

Smartio **Multiport Boards**
C104H/HS and C168H/HS

Universal 4/8 Port
Serial Boards

Smart multiport
serial boards family with
state-of-the-art ASIC technology

Feb. 1998 (5th edition)

All Rights Reserved

Table of Contents

Chapter 1 Introduction	1
1.1 Overview	1
1.2 <i>Smartio</i> C104 Family.....	4
1.3 <i>Smartio</i> C168 Family.....	6
Chapter 2 Quick Installation	12
Chapter 3 Installation	14
3.1 Hardware Configuration - lo-irq.exe	14
Chapter 4 For Windows NT Users	20
4.1 Software Installation.....	20
4.2 Board and Port Configuration.....	21
4.3 Board Initialization Status Checking.....	22
4.4 Driver Upgrade/Removal.....	23
4.5 Baud Rate Settings	23
4.6 MOXA Serial Comm Tool: <i>PComm</i>	24
4.7 Troubleshooting	24
Chapter 5 For Windows 95 Users	26
5.1 Driver Installation - Setup95.exe	26
5.2 Board and Port Configuration.....	26
5.3 Board Initialization Status Checking.....	28

5.4	Driver Upgrade/Removal.....	28
5.5	Baud Rate Settings	28
5.6	MOXA Serial Comm Tool: <i>PComm</i>	29
5.7	Troubleshooting	29
Chapter 6	<i>PComm</i>.....	32
6.1	Installation.....	32
6.2	Programming Library.....	32
6.3	Utilities	33
Chapter 7	For UNIX Users.....	34
7.1	MOXA UNIX Driver for C168H/C168HS.....	34
7.2	MOXA TTY Device Naming Convention	37
7.3	Baud Rate Settings	38
7.4	Administrations	39
7.5	Setting MOXA Ports to Terminal	41
7.6	Programming the MOXA Ports.....	41
7.7	Troubleshooting	42
Chapter 8	For DOS Users	44
8.1	Driver Installation, Setup, Loading and Unloading	44
8.2	MOXA Serial Port Naming Convention	47
8.3	Baud Rate Settings	47
8.4	DOS API-232 Library and Programming	48
8.5	Data Scope	48
Chapter 9	For Windows 3.x Users.....	50

Appendix A	Troubleshooting	54
Appendix B	C104 Pinouts and Cable Wiring	58
B.1	RS-232 Cable Wiring.....	58
B.2	DB62 Connector Pinouts	60
Appendix C	C168 Pinouts and Cable Wiring	62
C.1	RS-232 Cable Wiring.....	62
C.2	RS-422 Cable Wiring.....	64
C.3	RS-485 Cable Wiring.....	65
C.4	RS-422/485 Impedance Matching	68
C.5	DB62 Connector Pinouts	70
Appendix D	RS-485 Programming	72
Appendix E	Extended UNIX ioctl () Commands	76
Appendix F	PC I/O Port Address Map	80

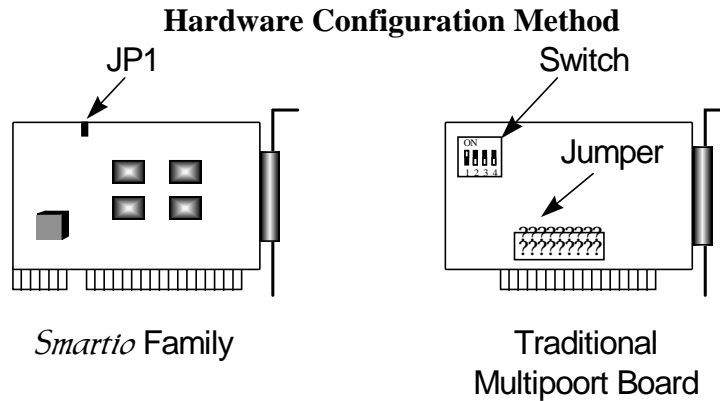
1.1 Overview

Smartio - The Smart Multiport Async Solutions

The term *Smartio* stands for smart multiport serial I/O solution. The *Smartio* **C104/C168 Family** multiport boards offer 4/8 serial ports for connecting terminals, modems, printers, data acquisition equipment and any other serial devices to the PC/AT and its compatible systems. With the well-designed and fine-tuned device driver, the *Smartio* boards make full use of the 32 byte Tx/Rx FIFO and on-chip H/W flow control, so that they can transfer data without data loss even at high speed such as 921.6 Kbps, which offers a reliable and high performance solution for serial multiport communications.

The *Smartio* C104/C168 Family is equipped with **custom-designed ASIC chip** which replaces lots of conventional ICs and reduces the board to half-size. The whole family supports 16 bit architecture. Full range of I/O addresses and IRQs are available. In addition, with **on-board EEPROM** for storing the configuration data, the family is designed without jumper or switch. These features make each port on the board truly independent to any other port and thus compatible with most existing multiport boards.

The *Smartio* Family is also available in PCI bus. Please contact MOXA dealer/distributor or MOXA Web site for more details.



New : set I/O address and IRQ via software Utility.

Traditional : set switch and jumper manually for I/O address and IRQ.

Instead of using traditional jumper or switch for IRQ and I/O address settings, hardware configuration of each port is easily set by DOS utility, **Io-irq.exe**, which read and write the on-board EEPROM for configuration information through the **CAP (Configuration Access Port) address**. The CAP address is the only channel via which the configuration utility Io-irq.exe can access the board, which is identical to **the first port's base I/O address**.

The only jumper, **JP1**, is designed in case that users forget the CAP address. Normally JP1 is left open. When JP1 is short, the CAP address is forced to a fixed I/O address, **0xA700**. However, to adopt quick installation (described later), it is a must to keep JP1 always short.

Quick Installation

To ease the hardware configuration, users who install **only one Smartio C104/C168 board under Windows NT/95** are recommended to adopt **quick installation** described in Chapter 2. Other users please see Chapter 3 for installation details.

Because the family is so flexible in hardware configuration that they are compatible virtually with all kinds of other manufacturer's multiport boards using 16450 or 16550 UART.

Surge Protection

To prevent the boards from damage caused by lightning or high potential voltage, **surge protection** technology is introduced in some model to protect the board.

Operating System Support

The family is operational under most popular operating systems such as Windows NT, Windows 95, SCO UNIX/XENIX/OpenServer, DOS, Windows 3.x, OS/2, Linux, QNX, FreeBSD, PICK OS, IMS Multiuser DOS, etc. However, **MOXA device drivers** for Windows NT, Windows 95, Linux, SCO UNIX/OpenServer, UNIX SVR4.2, DOS and Windows 3.x are provided for better installation, configuration and performance. In this manual, chapters for **MOXA Windows NT, Windows 95, UNIX, DOS and Windows 3.x** device drivers are included. For other compatible systems not mentioned, please refer to the respective operating system's manual for how to install and configure the standard driver.

MOXA Serial Comm Tool

For easy application development, MOXA provides an easy-use serial communication library under Windows NT/95 (*PComm*) and Windows 3.x/DOS (**API-232**). Users can use this library to develop your own applications using Microsoft C, Turbo C, Assembly, QuickBASIC, Turbo Pascal, Clipper, Visual Basic, Visual C++, Borland Delphi, etc. **Utilities**, such as diagnostic and monitor, are included for diagnosing the board/port or monitoring the communication status.

1.2 *Smartio* C104 Family

The *Smartio* C104 Family consists of members as follows,

- C104H** 4 port RS-232, high speed TI550C UART
- C104HS** 4 port RS-232, surge protection, TI550C UART

Features

- √ Custom-designed ASIC, compact board size (half-size)
- √ No switch no jumper, easily configured by software
- √ Independent I/O address, IRQ setting for each of 4 serial ports
- √ 16 bit AT bus architecture, more IRQs supported
- √ Surge protection (C104HS)
- √ High speed TI550C Communication Controller with on-chip hardware flow control, no data loss
- √ Windows NT/95/3.x and DOS device drivers
- √ *PComm* serial comm tool

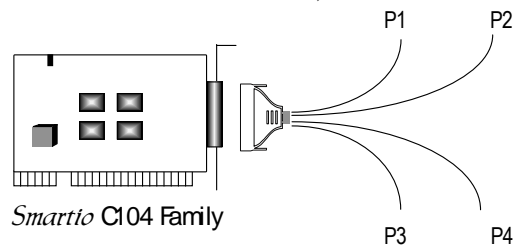
Specifications

- √ Bus interface : ISA (EISA compatible)
- √ Number of ports : 4
- √ I/O address : 0x0000 ~ 0xFFFF
- √ IRQ : 2, 3, 4, 5, 7, 10, 11, 12, 15
- √ Data bits : 5, 6, 7, 8
- √ Stop bits : 1, 1.5, 2
- √ Parity : none, even, odd, space, mark
- √ UART : 4»TI550C
- √ Speed (bps) : 50 ~ 921.6K
- √ Connectors : 4»DB25 male (DTE)
- √ Data signals : TxD, RxD, RTS, CTS, DTR, DSR, DCD, GND
- √ Surge protection : max. 2000V (C104HS)
- √ Operating temp. : 0 ~ 55 ¼[
- Power requirement : 90mA max. (+5V), 55mA max. (+12V), 75mA max. (-12V)
- √ Dimensions : 15.7cm»8.3cm

Check List

Upon unpacking the *Smartio* C104 Family, you should find the following items in the package,

1. *Smartio* C104 Family 4 port serial board
2. This User's Manual
3. Windows NT/95 driver diskette»1, DOS/Windows 3.x (API-232) driver diskette»1, *PComm* diskette»1
4. 30 cm cable (DB37 to 4»DB25 male connectors)

**System Support**

The *Smartio* C104 Family is compatible with various operating systems such as Windows NT, Windows 95, SCO UNIX/XENIX, OS/2, QNX and Linux. However, Moxa supports proprietary drivers for better performance. See the driver list below.

