Setup Inform
perm2.PNF	12 KB	Precompiled
phil1vid.inf	5 KB	Setup Inform
phil1vid.PNF	11 KB	Precompiled
pinball.inf	4 KB	Setup Inform
pinball.PNF	12 KB	Precompiled

Type: Setup Information Size: 4.45 KB | 4.45 KB | My Computer

Figure 10. *Finding the INF files in Windows Explorer*

To find these files you may need to set your Views (under My Computer) to show all files and folders if the INF directory and .inf files are not visible.



You can set all of your folders to the same view.

Advanced settings:

- Files and Folders
  - □ Display compressed files and folders with alternate color
  - □ Display the full path in the address bar
  - □ Display the full path in title bar
  - Hidden files and folders
    - ○ Do not show hidden files and folders
    - ● Show hidden files and folders
  - □ Hide file extensions for known file types
  - □ Hide protected operating system files (Recommended)
  - □ Launch folder windows in a separate process
  - ☑ Remember each folder's view settings
  - ☑ Show My Documents on the Desktop

Figure 11. Setting the Folder Options to Display Hidden Files

3. Delete the oemx.inf and oemx.pnf files found in Step 2..

CAUTION! Be careful to delete only the files associated with the PCI card you are trying to uninstall.

Chapter 7: RS-232 Connections/Operation

RS-232 Mode

In RS-232 Mode MIport serial ports function as buffered standard PC serial ports and operate as DTEs (Data Terminal Equipment). RS-232 interfaces are commonly used for communications with modems, serial printers, and computer-controlled devices such as security equipment, bar code scanners and point-of-sale devices.

RS-232 Signal Designations and DB-9 Pinout

RS-232 Mode supports eight single-ended signal lines and signal ground. The DB-9 male connector is configured as a standard RS-232 (DTE) serial port. The table below shows the signal names and pin numbers.

RS-232 Signal Name	**RS-232 Signal Description**	**Direction (DTE)**	**DB-9M Pin**
DCD	Data Carrier Detect	Input	1
RD	Receive Data	Input	2
TD	Transmit Data	Output	3
DTR	Data Terminal Ready	Output	4
GND	Signal Ground	------	5
DSR	Data Set Ready	Input	6
RTS	Request to Send	Output	7
CTS	Clear to Send	Input	8
RI	Ring Indicator	Input	9

Figure 12. RS-232 Signal Designations and DB-9 Pinout

Note: Refer to Appendix B for additional cable configurations and pinouts

RS-232 Signal Designations

The primary RS-232 signals are **TD** (transmit) and **RD** (receive). Together with GND (ground), they often are referred to as a “3-wire” interface.

The **RTS** (Request to Send) and **CTS** (Clear to Send) signals are handshaking lines used to indicate to the other device that data can be sent or received. These lines may be enabled or disabled on a byte-by-byte basis and are used to prevent buffer overrun or the loss of data.

Two secondary handshaking signals are **DTR** (Data Terminal Ready) and **DSR** (Data Set Ready). They are usually enabled when the device is powered up and the port is open. They may be used for flow control by some devices instead of RTS and CTS.

DCD (Data Carrier Detect) is used by a modem to indicate Carrier to the computer so data can be sent/received.

RI (Ring Indicator) is output by a modem to indicate that the phone or FAX line has a incoming call, so it can be handled.

DTE and DCE

There are two types of RS-232 ports:

- **DTE** (Data Terminal Equipment) which is typically a computer
- **DCE** (Data Communications Equipment) which is typically a modem.

Data Terminal Equipment (DTE)

When a DTE is connected to a DCE they are linked directly together using a modem cable wired pin to pin so that the inputs match the outputs of the other.

Data Communications Equipment (DCE)

When two pieces of equipment of the same type are interconnected (a DTE connected to a DTE, or a DCE to a DCE), a crossover (often called a null modem) cable is needed to route the outputs of one to the inputs of the other. This type of cable is needed to interconnect two computers with RS-232.

Note: *Refer to Appendix B for straight-through and null modem cable diagrams and pinouts.*

RS-232 Signal Levels

RS-232 signal lines are referenced to ground, and each signal can alternate above and below ground. The RS-232 standard specifies output voltages must be no less than +5 volts and no greater than +25 volts to represent a Space on a transmit line (or an asserted handshake line). Output voltages must be between –5 volts and –25 volts to represent a Mark on a transmit line (or an de-asserted handshake line).

The standard specifies that RS-232 inputs accept voltage levels between –3 volts and –25 volts for a Mark (asserted handshake) and between +3 volts to +25 volts for a Space (de-asserted handshake). Typically RS-232 voltages will be between +3V to +11V and –3V to –11V.

Handshaking

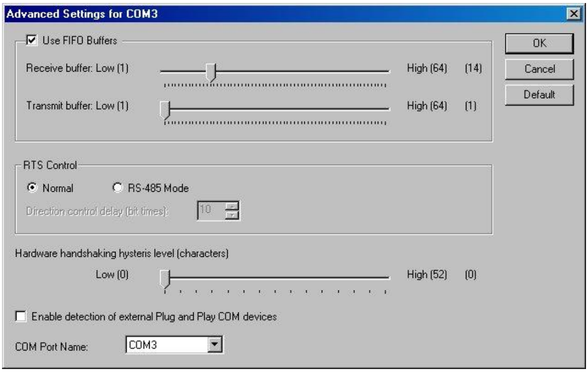
To control the flow of data between two devices, some software uses hardware handshaking. RS-232 devices which require hardware handshaking require connection of RTS/CTS and/or DTR/DSR lines in order to operate properly.

Programs for RS-232 may chose to use only the RS-232 TD and RD data lines and ignore hardware handshaking inputs (set up as None in the port configuration). In some cases the software may require connections to verify that a cable is connected and the devices are ready for access (typically DTR/DSR is used).