	<u>C104H/HS</u>
Windows NT	✓
Windows 95	✓
Windows 3.x	✓
DOS	✓
Linux	R
QNX	C
SCO UNIX/OpenServer	C
SCO XENIX	C
PICK OS	C
OS/2	C
MUMPS	C
Multuser DOS	C
FreeBSD	C

✓: Driver supported by Moxa and shipped with product
 R: Driver supported by Moxa but shipped by request
 C: Driver supported by OS
 Note: MOXA FTP site is available for driver download

1.3 *Smartio* C168 Family

The *Smartio* C168 Family consists of members as follows,

C168H 8 port RS-232 or RS-422, high speed TI550C UART
C168HS 8 port RS-232 or RS-422, surge protection, TI550C UART

Features

- √ Custom-designed ASIC, compact board size (half-size)
- √ No switch no jumper, easily configured by software
- √ Independent I/O address, IRQ setting for each of 8 serial ports
- √ 16 bit AT bus architecture, more IRQs supported
- √ Surge protection for RS-232 (C168HS)
- √ Isolation protection for RS-422 (optional connection box Opt8F)
- √ High speed TI550C Communication Controller with on-chip hardware flow control, no data loss
- √ Windows NT/95, DOS (API-232) device drivers
- √ SCO UNIX/OpenServer and UNIX SVR4.2 device drivers
- √ *PComm* serial communication tool

Specifications

- √ Bus interface : ISA (EISA compatible)
- √ Number of ports : 8
- √ I/O address : 0x0000 ~ 0xFFFF
- √ IRQ : 2, 3, 4, 5, 7, 10, 11, 12, 15
- √ Data bits : 5, 6, 7, 8
- √ Stop bits : 1, 1.5, 2
- √ Parity : none, even, odd, space, mark
- √ UART : 8»TI550C
- √ Speed (bps) : 50 ~ 921.6K
- √ Connectors : 8»DB25 male or female

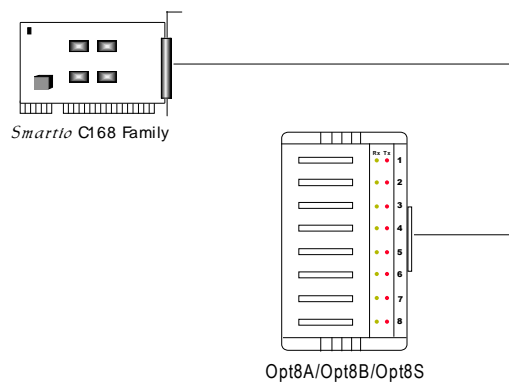
- v Data signals : RS-232»TxD, RxD, RTS, CTS, DTR, DSR, DCD, GND
 RS-422»TxD+(B), TxD-(A), RxD+(B), RxD-(A), GND
 (Opt8J: RTS+(B), RTS-(A), CTS+(B), CTS-(A))
 RS-485»TxD+(B), TxD-(A), RxD+(B), RxD-(A), GND
- v Surge protection : max. 2000V (C168HS)
- v Operating temp : 0 ~ 55 ¼
- v Power requirement: 180mA max. (+5V), 110mA max. (+12V), 160mA, max. (-12V)
- v Dimensions : 15.7cm»9.3cm

Check List

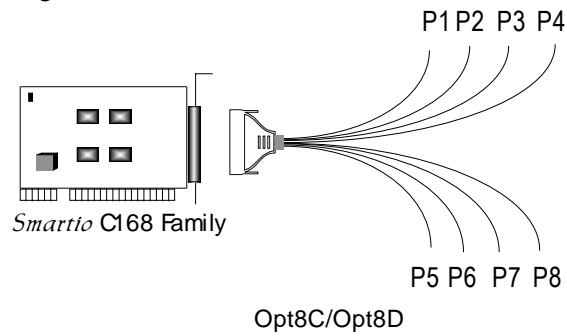
Upon unpacking the *Smartio* C168 Family, you should find the following items in the package,

1. *Smartio* C168 Family 8 port serial board
2. This User Manual
3. Windows NT/ 95 driver diskette»1, DOS (API-232) driver diskette»1, SCO UNIX/OpenServer and UNIX SVR4.2 device driver diskette»1, and *PComm* diskette»1
4. One of the following interface options:

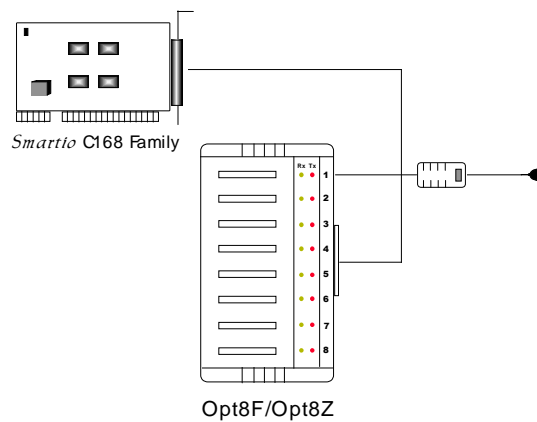
Opt8A/Opt8B/Opt8S: RS-232 connection box with 8 DB25 female/male ports (surge protection for Opt8S) and 1.5 meter DB62 to DB62 cable.



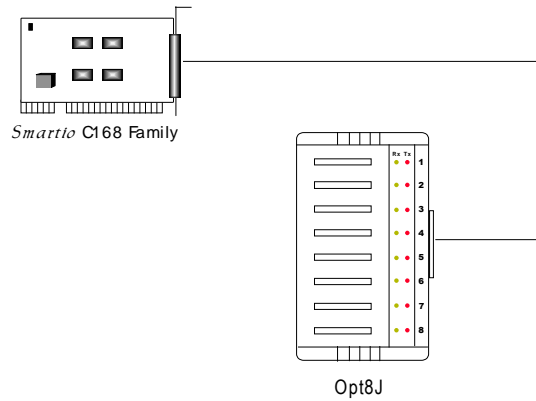
Opt8C/Opt8D: RS-232 octopus cable with 8 port male DB25 (or DB9 for Opt8D) connectors (1 meter long).



Opt8F/Opt8Z: RS-422 connection box with 8 port female DB25 connectors (isolation protection for Opt8F). 1.5 meter DB62 to DB62 cable and 110V or 220V adapter included.



Opt8J: RS-422/485 connection box with 8 port female DB25 connectors. 1.5 meter DB62 to DB62 cable and 110V or 220V adapter included.



System Support

The *Smartio* C168 Family is compatible with various operating systems such as SCO UNIX/XENIX, QNX and Linux. However, Moxa supports proprietary drivers for better performance. See the driver list below.

	<u>C168H/HS</u>
Windows NT	✓
Windows 95	✓
DOS	✓
SCO UNIX/OpenServer	✓
UNIX SVR4.2	✓
Linux	R
QNX	C
SCO XENIX	C
PICK OS	C
Multiuser DOS	C
FreeBSD	C

✓: Driver supported by Moxa and shipped with product

R: Driver supported by Moxa but sent by request

C: Driver supported by OS

Note: MOXA FTP site is available for driver download

To fully utilize the superior feature of flexible hardware configuration design of the *Smartio* C104/C168 Family, a quick and easy method of installation is designed for users, which absolutely free the users from hardware configuration effort, i.e., **installation without running configuration program: Io-irq.exe**. Simply always short the jumper JP1. The software and hardware configuration will be completed at the same time while doing the software configuration.

However, the method is valid under the circumstances that:

- u **Only one *Smartio* C104/C168 board** is allowed to be installed in a system
- u **Windows NT/95** are the only operating systems supported
- u I/O address **0xA700** must be free

Besides, the speed range will be set to **from 50 to 921.6K bps by default**, which is called **High Speed Spectrum** and detailed in Chapter 3 Section 3.1 Hardware Configuration - Io-irq.exe.

Users who install **only one *Smartio* C104/C168 board under Windows NT/95** are strongly recommended to do the quick installation as follows:

1. **Short the jumper JP1** on the upper-left corner of the board.
2. Plug the board in PC with the desired system installed, which is powered off in advance.
3. Do driver installation. See "**For Windows NT Users**" or "**For Windows 95 Users**" or corresponding chapters for details.

This is to specify the desired I/O address, IRQ and INT Vector in the software configuration panel, no matter what hardware settings are on the board. The software configuration program will automatically update the hardware settings.

After this, users already complete the whole installation.

4. Shutdown System (Windows NT/95).
5. DO power OFF and then ON (or Reset) the PC. (**Please cold start.**)
6. Restart System (Windows NT/95).

It is very important to keep the JP1 always short in this case. Without running the hardware configuration program, Io-irq.exe under DOS prompt, the software configuration program will automatically update the hardware settings of the board while updating the software settings. This save the trouble doing hardware configuration. However, remember to **cold start the system every time the configuration changed**. Should any problem occurs, see Section "Troubleshooting" within each chapter.