Some devices may use software handshaking in which data characters (e.g. Xon/Xoff) are sent to start and stop the incoming or outgoing data. These unprintable characters have the decimal value (17) and (19), and can often be generated in a terminal program with CTL+Q or CTL+S.

RTS Control In RS-232 Mode

In RS-232 mode, MIport cards support software and hardware handshaking. Handshake lines can be controlled from the communications software used with your application. When operating a MIport card port in RS-232 mode, set the **RTS Control** driver setting to **Normal**. This ensures that the RTS line will be free to be controlled by your software. If your communications software uses software handshaking, hardware handshake lines will not be required.



Advanced Settings for COM3

☑ Use FIFO Buffers

Receive buffer: Low (1) High (64) (14)

Transmit buffer: Low (1) High (64) (1)

RTS Control

● Normal ○ RS-485 Mode

Direction control delay (bit times): 10

Hardware handshaking hysteris level (characters)

Low (0) High (52) (0)

□ Enable detection of external Plug and Play COM devices

COM Port Name: COM3

Figure 13. RTS Control – Select Normal for RS-232 Operation

Chapter 8: RS-422/485 Connections/Operation

RS-422/485 Mode

In RS-422/RS-485 mode MIport serial ports provide two sets of differential signal pairs and signal ground for each port. The RS-422 and RS-485 standards use balanced differential drivers and receivers for each signal. This facilitates greater communication distances than unbalanced systems such as RS-232.

RS-422 operation is suitable for interconnecting a computer and one device for full duplex (point-to-point) bi-directional communication, or a computer and several devices for unidirectional (point-to-multipoint) communication. RS-422 interfaces are commonly used for video editing/control, camera control, electronic signage, television studio/satellite dish control, performance lighting and audio equipment control.

RS-485 operates with the same signals and signal levels as RS-422 but differs in that it allows multiple devices to share the same communication link using half duplex (2-wire) or full duplex (4-wire) connections. RS-485 interfaces are commonly used in manufacturing and industrial/commercial control applications such as programmable logic controllers, telemetry, and process control.

RS-422/485 Signal Designations and DB-9 Pinout

Name	Description	Direction	DB-9 M Pin
RDA(-)	Receive Data A (-)	Input	1
TDB(+)	Transmit Data B (+)	Output	2
TDA(-)	Transmit Data A (-)	Output	3
GND	Signal Ground	------	5
RDB(+)	Receive Data B (+)	Input	9

Figure 14. RS-422/485 Signal Designations and DB-9 Pinout

Note: *Refer to Appendix B for additional cable configurations and pinouts*

RS-422/485 Signal Designations

Typically RS-422 and RS-485 interfaces use five lines including two signal pairs and ground. One signal pair is the transmit pair, labeled **TDA(-)** and **TDB(+)**. The other signal pair is the receive pair, labeled **RDA(-)** and **RDB(+)**. Signal ground is labeled **GND**.

RS-422/485 Differential Signals

In RS-422 and RS-485 interfaces signals are sent on differential pairs. In the idle, or Mark, state, the TDA(-) line will be 0V with respect to (wrt) ground and the TDB(+) line will be about 4.4V wrt ground. Sensing differentially across the two lines, the voltage will be –4.4V (representing a Mark state). When the transmitter changes to the Space state TDA(-) goes to +4.4V and TDB(+) goes to 0V. In that state the differential voltage will be +4.4V

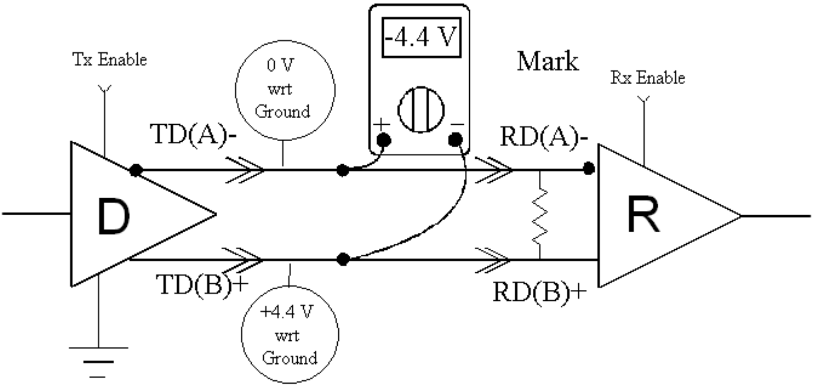


Figure 15. Differential transmitter/receiver

The receiver input is a differential circuit which senses the difference in voltage across the transmission line as described above. In addition, any common mode noise picked up equally on both wires of the twisted pair is cancelled. Receiver sensitivity is rated at 200mV. A separate signal ground/common connection provides a common mode reference for the transmitter and receiver (and is often used to ground a shield in the cable). These factors provide reliable communications at much greater distances than RS-232.

RS-422 Operation

In RS-422 mode, the transmitter is enabled (TX ENABLE) all the time, and the receiver is enabled (RX ENABLE) all the time. Typical point-to-point connections use a transmitter and receiver at each end with two wire pairs connecting them. The transmit lines of the device at one end of the link are connected to the matching receive lines of the device at the other end. The second device transmit lines are connected to the receive lines of the first.

Another common connection mode—point to multipoints—connects the transmit pair from the master device transmitter to the receive lines of many listening slave devices. In this scenario the communications is one-way. None of the listening devices have transmitters connected to reply to the master.

RS-422 Limitations

The limitation of RS-422 operation is that only one transmitter can be connected to a wire pair because the transmitter is active all the time. Even when the transmitter is not sending data it will assume the idle, or Mark, state in which TDA(-) is at 0V with respect to ground and TDB(+) line is at about 4.4V with respect to ground. If another transmitter output is connected to the same wire pair, and attempts to begin sending data by setting the line pair to Space state, the first transmitter will hold the opposite state and neither can communicate. To overcome this limitation, RS-485 mode was developed, in which transmitters connected to the line are put into a high-impedance (tri-state) state when not transmitting.

RS-485 Operation

RS-485 solves some of the limitations of RS-422, allowing multidrop communications using 2-wire and 4-wire connections from multiple transmitters and receivers.

To accomplish multidrop operation the transmit driver must be enabled only during transmit (by asserting the enable input (TX SD) of the transmitter) and tri-stated to a high impedance after the data has been sent. In the 2-wire (half duplex) mode, the receiver is enabled when not transmitting, and disabled (RX $\overline{\text{SD}}$) during transmit (called 'echo off' because it avoids having the device receive its own transmissions).

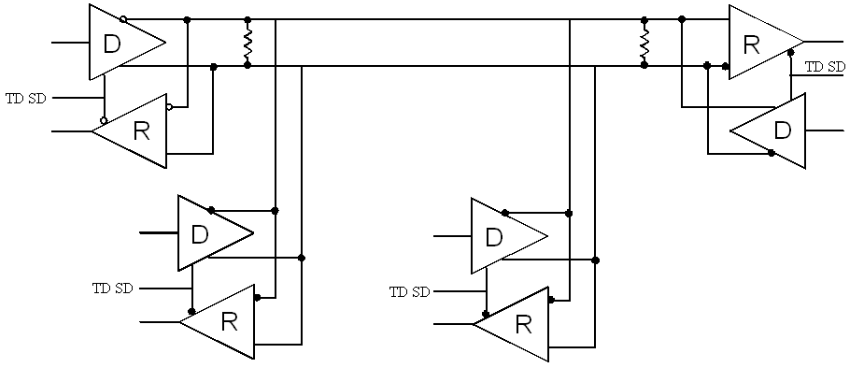
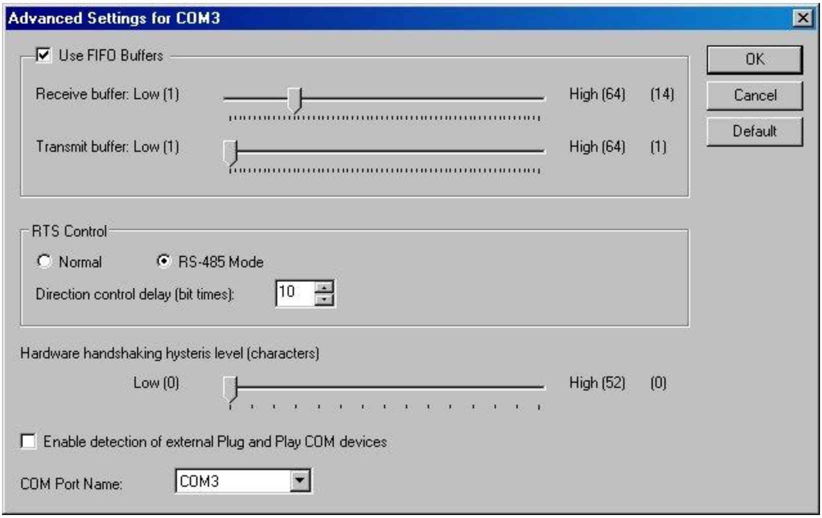


Figure 16. An RS-485 Two-Wire Multidrop Connection

Send Data Control

MIport cards provide Send Data Control (SDC) for the RS-485 driver and receiver. This is hardware controlled based on the contents of the UART output buffer. When data is present, the driver is enabled; when the output buffer becomes empty, it is disabled. This automatically handles whatever baud rate is used.

The RS-485 transmitter and receiver have separate settings for 2-wire modes (TX SD, RX SD) or 4-wire (TX SD, RX ENABLE).



Advanced Settings for COM3

☑ Use FIFO Buffers

Receive buffer: Low (1) High (64) (14)

Transmit buffer: Low (1) High (64) (1)

RTS Control

○ Normal ● RS-485 Mode

Direction control delay (bit times): 10

Hardware handshaking hysteris level (characters)

Low (0) High (52) (0)

□ Enable detection of external Plug and Play COM devices

COM Port Name: COM3

Figure 17. Setting RTS Control for RS-485

The RS-485 mode is set by configuring the DIP switch setting on the MIport card and by selecting **RS-485 Mode** under **RTS Control** in the Advanced COM port settings dialog box.

Note: For more information on COM port settings refer to Chapter 2 and Chapter 5

RS-485 Termination Resistors

In some applications termination resistors must be connected across the communications line to ensure reliable communications. Termination resistors absorb signal reflections that can occur when a data signal reaches the end of a transmission line and encounters an impedance higher than that of the transmission line itself. Typically termination resistors are not required for communications links that operate at less that 19.2 kbps. At higher bit rates a termination resistor of 120 ohms may be required. Termination resistors should be connected across the communications line at the extreme opposite ends of the network.

Through hole pads are provided on MIport cards to install termination resistors. When adding termination resistors the value of biasing resistors must be appropriately sized as well.

RS-485 Network Biasing

Unlike RS-422, where the transmitter holds the TDB(+) line high and the TDA(-) line low (in the idle, or Mark state) when not transmitting data, in a RS-485 network, the transmitter tri-states to a high impedance state when not sending data.

It is important to maintain the Mark state for all RS-485 receivers when no data is being sent. Maintaining the line in the Mark state minimizes noise (that would otherwise be picked up by a high impedance line). It also provides the starting point so that when a transmitter begins to send data the leading edge of the start bit can transition from the Mark to Space state.