This chapter is for users in the following cases, who can not use quick installation:

- u Install two or more *Smartio* C104/C168 boards in a system.
- u Fail to install due to the I/O address 0xA700 is not available or has conflict in the system.
- u Use operating systems other than Windows NT/95.

The installation of *Smartio* C104/C168 Family is as follows:

1. Start with **hardware configuration**. See next Section for **Io-irq.exe**.
2. Continue with the **software (driver) installation** according to the corresponding operating system chapters for details.

In this manual, chapters for MOXA **Windows 95**, **Windows NT**, **UNIX**, **DOS** and **Windows 3.x** device drivers are included. For other compatible systems not mentioned, please refer to system's manual for how to install and configure the standard driver.

Remember to keep the hardware settings in mind for the software installation.

After this, users already complete the whole installation.

3.1 Hardware Configuration - Io-irq.exe

Because the ASIC-designed *Smartio* C104/C168 Family has no switch no jumper, users **must** configure the I/O address, IRQ, INT Vector, etc. of the boards via utility, **Io-irq.exe**, before proceed to install. To set up the hardware

configuration for *Smartio* C104/C168 Family, please run **Io-irq.exe** in the driver diskette **under DOS** system.

1. Choose a PC that has **DOS** system inside.
2. Power off the PC
3. Make sure no hardware conflict and plug the board in a free 16-bit slot of the PC, **one board at a time with JP1 open.**



If you are installing multiple boards, insert one board at a time and configure it using the Io-irq program before inserting the next board. This is to prevent conflict between two boards with same default hardware settings.

The *Smartio* C104/C168 Family has the following default (factory) settings, I/O address:

	0x180 (1st port), 0x188 (2nd port)	
	0x190 (3rd port), 0x198 (4th port)	
	0x1A0 (5th port), 0x1A8 (6th port)	C168 Only
	0x1B0 (7th port), 0x1B8 (8th port)	C168 Only
IRQ:	10 (shared by 4/8 ports)	
INT Vector:	0x1C0	
Configuration Access Port (CAP):	0x180	

4. Power on the PC and enter into **DOS** system.
5. Run the utility "**Io-irq.exe**" contained in the driver diskette to set up I/O address, IRQ and INT vector of the board. Follow the on-line help to configure the *Smartio* C104/C168 Family board.

After completing the hardware configuration, the board is ready for use under operating systems such as Windows NT/95, SCO UNIX, DOS, etc.

Note that the CAP address is identical to **the first port's I/O address** except in one case that the JP1 jumper is installed before powering on the PC. In this case, the CAP address will be forced to **0xA700**. The CAP address must

be typed correctly. With the correct CAP address, the utility can find the configuration stored in the **on-board EEPROM** and display it on the configuration panel. The CAP address is the only channel via which the configuration utility Io-irq.exe can access the board.

The followings are fields for configuration:

Port Index: Indicate the port index for each port.

I/O address: Enter the I/O address for each port, either sequentially or not. Avoid to conflict with any other devices.



Hint: It is a good way to save your time if you can first determine in which operating system, e.g. Windows or OS/2 or SCO UNIX, the *Smartio* C104/C168 board will be used. In order to ease the configuration, the common used I/O address and IRQ combinations supported by most popular OSs have been collected in the IO-IRQ utility. Press "**F3 - Select OS Compatibility**" on 3rd level menu to bring the suggested list and choose a proper one. You can, however, enter the I/O and IRQ by yourself if no one is suitable.

For example, if the *Smartio* C104/C168 board is to be installed under SCO UNIX system as the primary (first) four port board, here is one of the possible settings,

- a. Press "**F3 - Select OS Compatibility**" on 3rd level menu.
- b. Then choose the entry of "SCO UNIX 1 (Arnet Compatible ..)" which will designate the *Smartio* C104/C168's I/O base to 0x280, IRQ to 4 and interrupt vector to 0x2C2.

Speed: This field specify the use of **normal or high speed** capability. Normal speed ranges from 50 bps to 115.2 Kbps. High speed ranges from 50 to 921.6K bps. **Smartio C104/C168 support both normal and high speed spectra**, such as C104H, C104HS, C168H and C168HS. In other word, to use speed faster than **115.2 Kbps**, e.g. 460.8 Kbps, *Smartio* C104/C168 set to High Speed Spectrum is required.

Note that, currently, port that uses **MOXA Windows NT/95 driver** will run at the displayed speed. To be clear, when *Smartto C104/C168* board is configured as **High Speed Spectrum**, any port driven by the **Moxa-provided Windows NT/95 driver** will display the exact working speed. For example, the displayed speed 38.4 Kbps is equal to the working speed 38.4 Kbps.

However, if the port is driven by **non Moxa-provided driver**, such as **standard serial driver**, or Moxa drivers other than Windows NT/95, such as **DOS, UNIX and Windows 3.x**, the real working speed is equal to **8 times** of the displayed speed. For example, a port, if set to Normal Speed Spectrum with 38.4 Kbps, will work at 38.4 Kbps for sure; while a port, if set to High Speed Spectrum with displayed speed 38.4 Kbps, will actually work at 307.2 Kbps (38.4 Kbps \times 8).

The following is the **8 times speed mapping list** for quick reference purpose.

Normal Speed Spectrum	High Speed Spectrum
50 (bps)	400 (bps)
75	600
110	880
134.5	1076
150	1200
300	2400
600	4800
1200	9600
1800	14.4K
2400	19.2K
4800	38.4K
7200	57.6K
9600	76.8K
19.2K	153.6K
38.4K	307.2K
57.6K	460.8K
115.2K	921.6K

IRQ: Enter the IRQ, 2, 3, 4, 5, 7, 10, 11, 12 or 15, for each port, independently or not.

Interrupt Vector: Enter the Interrupt vector address for all ports. I/O address for Interrupt vector is from 00000H to 0FFFFH. Interrupt vector is

one byte of I/O address, in which each bit is used to indicate the occurrence of interrupt for corresponding port. To use Interrupt vector, type in the hardware Interrupt vector I/O address. If not using Interrupt vector, type 0 or leave blank as the Interrupt vector.

There are two modes for the *Smartio* C104/C168 driver. One is using Interrupt vector, the other is not using Interrupt vector. Driver employing Interrupt vector scheme is supposed to have better performance than employing polling scheme.

Chapter 4

For Windows NT Users

In this chapter, software driver installation, configuration and upgrade/removal procedure are described. If it is necessary for you to develop your own applications, we strongly recommend the easy but powerful *PComm* serial comm tool, which includes library for easy programming and useful utilities for diagnostic and monitor. Finally, Windows NT-specific troubleshooting is included.

4.1 Software Installation

The following is the procedure for installing the *Smartio* C104/C168 Family device driver under Windows NT 4.0 (or NT 3.51):

1. Login NT as **Administrator**.
2. Enter [**Control Panel**] group, then select [**Network**] icon, and then [**Adapters**] folder.
(Enter [**Control Panel**] folder, then select [**Network**] icon for NT 3.51.)
3. Click [**Add**] button, then [**Have Disk...**] button for "Select Network Adapter".
(Click [**Add Adapter...**] button, then "<Other> **Requires disk from manufacturer**" item for "Network Adapter Card:" for NT 3.51.)
4. Specify the exact path for the driver diskette, e.g. **A:\WINDOWS.NT**. Click [**Continue/OK**].
5. Select "**MOXA C102/C104/C168 Family Adapter**" in "Select OEM Option" dialog. Click [**OK**] to start installation.
6. When MOXA C102/C104/C168 Configuration Panel is popped up, to configure the board(s) and ports, please refer to the Section "**Board and Port Configuration**" for more details.
7. Select [**OK**] for "Network Settings" dialog. Restart system as prompted.

4.2 Board and Port Configuration

The following is the procedure for configuring the *Smartio* C104/C168 Family driver under Windows NT 4.0 (or NT 3.51):

1. Enter **[Control Panel]** group, then select **[Network]** icon, and then **[Adapters]** folder.
(Enter **[Control Panel]** folder, then select **[Network]** icon for NT 3.51.)
2. Select "MOXA C102/C104/C168 Adapter" item for "Installed Adapter Cards:", then **[Configure/Properties]** button to start the "MOXA C102/C104/C168 Family Configuration Panel" dialog.

When the configuration panel is popped up:

Click **[Add]** button to add a board.

Click **[Remove]** button to remove a board.

Click **[Properties]** button to set up a board with correct "Board Type", "IRQ", "INT Vector", "I/O Address", "COM Number" and "Received FIFO Trigger Level", where the possible parameters are as follows:

Board Type: C104, C168 Family.

IRQ No.: 2, 3, 4, 5, 7, 10, 11, 12, 15

INT Vector: I/O address for interrupt vector from 00000H to 0FFFFH. To use INT vector, type in the hardware INT vector I/O address. If not using INT vector, type 0 or leave blank as the INT vector.

There are two modes for the C104/C168 driver. One is using INT vector, the other is not using INT vector. INT vector is one byte of I/O address, in which each bit is used to indicate the occurrence of interrupt for corresponding port. Driver employing INT vector scheme is supposed to

have better performance than employing polling scheme.