To maintain the Mark state biasing is required. Biasing is the technique of placing pull up (to positive voltage) and pull down (to ground) resistors on the line so that all RDA(-) receiver inputs remain at least 200 mV more negative than RDB(+) inputs (Mark state). This ensures the inputs will remain above the receiver threshold of 200mV for all devices on that section of the network. If the biasing is not maintained, the first data bit of a signal may not be detected, and one or more devices may not respond to commands, or may operate intermittently.

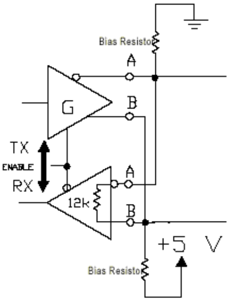


Figure 18. RS-485 Biasing Resistors

MIport RS-485 receivers come pre-biased from the factory with a 4.7 kΩ pull-up resistor on the RDB(+) line and a 4.7 kΩ pull-down resistor on the RDA(-) line. These values are usually adequate for networks that do not implement termination resistors and have 31 or fewer nodes. When termination is used, biasing must be increased (resistors decreased), calculated according to the termination value and number of nodes.

Through hole pads are provided on MIport cards for adding termination resistors and bias resistors. The photograph below shows the layout of through hole pads for installing termination and bias resistors. On all MIport cards the top set of holes is for the termination resistor. The middle and bottom sets of holes are for bias resistors. (Note the factory installed 4.7k resistors in the photograph.) Since bias resistors are always the same value (e.g R pullup = R pulldown = 4.7k) there is no need to identify which set of holes is for pull up and which is for pull down.

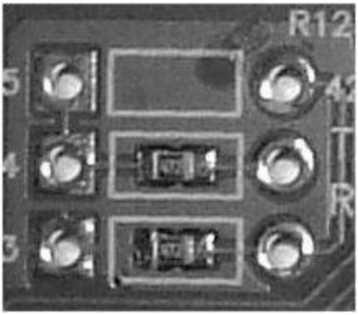
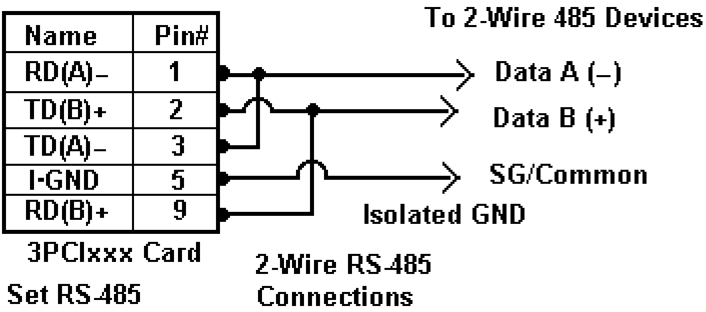


Figure 19. Termination and Bias Resistor Placement

Note: *For more information on termination, biasing and how to calculate resistor values download the RS-422/485 Application Note from www.bb-elec.com*

2-Wire RS-485 Connections

The following diagram shows how to wire the DB-9 connector that will plug into your MIport card for 2-wire RS-485 operation.



Name	Pin#
RD(A)–	1
TD(B)+	2
TD(A)–	3
I-GND	5
RD(B)+	9

Figure 20. 2-Wire RS-485 Connections

2-Wire RS-485 Mode:

Your cables must bridge pins #1 & #3 and pins #2 & #9 in order to receive and transmit. Connect from Pin #2 to Data B(**+**) of your devices and from pin #3 to Data A(-) of your devices. Make sure your DIP switches are set, and that the driver Setting for RTS Control is **RS-485 Mode** in the Device Manager, Port Settings, Advanced.

Note that the EIA RS-422 Specification labels data lines with an "A" and "B" designator. Some RS-422 or RS-485 equipment uses a "–" and "+" designator. In most cases, the "A" line is the equivalent of the "–" line and the "B" line is the equivalent of the "+" line. Some device manufacturers may not follow the standard designation for RS-422 or RS-

485, using the A connection for "+" and the B for "-". In such cases, reversing the line pair permits operation.

4-Wire RS-422 and RS-485 Connections

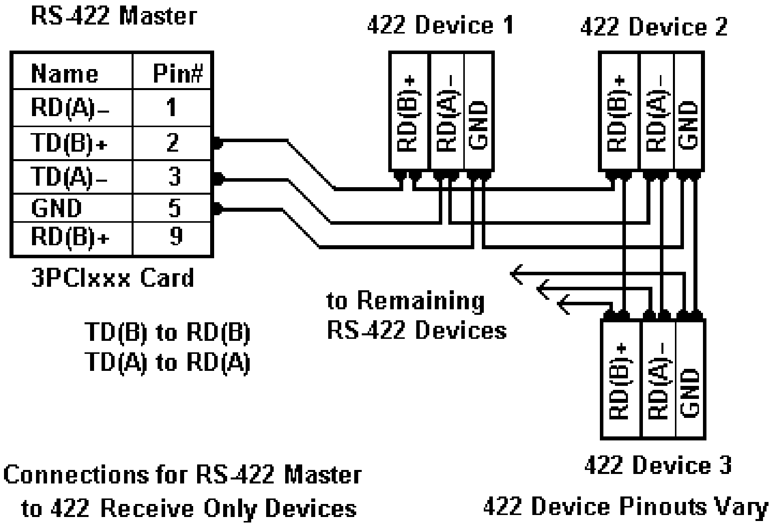
RS-422 Point to Point Connection

For RS-422 point to point communications connect the MIport card and to one RS-422 device only.

1. Connect the TD(B) pin #2 on the computer to RD(B) on the device.
2. Connect the TD(A) pin #3 on the computer to RD(A) on the device.
3. Connect the RD(B) pin #9 on the computer to TD(B) on the device.
4. Connect the RD(A) pin #1 on the computer to TD(A) on the device.
5. Connect the Signal Ground pin #5 to Signal Ground on the device.