Sequential Port Mapping: For I/O Address, specify the I/O address of the first port and subsequent ports are set to continuous I/O addresses by the offset of 8 bytes. For instance, if the first port is set to 0x180, then the second port is set to 0x188 sequentially. For COM number, specify the COM number of the first port and subsequent ports are mapped to continuous COM numbers. For instance, if the first port is mapped to COM10, then the second port is mapped to COM11 sequentially. For receive FIFO trigger level (bytes), every port will have the same trigger level as set. For 16550-compatible type of UART, receive FIFO trigger level is 1, 4, 8 or 14 bytes.

Non-sequential Port Mapping: For I/O Address, set the I/O address of each port independently. For COM number, set the COM number of each port independently. For receive FIFO trigger level (bytes), set the trigger level of each port independently. For 16550-compatible type of UART, receive FIFO trigger level is 1, 4, 8 or 14 bytes.

At most 4 boards of C102/C104/C168 Family can be installed together as long as the IRQ and I/O address resources are sufficient and available in a system.

3. Unless the system restarted, the latest configuration will not take effect.

4.3 Board Initialization Status Checking

There are two alternative ways to find out if ports of the board are initialized successfully:

1. Enter [**Control Panel**] group and then [**Ports**] icon, to check the port list and see if all configured ports are already added in the system correctly.

If no configured port shown in the port list, refer to Section "**Troubleshooting**" for solutions.

2. Enter [**Administrative**] group, then [**Event Viewer**] icon, and then [**System**

Event Log] to check for message like "COM10 (C168 port 8) O.K." for each configured port.

If message is "Can not find the configured C102/C104/C168 (CAP=0XXXX) !", refer to Section "**Troubleshooting**" for solutions.

4.4 Driver Upgrade/Removal

To upgrade driver, remove the installed driver first and install the new one.

To remove driver,

1. Enter [**Control Panel**] group, then select [**Network**] icon, and then [**Adapters**] folder.
(Enter [**Control Panel**] folder, then select [**Network**] icon for NT 3.51.)
2. Select "**MOXA C102/C104/C168 Adapter**" in adapter list, then [**Remove**] button to start removing the MOXA board.

4.5 Baud Rate Settings

When *Smartio* board is configured as **High Speed Spectrum**, any port driven by the **Moxa-provided driver** will display the exact working speed. For example, the displayed speed 38.4 Kbps is equal to the working speed 38.4 Kbps.

However, if the port is driven by **non Moxa-provided driver**, such as **standard serial driver**, the real working speed is equal to **8 times** of the displayed speed. For example, a port, if set to Normal Speed Spectrum with 38.4 Kbps, will work at 38.4 Kbps for sure; while a port, if set to High Speed Spectrum with displayed speed 38.4 Kbps, will actually work at 307.2 Kbps (38.4 Kbps»8).

4.6 MOXA Serial Comm Tool: *PCComm*

PCComm is a software package consisting of MOXA serial communication library and utilities for diagnostic and monitor as well as example programs under Windows NT/95. Please see Chapter 6 "*PCComm*" for details.

4.7 Troubleshooting

The error messages and solutions are stated as clearly as possible. If all the possible solutions fail, the board or connection box might be defective. Please check the board or connection box ONE AT A TIME in the system and find out the defective one.

1. Can not find the configured C104/C168 (CAP=0xXXXX) !

To avoid this, please double check the board settings from the driver configuration for each board installed, then shut down the system before checking the following items:

- Plug in the board(s) properly.
- Inspect carefully on the settings of I/O address for each board installed.
- Sometimes slot for plugging board is malfunctional.
In this case, please try other slots until good one is found.
- The board might be defective.
- For "quick installation" users, make sure the jumper JP1 is always short and the additional I/O address 0xA700 is not occupied by other devices.

If some other device use 0xA700 at the same time, I/O address conflict occurs. For example, a **partially decoded** device is using I/O address **0x300**. This will cause conflict with 0xA700, a **fully decoded** I/O address. In this case, either find out the other device causing conflict and change its I/O address if possible. Or use normal installation procedure for C104/C168 board, instead, to avoid conflict.

2. The C104/C168's software and hardware configuration mismatch (CAP=0xXXXX) !

- To avoid this, please check the board settings of the software configuration via entering [Control Panel] group, then [Network] icon, then [Adapters] folder and then [MOXA C102/C104/C168 Configuration Panel] for each board installed, then shut down the system and run Io-irq.exe to check the board settings of the hardware configuration. Make sure both are same.
- The board might be defective.

Chapter 5

For Windows 95 Users

In this chapter, software driver installation, configuration and removal procedure are described. If it is necessary for you to develop your own applications, we strongly recommend that try out the easy but powerful *PComm* serial comm tool, which includes MOXA serial communication library for easy programming and utilities for diagnostic and monitor under Windows NT/95. Finally, Windows 95-specific troubleshooting is included.

5.1 Driver Installation - Setup95.exe

1. Run **Setup95.exe** in the driver diskette.
2. Select "**MOXA C102/C104/C168 Family Driver**" to install the driver.
3. When MOXA C102_C104_C168 Configuration Panel is popped up, to configure the board(s) and ports, please refer to the Section "**Board and Port Configuration**" for more details.

After this, users already complete the whole installation.

4. Restart Windows 95.
5. Check the board initialization status after logging in the system. Refer to Section "**Board Initialization Status Check**" for details.

5.2 Board and Port Configuration

You will enter the configuration program when installing the driver. Or you can click on the Taskbar [Start] button, then select [Programs] menu, then [MOXA Utilities] menu, and then [MOXA C102_C104_C168 Configuration Panel] icon. When the configuration panel popped up:

Click [**Add**] button to add a board.

Click [**Remove**] button to remove a board.

Click [**Properties**] button to set up a board with correct "Board Type", "IRQ", "INT Vector", "I/O Address", "COM Number" and "Received FIFO Trigger Level", where the possible parameters are as follows:

Board Type: C104, C168 Family.

IRQ: 2, 3, 4, 5, 7, 10, 11, 12, 15.

INT Vector: I/O address for interrupt vector from 00000H to 0FFFFH. To use INT vector, type in the hardware INT vector I/O address. If not using INT vector, type 0 or leave blank as the INT vector.

There are two modes for the C104/C168 driver. One is using INT vector, the other is not using INT vector. INT vector is one byte of I/O address, in which each bit is used to indicate the occurrence of interrupt for corresponding port. Driver employing INT vector scheme is supposed to have better performance than employing polling scheme.

Sequential Port Mapping: For I/O Address, specify the I/O address of the first port and subsequent ports are set to continuous I/O addresses by the offset of 8 bytes. For instance, if the first port is set to 0x180, then the second port is set to 0x188 sequentially. For COM number, specify the COM number of the first port and subsequent ports are mapped to continuous COM numbers. For instance, if the first port is mapped to COM10, then the second port is mapped to COM11 sequentially. For receive FIFO trigger level (bytes), every port will have the same trigger level as set. For 16550-compatible type of UART, receive FIFO trigger level is 1, 4, 8 or 14 bytes.

Non-sequential Port Mapping: For I/O Address, set the I/O address of each port independently. For COM number, set the COM number of each port independently. For receive FIFO trigger level (bytes), set the trigger level of each port independently. For 16550-compatible type of UART, receive FIFO

trigger level is 1, 4, 8 or 14 bytes.

At most 4 boards of C102/C104/C168 Family can be installed together as long as the IRQ and I/O port resources are sufficient and available in a system.

Unless the system restarted, the latest configuration will not take effect.

5.3 Board Initialization Status Checking

All the error conditions, during the initialization of Windows 95, will be popped up onto the screen. Otherwise, everything should be fine.

If message like "Can not find the configured C102/C104/C168 (CAP=0xXXXX) !" or "The C102/C104/C168's software and hardware configuration mismatch (CAP=0xXXXX) !" shows, refer to Section "Troubleshooting" for solutions.

5.4 Driver Upgrade/Removal

To upgrade the driver, simply install the driver again.

To remove driver, just remove all the configured boards. Once the MOXA driver is installed, it exists permanently in Windows 95 like other adapters do. Removing changes nothing but that particular board's existence in Windows 95 Registry.

5.5 Baud Rate Settings

When *Smartio* board is configured as **High Speed Spectrum**, any port driven by the **Moxa-provided driver** will display the exact working speed. For example, the displayed speed 38.4 Kbps is equal to the working speed 38.4 Kbps.

However, if the port is driven by **non Moxa-provided driver**, such as **standard serial driver**, the real working speed is equal to **8 times** of the

displayed speed. For example, a port, if set to Normal Speed Spectrum with 38.4 Kbps, will work at 38.4 Kbps for sure; while a port, if set to High Speed Spectrum with displayed speed 38.4 Kbps, will actually work at 307.2 Kbps (38.4 Kbps»è8).

5.6 MOXA Serial Comm Tool: *PCComm*

PCComm is a software package consisting of MOXA serial communication library and utilities for diagnostic and monitor as well as example programs under Windows NT/95. Please see Chapter 6 "*PCComm*" for details.

5.7 Troubleshooting

The error messages and solutions are stated as clearly as possible. If all the possible solutions fail, the board or connection box might be defective. Please check the board or connection box ONE AT A TIME in the system and find out the defective one.

1. Can not find the configured C104/C168 (CAP=0xXXXX) !

To avoid this, please double check the board settings via clicking on [Start] button, then [Programs] menu, then [MOXA Utilities] menu, and then [MOXA C102_C104_C168 Configuration Panel] icon for each board installed, then shut down the system before checking the following items:

- Plug in the board(s) properly.
 - Inspect carefully on the setting of I/O address for each board installed.
 - Sometimes slot for plugging board is malfunctional.
In this case, please try other slots until good one is found.
 - The board might be defective.
 - For "quick installation" users, make sure the jumper JP1 is always short and the additional I/O address 0xA700 is not occupied by other devices.
- If some other device use 0xA700 at the same time, I/O address conflict occurs. For example, a **partially decoded** device is using I/O address **0x300**.

This will cause conflict with 0xA700, a **fully decoded** I/O address. In this case, either find out the other device causing conflict and change its I/O address if possible. Or use normal installation procedure for *Smartio* C104/C168 board, instead, to avoid conflict.

2. The C104/C168's software and hardware configuration mismatch (CAP=0XXXX) !

- To avoid this, please check the board settings of the software configuration via clicking on [Start] button, then [Programs] menu, then [MOXA Utilities] menu, and then [MOXA C102_C104_C168 Configuration Panel] icon for each board installed, then shut down the system and run Io-irq.exe to check the board settings of the hardware configuration. Make sure both are same.
- The board might be defective.

PComm, the professional serial comm tool for PC, is a software package under Windows NT/95, which consists of powerful serial communication library for easy programming in most popular languages, useful utilities such as diagnostic and monitor, illustrative example programs and comprehensive on-line documents.

The serial communication library is especially for users who develop a system for data communication, remote access, data acquisition or industrial control in the Windows NT/95 environment, which offers an easier solution compared with the more complex Windows Win32 COMM API.

6.1 Installation

To install *PComm*, please run `\Setup.exe` in the diskette.

Note that the MOXA Windows NT/95 device driver is required as using *PComm* diagnostic and monitor utilities.

6.2 Programming Library

The serial communication library is to assist users to develop programs for serial communications **for any COM port** complying with Microsoft Win32 API. It can ease the implementation of multi-process and multi-thread serial communication programs and hence greatly reduce the developing time.

For complete library function description and example programs for Visual C++, Visual Basic and Delphi, please see help file and example programs in *PComm* directory for more details.

6.3 Utilities

The followings are short descriptions of each utility. For details, please see on-line help in the diskette.

Diagnostic

A convenient diagnostic program provides internal and external testing, such as IRQ, TxD/RxD, UART, CTS/RTS, DTR/DSR, DTR/DCD testing, etc., for the MOXA boards and ports to verify correct operation of both the software and hardware.

Monitor

A useful port status monitoring program allows users to watch the selected MOXA COM ports' data transmitting/receiving throughput and communication line status which are updated and displayed on the screen at every time interval. In addition, users may click on one of the specific displayed port in order to see the current communication parameters and status of that port.

In this chapter, driver installation procedure is described. Administration utility, **moxaadm**, is explained, which is good for configuration, monitor and terminal emulation. Related issues such as device naming, baud rate settings and terminal enable are stated. Additional `Ioctl()` commands are provided for advanced programming. Finally, UNIX-specific troubleshooting is included.

7.1 MOXA UNIX Driver for C168H/C168HS

The following description is for C168H/C168HS under SCO UNIX/OpenServer and UNIX SVR4.2.

Step 1. Login the UNIX system as a super user (root).

Step 2. Change to root directory.

```
# cd /
```

Step 3. Insert the driver diskette labeled as "**Device Driver for C168H/C168HS (SCO UNIX/OpenServer + UNIX SVR4.2)**" into the floppy drive A: (or B:).

If driver files are obtained from MOXA FTP service, put them under **/tmp/moxa** directory and skip the following '**tar**' command.

Step 4. Extract the files by,

```
# tar xvf /dev/fd0135ds18 /tmp/moxa/install (if A: floppy drive)
```

Step 5. Start the installation program,

```
# /tmp/moxa/install
```

Now follow the prompted instructions to finish the driver installation.

Choose the listed and desired operation system when asked. Currently, MOXA supports SCO UNIX (and SCO OpenServer, SCO Open Desktop) and UNIX SVR4.2 (and UnixWare). Ask your dealer for newly supported device drivers. If your system is none of the listed, choose the closest one to try.

Copyright (C) 199x Moxa Technologies Co., Ltd. All Rights Reserved.

MOXA UNIX Device Driver Installation Ver. x.x

Please select one of the following OSs:

1. SCO UNIX
2. UNIX SVR4.2

Select :

Please select one of the devices where the driver diskette/files put:

- | | |
|--------------------|--------------|
| 1. /dev/fd0135ds18 | (A: 1.44MB) |
| 2. /dev/fd096ds15 | (A: 1.2 MB) |
| 3. /dev/fd1135ds18 | (B: 1.44MB) |
| 4. /dev/fd196ds15 | (B: 1.2 MB) |
| 5. Hard Disk | /tmp/moxa |

Select :

Then the **MOXA Multiport Board Installation Utility** will show for board installation.

Choose **C168 Series driver** and follow the instructions to install.

Step 6. Configure the board by,

moxaadm

The moxaadm utility is used to tell the UNIX driver how many C168 boards are installed, what their basic setting (I/O or IRQ), etc. It is also used to remove the C168 UNIX driver from the system. Whenever the setting is changed, run this utility again and tell the UNIX driver what is changed. Refer to Section "Administrations" for more information.

This utility is not intended to change the C168 "hardware" I/O address nor IRQ setting. If you need to change the C168 hardware IRQ, for example, you need to execute IO-IRQ utility under DOS environment. The IO-IRQ utility is contained in the API-232 software diskette.

Only up to two C168H/HS boards are allowed to be installed under UNIX systems when MOXA UNIX driver is used.

- Step 7.** Now you may shutdown the system and reboot. If you have trouble in booting the new UNIX kernel, use the backup kernel (**/unix.moxa** or **/stand/unix.moxa**) to boot your system and the system will work as before. Refer to Section "Troubleshooting" for more information.

After the system starts up again, the following successful messages will show:

For SCO UNIX/OpenServer,

```
"C168      0x0180-0x0187      11  Ver = x.x (high speed)"
```

For UNIX SVR4.2,

```
"ttye11/ttyE11  0x0180-0x0187      11  Ver = x.x (high speed)"
```

where high speed indicates the port is in high speed status; otherwise, in normal speed status.

If the following error message appears:

For SCO UNIX/OpenServer,

```
"C168 0x0180-0x0187      11  IRQ mismatch! (port disabled)"
```

For UNIX SVR4.2,

```
"ttye11/ttyE11 0x0180-0x0187 11  IRQ mismatch! (port disabled)"
```

indicates the IRQ of the port is different between hardware and software configurations. However, only this port is disabled. Other ports should still work.

If the following common error message appears:

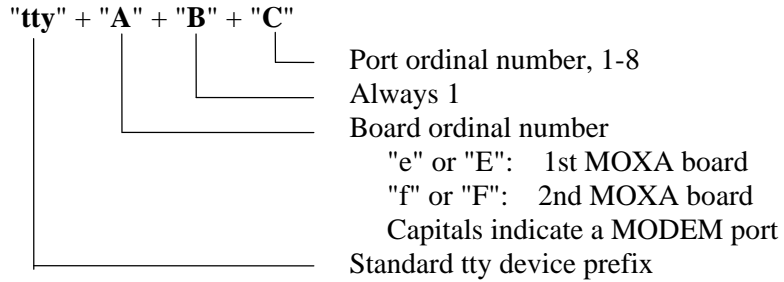
```
"WARNING! C168 board base address at 0x180 not found!" or  
"WARNING! C168 board interrupt vector disabled (Board base =  
0x180 )!", indicates the base address of the board is not found or  
interrupt vector of the board is disabled. In these cases, all ports will  
not work.
```

Please see Section "Troubleshooting" for solutions.

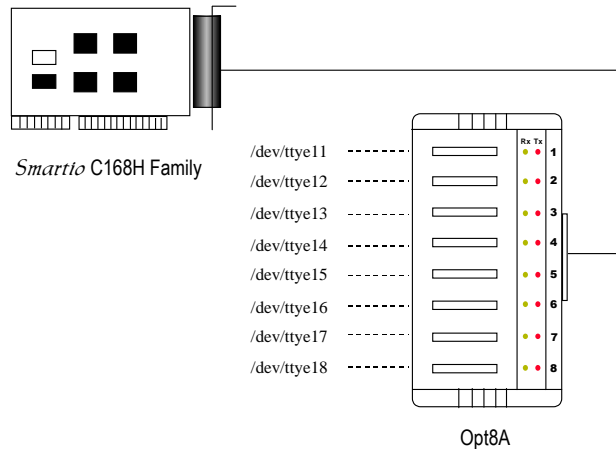
7.2 MOXA TTY Device Naming Convention

If the C168 is successfully configured, there will be two tty devices created for each port at **/dev** directory: one is non-MODEM tty (e.g. ttye11), and the other is MODEM tty (e.g. ttyE11). The two devices are actually accessing the same physical port except that the MODEM tty has to check the ON status of DCD signal to be able to open device, and closing device automatically as DCD signal is OFF.

The convention of the C168 tty device name is `/dev/tty{ef}{1}{1-8}`, where:



For example:



7.3 Baud Rate Settings

For *Smartio C168* set to **High Speed Spectrum**, the real working speed, is exactly **eight times** of the speed displayed by `"stty"` command. A port, if set to Normal Speed Spectrum with 38.4 Kbps, will work at 38.4 Kbps for sure; while a port, if set to High Speed Spectrum with displayed speed 38.4 Kbps, will actually work at 307.2 Kbps (38.4 Kbps»ë8).