RS-422 Point to Multipoint Connection

In a multi-slave RS-422 connection, TD(B) and TD(A) connect to RD(B) and RD(A) on all the slaves. If the slaves have Transmit connections, only one device can be connected back to the master.



RS-422 Master

Name	Pin#
RD(A)-	1
TD(B)+	2
TD(A)-	3
GND	5
RD(B)+	9

Figure 21. RS-422 Master to RS-422 Receive Only Devices

4-Wire RS-485 Connection

In a 4-wire RS-485 system, each receive line connects to the same receive terminal on all slaves, and each transmit line connects to the same transmit terminal on all slaves. The master to the first slave is connected as above.

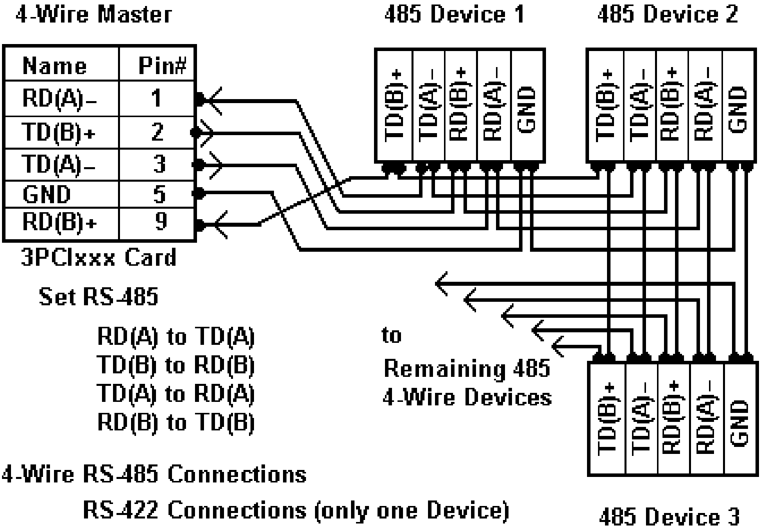


Figure 22. 4-Wire RS-422 or RS-485 Connections

Chapter 9: Troubleshooting MIport Cards

Your MIport card should be fully functional when you receive it from the factory. Operational problems encountered on first use will typically be the result of incorrect connections or operation. The following procedure will assist you in locating the source of you problems.

Starting Up

If you have any trouble starting your system after installing the card, the card may not be properly seated in the slot. Power down the computer, remove and re-insert it, or try a different slot. Ensure your system is set for PnP OS.
(Windows NT and Linux should be set to non-PnP.)

Checking Connections

RS-232/422/485 Operation

1. Ensure that you have set up your card for the correct mode of operation (RS-232, RS-422, RS-485 2-wire or 4-wire) using the DIP switches on the card and Advanced Port Settings.

> ***Note:*** *For more information on DIP switches refer to Chapter 2. For more information on Driver Settings refer to Chapter 5.*

2. Ensure your communications parameters (baud rate, parity, stop bits) are set correctly at both ends of your link.

RS-232 Operation

1. MIport cards are DTE devices. If you are connected to a DCE device use a straight through cable. If you are connected to another DTE device, use a crossover, or null cable.
2. If hardware handshaking is required, ensure you have the right cables and that both ends of your link are configured the same for handshaking.

RS-422/485 Operation

1. Check your pinouts. In RS-422 or RS-485 mode the "A" lines should match your "A" or "–" lines. "B" lines should match your "B" or "+" lines.

> ***Note:*** *RS-422/485 pinouts are non-standard.*

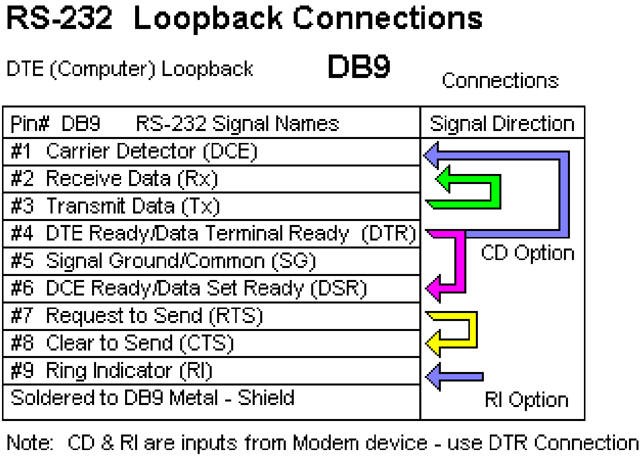
2. Make sure you have RTS Control set to the correct mode: Normal for RS-232, RS-485 Mode for RS-485. RS-422 mode works in either setting if the mode jumpers are set correctly.

Checking the MIport Card

Using ComTest and a loopback connection, you can check the operation of your MIport card separate from the rest of your communications system.

> *Note: Refer to Appendix C for information on how to install and use ComTest*

1. A loopback connection for RS-232 connects the Transmit output to the Receive input (pins #2 & #3 on the DB-9 connector). Use connections below to check all.



RS-232 Loopback Connections

DTE (Computer) Loopback DB9 Connections

Pin# DB9 RS-232 Signal Names	Signal Direction
#1 Carrier Detector (DCE)	
#2 Receive Data (Rx)	
#3 Transmit Data (Tx)	
#4 DTE Ready/Data Terminal Ready (DTR)	
#5 Signal Ground/Common (SG)	CD Option
#6 DCE Ready/Data Set Ready (DSR)	
#7 Request to Send (RTS)	
#8 Clear to Send (CTS)	
#9 Ring Indicator (RI)	
Soldered to DB9 Metal - Shield	RI Option

Note: CD & RI are inputs from Modem device - use DTR Connection

Figure 23. RS-232 Loopback w/Handshaking Connections

2. For RS-422 or 4-wire RS-485, connect the TD(A) to RD(A) and the TD(B) to RD(B). Then use the ComTest program to send characters, and observe the characters being received.