Note also that the 50 baud rate, **B50**, will no longer stand for 50 bps, instead, it

means 57600 bps, and 75 baud rate, **B75**, for 115.2 Kbps. Furthermore, if the C168 Family H Series board is set to High Speed Spectrum, the real working speed is 8 times of the displayed speed. Hence, B50, 57.6 Kbps, is for 460.8 Kbps and B75, 115.2 Kbps, is for 921.6 Kbps.

7.4 Administrations

Users can use the administration utility, **moxaadm**, to change the *Smartio* C168 basic configuration, to monitor the ports' activity, to use terminal emulation and to remove the installed MOXA driver from the system.

Basic Configuration

In the main dialog, users can set base I/O address and interrupt, which should be same as the hardware configuration of the boards. **Kernel rebuilding** is required if any setting is changed.

In [Basic Port Setting] sub-dialog, there are two noticeable fields :

RTS/CTS Hardware Flow Control:

This feature is for more flexible driver behavior. If set to OFF (default), CTS signal is not needed for tty port to transfer data and RTS/CTS hardware flow control function in driver is disabled. On contrary, if set to ON, CTS signal is needed for tty port to transfer data and RTS/CTS hardware flow control function in driver is enabled. **Kernel rebuilding** is required if this setting is changed.

Non-Modem (Modem) Baud Rate:

This field stands for the initial baud rate symbol and hunt sequence for Non-Modem (Modem) tty and is simply for setting parameters of getty entries in system file **/etc/inittab** which could also be manually modified by system administrator. Its value comes from the UNIX system "getty default" file, e.g., **/etc/gettydefs**. Modify this field to suit your need.

In some cases, you may need to modify the "getty default" file as well. For

example, in some UNIXs, the "9600" symbol indicated 8-data-bit no-parity while in others it may stand for 7-data-bit, even-parity. So, please examine the "getty default" file carefully, and make sure the terminal settings is the same. Otherwise, garbled data will be inevitable.

Port Monitoring

This utility gives the user a quick view about all the MOXA ports' activities. One can easily learn each port's total received/transmitted (Rx/Tx) character count since the time when the monitoring is started. Rx/Tx throughputs per second are also reported in interval basis (e.g. the last 5 seconds) and in average basis (since the time the monitoring is started). You can reset all ports' count by <HOME> key. <+> <-> (plus/minus) keys to change the displaying time interval. Press <ENTER> on the port, that cursor stay, to view the port's communication parameters, signal status and input/output queue.

Terminal Emulation

This utility provides data sending and receiving capability of all tty ports, especially for MOXA ports. It is quite useful for testing simple application, for example, sending AT command to a modem connected to the port or used as a terminal for login purpose. Note that this is only a dumb terminal emulation without handling full screen operation. Besides, pattern/file transmission is provided.

Driver Removal

If you want to remove the *Smartio* C168 device driver and return to your system configuration, just press <ENTER> in this function entry and answer "y" to confirm. Then the system will be rebuild. This may take some time. If you answer "n", no action will be taken.

7.5 Setting MOXA Ports to Terminal

To enable the terminal port for login purpose, taking ttye11 as an example,

for SCO UNIX/OpenServer

```
# enable /dev/ttye11
```

for UNIX SVR4.2

1. Edit (e.g. use vi editor) the file /etc/initab.
2. Modify the tty entry from "me11:23:off:/etc/getty ttye11 9600" to "me11:23:respawn:/etc/getty ttye11 9600".
3. # init q

Or refer to your UNIX system manuals for how to activate a tty port.

7.6 Programming the MOXA Ports

The system services that apply to standard tty port can also be applied to MOXA port since MOXA port conforms to UNIX tty standard. For example: open(), ioctl(), read(), write(), close(), etc. Please refer to your UNIX Programmer's Reference manual.

However, these system services only provide a limited functions and thus may not satisfy the sophisticated programmers' need. In order to meet the user's special purpose, MOXA supports extended services through ioctl() command.

Here is the list of the functions.

1. MTCRTS (= 0x403) To control RTS output signal.
2. MTCDTR (= 0x404) To control DTR output signal.
3. MSTATUS (= 0x407) To read modem line status (CTS/DSR/DCD.)
4. MHWFLOW (= 0x40e) To enable/disable the hardware flow control.
5. HighSpeedOn (= 0x43d) To check the high speed status for a port

For details, please refer to Appendix " Extended UNIX Ioctl() Commands".

7.7 Troubleshooting

This section is specific for UNIX. For general problems and solutions please see Appendix A.

1. **Under UNIX, when respawning quite a few number of tty ports, the following error messages appear: "Time out table overflow", "File table overflow", "Region table overflow".**

Software Cause and Solution:

The above error messages imply that the system resources are exhausted. You should tune the kernel parameters to a larger value and rebuild the kernel to be able to accommodate the new configuration. Refer to UNIX system manual about how to tune the parameters and rebuild kernel.

"Time out table overflow" : NCALL parameter too small.

"File table overflow" : NFILE or NINODE parameter too small.

"Region table overflow" : NREGION or NPROC parameter too small.

2. **Under UNIXs, the newly rebuilt kernel could not boot.**

Software Causes and Solutions:

The C168 driver might not be built into the new kernel correctly.

- a. Please use the last good kernel backup to boot again. The kernel backup in SCO UNIX is /unix.moxa while in UNIX SVR4.2 /stand/unix.moxa.
- b. Then remove the C168 driver. Refer to Section 7.4 for driver removal.
- c. Re-install the C168 driver once more.

The *Smartio* C104/C168 Family is useful for office automation and industrial control as well as data acquisition and communication. Because many users need to implement an industrial control system or data acquisition system in the DOS environment, but DOS did not have any better communication API, thus Moxa has developed device drivers and library functions under DOS. Furthermore, Data Scope utility is very helpful for troubleshooting and debugging the serial communications.

In this chapter, driver installation, setup, loading and removal procedure are described. Utility, **Data Scope**, is explained, which is good for terminal emulation, data scope and diagnostics. Related issues such as device naming and programming are also stated.

Moxa device driver is a port-based one and supports maximum of 8 serial ports. Settings of each port can be set independently.

8.1 Driver Installation, Setup, Loading and Unloading

MOXA provides a software package that assists users to develop and/or debug programs for serial communications under DOS. It supports device drivers for the MOXA boards such as *Smartio* C104/C168 Family, C218/C320 Series and standard PC COM ports, etc.

Driver Installation

Run the installation program, **DOSINST.EXE**, in the DOS/Windows 3.x (API-232) software diskette. Specify the target directory (e.g. **C:\MOXA**) where software driver and library will be copied to. Press **F2** to start the installation.

Driver Setup

1. Change to target directory and run the setup program, **BIN\SETUP.EXE**.

This program is to configure IRQ, port number, etc., of the MOXA boards. The configuration will be referred to by the later executed DOS driver. Note that **before running SETUP program, execute Io-irq to designate I/O address and IRQ.**

2. Choose "**C102/C104/C168 Series Multiport Board**" in the "DRIVER SELECTION" dialog and then enter or modify each port's configuration which are the port initial value when driver is loaded.
3. Use **F8** function to load the on-board configuration for *Smartio* C104/C168 Family. Use other functions to modify settings if necessary. Finally remember to use **F10** function to save the configuration.

Some noticeable functions keys and fields are explained below.

Press **F1**: "Help" to learn an up-to-date instructions.

Press **F3**: "Add port" to add one port entry and modify the settings if necessary.

Press **F4**: "Delete port" to delete the port entry where the cursor stops.

Press **F5**: "Group edit" to set the selected ports (as a group) to the same settings.

Press **F6**: "INT vector" to specify the interrupt vector I/O address and set the status for each port. Interrupt-driven scheme is applied for those ports using interrupt vector while polling scheme for those ports not using interrupt vector. The former is more efficient and has better performance in terms of driver. In addition, for two C104 boards in a system, two separate interrupt vectors are required which can be accomplished with **F3: Toggle one/two INT vector.**

Press **F8**: "Load config" to load the hardware configuration set by Io-

irq.exe. This is the most convenient way of configuration, comparing to add port one by one with function F3.

- Port number** : This is actually the port ID of each port. The application software will refer to the port by its port number (ID). Duplicated port number is not allowed.
- Base I/O address** : The I/O address of each port. Overlapped or duplicated I/O address is not allowed.
- Interrupt number** : The IRQ number of each port. Several ports may share one common IRQ.
- TxD buffer size** : The transmit (output) buffer reserved for the port.
- RxD buffer size** : The receive (input) buffer reserved for the port.

Driver Loading

Having completed the setup, load the driver, **BIN\SER-DRV.EXE**, at the DOS prompt. The driver will detect the multiport board automatically. If the board(s) is(are) detected, a message similar to below will show:

```
Universal 2/4/8 Serial Port Communication Driver (Ver x.x)
Device Driver Setup O.K.
```

Which means the board and the driver have been successfully installed. At this point, user is ready to execute application that supports API functions, or start developing applications.

If there is no matched port, the screen will show a message like:

```
Universal 2/4/8 Serial Port Communication Driver (Ver x.x)
No Serial Port Found!!
```

Please refer to Appendix A "Troubleshooting" for possible reasons and

solutions.

To support various types of UARTs such as 16450, 16550 and TI550C, several options for SER-DRV are provided. Type **SER-DRV/?** to see the available options.

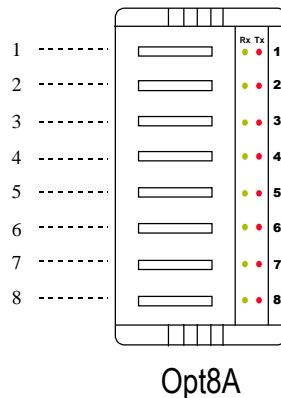
Driver Unloading

To unload the driver from memory, type **SER-DRV/Q** at the DOS prompt.

8.2 MOXA Serial Port Naming Convention

Each MOXA serial port are referenced as port number in terms of programming. The port numbers are automatically assigned once the starting port number is decided by user when configuring the ports of the board. Duplication of the port numbers is not allowed.

For example, if 8 ports are configured and the starting port number is 1, then the mapping of serial port numbers will be as depicted.



8.3 Baud Rate Settings

Be aware that, for *Smartio* boards configured as **High Speed Spectrum**, the real working speed is equal to **8 times** of the displayed speed. For example, a

port, if set to Normal Speed Spectrum with 38.4 Kbps, will work at 38.4 Kbps for sure; while a port, if set to High Speed Spectrum with shown speed 38.4 Kbps, will actually work at 307.2 Kbps (38.4 Kbps \times 8). This is applicable to Moxa-provided driver and utility, such as Setup and Data Scope, and also any non Moxa-provided driver and utility, such as **Telix**.

8.4 DOS API-232 Library and Programming

For DOS programming, API-232 includes powerful libraries supporting languages like Microsoft C, Turbo C, Macro Assembly, QuickBasic, Turbo Pascal, Clipper, etc. Sample programs for each supported language are included, and placed in the sub-directory `..\EXAMPLE\language` of the API-232 directory.

In addition, for DOS C language only, there are also Modem Control and File Transfer library available, supporting Hayes compatible modem control as well as ASCII, KERMIT, XMODEM, YMODEM and ZMODEM file transfer protocol functions.

For complete and last-minute API function description, see file **API-232.TXT** in the API-232 directory for more details.

8.5 Data Scope

The Data Scope, **BIN\SCOPE.EXE**, is a suite of utility programs that can help users with system troubleshooting and serial communication debugging. There are three major functions:

1. The Data Scope utility offers transparent monitoring of serial communication lines and allows data to be streamed to disk storage for later analysis.
2. The TTY terminal emulation utility allows user to view the signal status and

transfer data interactively or files using ASCII, XMODEM, YMODEM, ZMODEM and KERMIT protocols.

3. The diagnostic test utility provides port connection test with two MOXA ports connected via a properly-wired cable.

Please see on-line help as running BIN\SCOPE.EXE for more usage and capability information.

Chapter 9

For Windows 3.x Users

This chapter is for *Smartio C104 Family only*. *Smartio C168 Family* does not support Windows 3.x. **MOXA API-232** is a software package that assists users to develop and/or debug programs for serial communications under Windows 3.x (and also DOS).

In this chapter, driver installation and configuration procedure are described. Utility, **TTY**, is explained, which is good for terminal emulation. Related issues such as driver removal, baud rate settings, programming and existing applications are also stated.

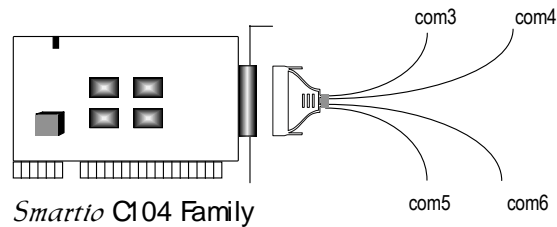
Installation

To install, run **WININST.EXE** in the DOS/Windows 3.x driver (API-232) diskette, select **COMM Driver** and **C104**, and then do configuration as prompted.

Configuration

The Windows-based configuration program, **CONFIG**, has the easiest way to configure the 4 ports of C104, either from COM1 to COM4, from COM2 to COM5, from COM3 to COM6, from COM4 to COM7, from COM5 to COM8 or from COM6 to COM9, depending on user's need. Normally COM1 is used by mouse and to fully use the 4 MOXA ports, thus COM3 to COM6 is recommended. In this case, the original COM1 and COM2 on PC will be still available.

For example, if 4 ports are configured and starting from COM3, then the mapping of serial port numbers will be as depicted.



Due to the limitations of Windows 3.x operating system itself, only up to 9 COM ports are supported, i.e., COM1 to COM9. Hence, C104 with 4 ports or other 4 port non-intelligent boards, maximum 6 ports is supported if the existing standard COM ports (COM1 and COM2) are included.

After successful installation, the **MOXA program group** will be created which contains all the useful programs Moxa provided. **Restarting Windows 3.x system** is required for MOXA Standard Windows-compatible COMM Driver after installation. When system restarted, the MOXA COMM Driver is ready to go.

Driver Removal

The program, **Remove**, in the **MOXA program group** is provided to remove the installed driver from the Windows.

Baud Rate Settings

For those *Smartio C104* boards configured as **High Speed Spectrum**, the real working speed is equal to **8 times** of the displayed speed. For example, a port, if set to Normal Speed Spectrum with 38.4 Kbps, will work at 38.4 Kbps for sure; while a port, if set to High Speed Spectrum with shown speed 38.4 Kbps, will actually work at 307.2 Kbps (38.4 Kbps \times 8). This is applicable to Moxa-provided utility, such as CONFIG and TTY, existing applications and programming which are described later in this chapter.

Utility

The utility, **TTY**, in the **MOXA program group** is intended to help users monitor and debug RS-232 communications under Windows 3.x which can manipulate ports from COM1 to COM9. It is just a simplest example program which can send and receive data after each port opened with selected communication parameters. Multiple windows for ports simultaneously are available for a demonstration of multitasking feature of Windows 3.x. The Windows-provided application, **Terminal**, can only make use of COM1 to COM4 which is obviously a restriction.

Windows 3.x COMM API Programming

MOXA Windows-compatible COMM Driver supports **Microsoft Windows COMM API** such as `OpenComm()`, `ReadComm()`, `WriteComm()`, etc. It supports any language conforming to the Windows COMM API like Microsoft C, Borland C, Visual C, Visual Basic, Delphi, etc. Sample programs for only Microsoft C, Borland C and Visual Basic are supported. For other languages' sample programs, please refer to the language-provided communication example programs.

Existing Applications

Many Windows software packages, such as pcANYWHERE, LabView, FIX, WinFax Pro, Fax Server, PROCOMM PLUS, LapLink, etc. can access MOXA C104 COM ports directly since these applications follow the Microsoft Windows COMM API.

Common *Smartio* C104/C168 Family problems and possible solutions are listed below. If you still have problems, contact your dealer or Moxa for help. Or report it using "**Problem Report Form**" at the end of this manual to your dealer at once for faster technical support.

1. MOXA board not found.

Hardware causes and solutions:

- a. The base I/O address you have selected for the board conflicts with that of other interface boards such as SCSI, LAN or Sound boards. If so, adjust the address to forestall the conflict.
- b. The board is not properly plugged in the system. If that is the case, then re-install the board and make sure that it fits well in a 16-bit slot this time. Sometimes the slot for plugging board is bad. In this case, please try other slots until a good one is found.
- c. The board is not installed or missing (absent). Please install it.
- d. The board is out of order. Please check the board(s) ONE AT A TIME in the system and find out the defective one. It is also recommended to check it under DOS with F3 Diagnostics function of utility "SETUP" to verify the hardware conditions.

Software Causes and Solutions:

- a. The base I/O address settings on board is not the same as the settings in configuration software.

2. MOXA board found but can not transfer (transmitting/receiving) data.

Hardware Causes and Solutions:

- a. Check if wrong cable wiring. Refer to Appendix C for precise pinouts of the connector type you are using.
- b. Cable or external box is defective.

- c. IRQ is used or reserved by PCI bus.

Software Causes and Solutions :

- a. *Smartio* C104/C168 Family will check line status (CTS) before it sends data out if RTS/CTS flow control feature is set to ON in configuration or application program. Please see Appendix C for proper wiring.
- b. Perhaps the application controlling the board is not correctly written according to the corresponding API of the operating system. To verify, please run existing and known good application or Moxa-provided utility. For example, under Windows 95, "Hyperterminal" is good for testing COM ports. Under UNIX, "stty", "cat", or "getty" is good for testing tty ports.
- c. IRQ settings of software and hardware do not match.

3. Why the DOS utility IQ-IRQ can not access *Smartio* C104/C168 to configure?

There are several reasons that may lead to this trouble:

- a. The user forgets or does not know the Configuration Access Port (CAP) of the board. See next problem 4 for how to solve this problem.
- b. The CAP of the board conflicts with other add-on boards' I/O address. Please change other add-on boards' I/O address to avoid the conflict.
- c. The *Smartio* C104/C168 board is not plugged in a right or good slot. Please plug the board in a good 16-bit ISA slot.
- d. The *Smartio* C104/C168 board may malfunction. Please return for repair.

If any existing board has the same I/O address as 0x180, the **default** CAP address, or the 1st port's I/O address, you must try to avoid the conflict by doing either one the following.