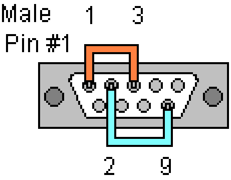


Figure 24. RS-232 Loopback Connector

3. To check 2-wire RS-485 RS-422 or 4-wire RS-485 Loopback Connections, you must either enable the receiver by moving the receive jumper to RX ENABLE mode, or use one port to transmit to another 2-wire RS-485 port or converter by cross connecting and loading ComTest twice, one copy for each port.

   Characters typed in one copy of ComTest will appear in the receive window of the other. Note that software must ignore the RS-232 handshaking lines in RS-422/RS485 mode, the input lines (CTS, DSR, DCD, RI) are not pulled high.

4. Some manufacturers label their data connections for A and B reverse of the standard for RS-422 or RS-485, so that the A line is (+) and B line is inverted (-) following the IC standard. In such a case, just swap the wires and try it. No damage occurs in the RS-485 mode, just no devices will respond. The EIA standard for signal labels defines the B line as positive relative to A during the “MARK” state.

5. Try another software package for troubleshooting. Some applications try to bypass the Windows drivers and access the hardware directly.

6. Check the B&B website for available FAQs or troubleshooting hints.

7. Contact B&B Electronics Technical Support for troubleshooting assistance.

Appendix A: DIP Switch/Mode Settings

Setting the DIP Switches on RS-232/422/485 Ports

Set the DIP switches to configure the desired operating mode as follows:

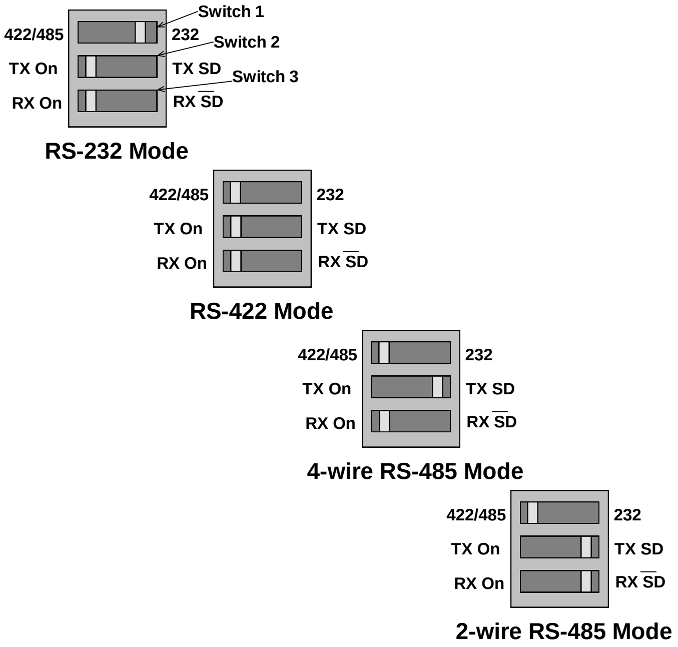


Figure 25. RS-232/422/485 DIP Switch Settings.

DIP Switch 1 (RS-232/422/485 ports)

The top DIP switch (1) configures the port for RS-232 or RS-422/485 operation. This switch is the only one that is required to be set for RS-232 operation. The positions of switches 2 and 3 do not matter when switch 1 is set for RS-232 operation.

DIP Switch 2 (RS-232/422/485 ports)

The middle DIP switch (2) configures the port for RS-485 or RS-422 operation. For RS-422 operation (which uses two wire pairs and sends point-to-point or point-to-multipoints) the transmitter can be enabled all the time. Placing the middle DIP switch in the TX ON position accomplishes this.

For RS-485 operation the middle DIP switch is placed in the TX SD position. In this position the transmitter is only enabled when data is being sent. The transmitter is tri-stated when not sending data, allowing other transmitters on the communications line to transmit without interference.

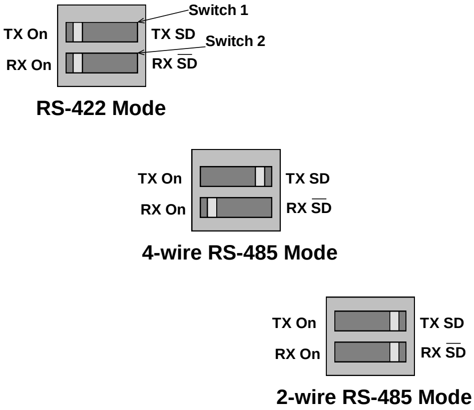
DIP Switch 3 (RS-232/422/485 ports)

The bottom DIP switch (3) configures the port for half-duplex (two-wire) RS-485 operation or full-duplex (four wire) RS-422/RS-485 operation. Placing the bottom DIP switch in the RX ON position configures the port for four wire operation. In this mode the receiver is continuously enabled, allowing it to receive all data on the communications line. Since the transmitter sends data on the other wire pair the port does not receive its own transmissions.

Placing the bottom DIP switch in the RX $\overline{\text{SD}}$ position configures the port for two wire operation. In this mode the transmitter and receiver are connected to the same wire pair. The receiver is disabled when its transmitter is sending, preventing the port from receiving its own data.

Setting the DIP Switches on RS-422/485 Only Ports

The 3PCIOU4 MIport card provides a combination of RS-232/422/485 and RS-422/485 only ports. Ports that do not include RS-232 operation use double DIP switches rather than triple DIP switches. These DIP switches operate the same as the two bottom DIP switches in the RS232/422/485 ports


Switch 1
TX On
TX SD
Switch 2
RX On
RX $\overline{\text{SD}}$
TX On
TX SD
RX On
RX $\overline{\text{SD}}$
TX On
TX SD
RX On
RX $\overline{\text{SD}}$

RS-422 Mode

4-wire RS-485 Mode

2-wire RS-485 Mode

Figure 26. RS-422/485 only DIP Switch Settings

DIP Switch 1 (RS-422/485 only)

The top DIP switch (1) configures the port for RS-485 or RS-422 operation. For RS-422 operation (which uses two wire pairs and sends point-to-point or point-to-multipoints) the transmitter can be enabled all the time. Placing the middle DIP switch in the TX ON position accomplishes this.

For RS-485 operation the middle DIP switch is placed in the TX SD position. In this position the transmitter is only enabled when data is being sent. The transmitter is tri-stated when not sending data, allowing other transmitters on the communications line to transmit without interference.

DIP Switch 2 (RS-422/485 only)

The bottom DIP switch (2) configures the port for half-duplex (two-wire) RS-485 operation or full-duplex (four wire) RS-422/RS-485 operation. Placing the bottom DIP switch in the RX ON position configures the port for four wire operation. In this mode the port's receiver is continuously enabled, allowing it to receive all data on the communications line. Since the port's transmitter sends data on the other wire pair the port does not receive its own transmissions.

Placing the bottom DIP switch in the RX $\overline{\text{SD}}$ position configures the port for two wire operation. In this mode the port's transmitter and receiver are connected to the same wire pair. The receiver is disabled when its transmitter is sending, preventing the port from receiving its own data.

Appendix B: CONNECTOR PINOUTS

RS-232 Pinouts

Name	Description	Direction (DTE)	DB-9M Pin
DCD	Data Carrier Detect	Input	1
RD	Receive Data	Input	2
TD	Transmit Data	Output	3
DTR	Data Terminal Ready	Output	4
GND	Signal Ground	------	5
DSR	Data Set Ready	Input	6
RTS	Request to Send	Output	7
CTS	Clear to Send	Input	8
RI	Ring Indicator	Input	9

Figure 27. RS-232 Signal Designations and DB-9 Pinout

Signal Name	DB-9 Pin (DTE)	DB-25 Pin (DTE)
DCD	1	8
RD	2	3
TD	3	2
DTR	4	20
GND	5	7
DSR	6	6
RTS	7	4
CTS	8	5
RI	9	22
Chassis GND	Shield (DB-9 Shell)	1

Figure 28. RS-232 DB-9 to DB-25 Conversion Cable Pinout

RS-422/485 Pinouts

Name	Description	Direction	DB-9M Pin
RDA(–)	Receive Data A (-)	Input	1
TDB(+)	Transmit Data B (+)	Output	2
TDA(–)	Transmit Data A (-)	Output	3
GND	Signal Ground	------	5
RDB(+)	Receive Data B (+)	Input	9

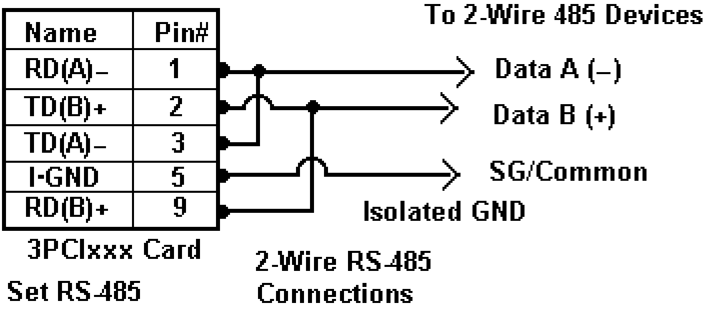
Figure 29. RS-422/485 Signal Designations and DB-9 Pinout

With 2-wire RS-485 mode operation, your connection cable must jumper TDA(-) to RDA(-) and TDB(+) to RDB(+). Connect from TDA(-) and TDB(+) to the Data A(–) and Data B(+) wires of your RS-485 network.

Note: *Refer to Chapter 2 for example connections.*

Note that the EIA RS-422 Specification labels data lines with an "A" and "B" designator. Some RS-422 equipment uses a "–" and "+" designator. In almost all cases, the "A" line is the equivalent of the "–" line and the "B" line is the equivalent of the "+" line.

Note: *For more information on RS-422 communications refer to the B&B Electronics* ***RS-422/485 Application Note*** *at www.bb-elec.com*



Name	Pin#
RD(A)–	1
TD(B)+	2
TD(A)–	3
I-GND	5
RD(B)+	9

Figure 30. 2-Wire RS-485 Wiring Diagram

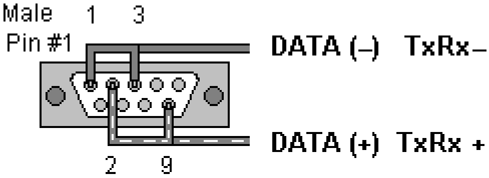


Figure 31. 2-Wire RS-485 Wiring

Appendix C: Troubleshooting With ComTest

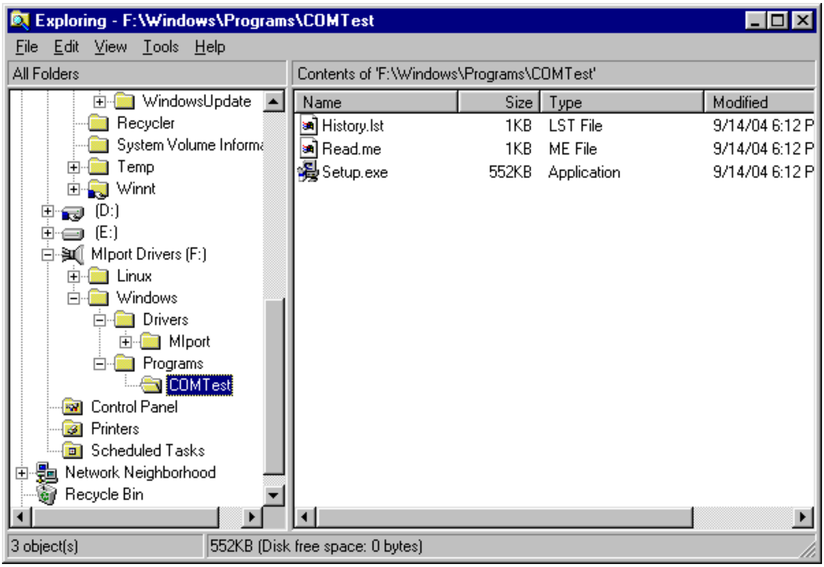
ComTest is a simple 32-bit Windows (Windows 98, 2000, 2003 Server, ME, XP, NT 4.0 and Vista) COM port test program included on the MIport CD. (It can also be downloaded from the B&B Electronics website at: **www.bb-elec.com.** The program allows multiple ports at any address and IRQ, to be opened at any given time.

ComTest Features

- A Windows Terminal Program for Simple Checks of Serial Ports. Works with USB Serial Converters & ISA or PCI Serial Cards
- Serial Port Access under Windows 98/2000/2003 Server/XP/Vista – COM1-COM8 or above.
- Dual Windows show typed transmit characters in a separate window from received characters.
- Transmit & Receive Activity Indicators
- Unprintable Characters Shown in Hex – 2 Digits Within Left & Right Angle Brackets.
- Visual Indication of Handshaking Lines – DTR DSR DCD and RTS CTS. (Red = Hi)
- Option for Setting DTR or RTS lines high or low.
- Option for Repeat Mode – last character or Function Character Sequence is repeated until set off.
- Configuration Settings:
    - Baud Rates 150, 300, 600, 1200, 2400, 9600, 19.2, 28.4, 57.6, 115.2K (9600 default and recommended)
    - Parity: None, Odd, Even, Mark, Space (None default)
    - Data Bits: 5, 6, 7, 8 (8 default)
    - Stop Bits: 1, 1.5, 2 (1 default)

Installing ComTest

1. From Windows Explorer, under Windows, Programs, COMTest, find the setup.exe file on the MIport CD.



Contents of 'F:\Windows\Programs\COMTest'

Name	Size	Type	Modified
History.lst	1KB	LST File	9/14/04 6:12 P
Read.me	1KB	ME File	9/14/04 6:12 P
Setup.exe	552KB	Application	9/14/04 6:12 P

3 object(s) 552KB (Disk free space: 0 bytes)

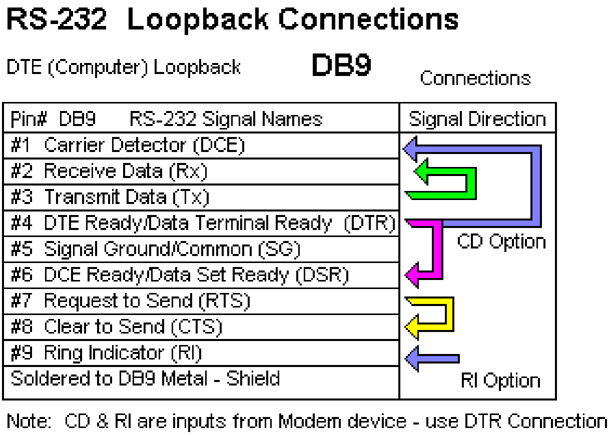
Figure 32. Loading ComTest

2. Run Setup.exe to install ComTest on your program menu under B&B Electronics.

Loopback Testing with ComTest

To familiarize yourself with the operation of ComTest connect a loopback plug to a COM port on your PC and perform the following procedure:

1. Make any required loopback connections on the port to be tested.



RS-232 Loopback Connections

DTE (Computer) Loopback DB9 Connections

Pin# DB9 RS-232 Signal Names	Signal Direction
#1 Carrier Detector (DCE)	
#2 Receive Data (Rx)	
#3 Transmit Data (Tx)	
#4 DTE Ready/Data Terminal Ready (DTR)	
#5 Signal Ground/Common (SG)	CD Option
#6 DCE Ready/Data Set Ready (DSR)	
#7 Request to Send (RTS)	
#8 Clear to Send (CTS)	
#9 Ring Indicator (RI)	
Soldered to DB9 Metal - Shield	RI Option

Note: CD & RI are inputs from Modem device - use DTR Connection

Figure 33. RS-232 Loopback w/Handshaking Connections

2. From the **Windows Desktop** click **Start → Programs → B&B Electronics → ComTest → ComTest**.

   ComTest will start and then open the **Select Port** dialog box.

3. Select the COM port you want to access or test. (The drop down box shows available ports that are not currently in use).
4. Click **OK**. The **Configure Port** dialog will appear.
5. Select the desired baud rate, parity, data bits, stop bits. (Defaults are common settings).
6. Type characters. They will appear in the upper window.

   If you are performing a loopback test, you should see the characters appear in the lower window as they are received.

7. On the **Option** menu, click **DTR** to assert or de-assert the DTR hardware handshake line. The DTR indicator at the bottom of the window will be red when asserted.

   If DTR is looped back to DSR and DCD their indicators should also be red.

8. On the **Option** menu, click **RTS** to assert or de-assert the RTS hardware handshake line. The RTS indicator at the bottom of the window will be red when asserted.

   If RTS is looped back to CTS, the CTS indicator should be red.

9. When testing is completed, close the program.