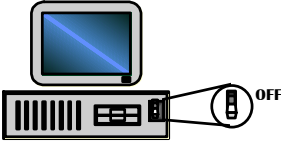
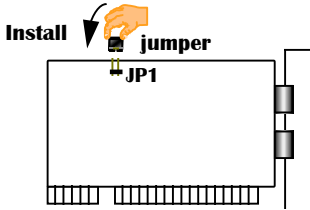
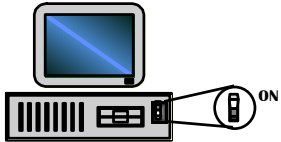
- a. Install a jumper (short) at position JP1 on the upper-left corner of the board. This will force the CAP address to 0xA700.
- b. Change or disable the existing board's I/O address.

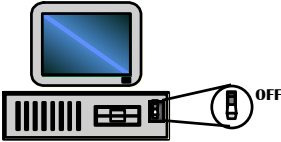
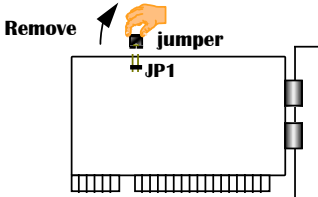
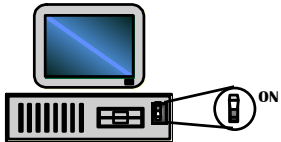
4. What to do if user forgets or does not know the Configuration Access Port (CAP) of C104/C168?

The *Smartio* C104/C168 Family multiport boards are designed without jumper or switch, so the configuration is completed only by DOS utility Io-irq.exe.

To configure the board, you need to know the board's Configuration Access Port (CAP) address. Because the CAP address is the only channel via which the Io-irq.exe can access to the board.

The following procedure instructs user to recover once the CAP is unknown.

<p>Step 1.</p>	<p>Power off the PC.</p>	
<p>Step 2.</p>	<p>Install jumper onto the JP1 of the board.</p>	
<p>Step 3.</p>	<p>Power on the PC. Now the CAP address of the board will be 0xA700.</p>	
<p>Step 4.</p>	<p>Execute Io-irq under DOS environment.</p> <p style="text-align: center;">A:> Io-irq</p>	

Step 5.	Enter CAP address 0xA700 to access the board. <i>Enter the "Configuration Access Port" in HEX: A700</i>	
Step 6.	The previous hardware configuration will be shown. <i>Modify them if necessary. Remember the CAP address this time.</i>	
Step 7.	Exit the IO-IRQ.	
Step 8.	Power off the PC.	
Step 9.	Remove the jumper on position JP1.	
Step 10.	Power on the PC.	

Appendix B C104 Pinouts and Cable Wiring

In data communications, the term **DTE** is Data Terminal Equipment like terminal or PC COM1/2. The term **DCE** is Data Communication Equipment like modem. Please check the precise pinouts, the following pinouts are typical examples.

B.1 RS-232 Cable Wiring

The following lists pin assignments of each of four **DB25** RS-232 ports

Pin no.	Signal
2	TxD
3	RxD
4	RTS
5	CTS
6	DSR
8	DCD
20	DTR
7	GND

Type 1: To connect *Smartio C104* to a DTE.

<i>Smartio C104</i> DB25	DTE DB25 Male	<i>Smartio C104</i> DB25	DTE DB9 Male
TxD	3 RxD	TxD	2 RxD
RxD	2 TxD	RxD	3 TxD
RTS	5 CTS	RTS	8 CTS
CTS	4 RTS	CTS	7 RTS
DSR	20 DTR	DSR	4 DTR
GND	7 GND	GND	5 GND
DTR	6 DSR	DTR	6 DSR
DCD	8 DCD	DCD	1 DCD

Type 2: To connect *Smartio* C104 to a DCE.

<i>Smartio</i> C104 DB25	DCE DB25 Female
TxD	2 RxD
RxD	3 TxD
RTS	4 CTS
CTS	5 RTS
DSR	6 DTR
GND	7 GND
DTR	20 DSR
DCD	8 DCD

Type 3: To connect *Smartio* C104 to a DTE with 3-pin wiring.

If [Hardware flow control] feature is set to "ON", user must loop back (or short) the RTS with CTS and DSR with DTR, DCD on MOXA site.

<i>Smartio</i> C104 DB25	DTE DB25 Male	<i>Smartio</i> C104 DB25	DTE DB9 Male
TxD	3 RxD	TxD	2 RxD
RxD	2 TxD	RxD	3 TxD
GND	7 GND	GND	5 GND
RTS		RTS	
CTS		CTS	
DSR		DSR	
DTR		DTR	
DCD		DCD	

If [Hardware flow control] feature is set to "OFF", users could just leave RTS, CTS, DSR, DTR, DCD open.

<i>Smartio</i> C104 DB25	DTE DB25 Male	<i>Smartio</i> C104 DB25	DTE DB9 Male
TxD	3 RxD	TxD	2 RxD
RxD	2 TxD	RxD	3 TxD
GND	7 GND	GND	5 GND
RTS		RTS	
CTS		CTS	
DSR		DSR	
DTR		DTR	
DCD		DCD	

B.2 DB37 Connector Pinouts

The following lists pin assignment of the *Smartio* C104 Family **DB37** connector on the bracket.

Pin no.	Signal	Pin no.	Signal
1		20	RI3
2	DCD3	21	DTR3
3	GND	22	DSR3
4	CTS3	23	RTS3
5	RXD3	24	TXD3
6	RI4	25	DCD4
7	DTR4	26	GND
8	DSR4	27	CTS4
9	RTS4	28	RXD4
10	TXD4	29	RI2
11	DCD2	30	DTR2
12	GND	31	DSR2
13	CTS2	32	RTS2
14	RXD2	33	TXD2
15	RI1	34	DCD1
16	DTR1	35	GND
17	DSR1	36	CTS1
18	RTS1	37	RXD1
19	TxD1		

Note: make shield grounded to connector.

Appendix C C168 Pinouts and Cable Wiring

In data communications, the term **DTE** is Data Terminal Equipment like terminal or PC COM1/2. The term **DCE** is Data Communication Equipment like modem. Please check the precise pinouts, the following pinouts are typical examples.

C.1 RS-232 Cable Wiring

RS-232 connection boxes or octopus cable designed for *Smartio* C168 are:

Opt8A : Connection box with 8 female RS-232 DB25 ports.

Opt8B : Connection box with 8 male RS-232 DB25 ports.

Opt8C : Octopus cable with 8 male RS-232 DB25 ports.

Opt8D : Octopus cable with 8 male RS-232 DB9 ports.

Opt8S : Connection box with 8 female RS-232 DB25 ports and 2000V surge protection.

The following lists the port pin assignments:

Opt8A/S (DCE, DB25 Female)	Opt8B/C (DTE, DB25 Male)	Opt8D (DTE, DB9 Male)
3 TxD	2 TxD	2 RxD
2 RxD	3 RxD	3 TxD
5 RTS	4 RTS	4 DTR
4 CTS	5 CTS	5 GND
20 DSR	6 DSR	6 DSR
6 DTR	20 DTR	7 RTS
7 GND	7 GND	8 CTS
8 DCD	8 DCD	1 DCD

Type 1: To connect *Smartio C168* to a DTE.

<i>Smartio C168</i> DB25	DTE DB25 Male	<i>Smartio C168</i> DB25	DTE DB9 Male
TxD _____	3 RxD	TxD _____	2 RxD
RxD _____	2 TxD	RxD _____	3 TxD
RTS _____	5 CTS	RTS _____	8 CTS
CTS _____	4 RTS	CTS _____	7 RTS
DSR _____	20 DTR	DSR _____	4 DTR
GND _____	7 GND	GND _____	5 GND
DTR _____	6 DSR	DTR _____	6 DSR
DCD _____	8 DCD	DCD _____	1 DCD

Type 2: To connect *Smartio C168* to a DCE.

<i>Smartio C168</i> DB25	DCE DB25 Female
TxD _____	2 RxD
RxD _____	3 TxD
RTS _____	4 CTS
CTS _____	5 RTS
DSR _____	6 DTR
GND _____	7 GND
DTR _____	20 DSR
DCD _____	8 DCD

Type 3: To connect *Smartio C168* to a DTE with 3-pin wiring.

If [Hardware flow control] feature is set to "ON", user must loop back (or short) the RTS with CTS and DSR with DTR, DCD on MOXA site.

<i>Smartio C168</i> DB25	DTE DB25 Male	<i>Smartio C168</i> DB25	DTE DB9 Male
TxD _____	3 RxD	TxD _____	2 RxD
RxD _____	2 TxD	RxD _____	3 TxD
GND _____	7 GND	GND _____	5 GND
RTS _____		RTS _____	
CTS _____		CTS _____	
DSR _____		DSR _____	
DTR _____		DTR _____	
DCD _____		DCD _____	

If [Hardware flow control] feature is set to "OFF", users could just leave RTS, CTS, DSR, DTR, DCD open.

<i>Smartio C168</i> DB25	_____	DTE DB25 Male	<i>Smartio C168</i> DB25	_____	DTE DB9 Male
TxD	_____	3 RxD	TxD	_____	2 RxD
RxD	_____	2 TxD	RxD	_____	3 TxD
GND	_____	7 GND	GND	_____	5 GND
RTS			RTS		
CTS			CTS		
DSR			DSR		
DTR			DTR		
DCD			DCD		

C.2 RS-422 Cable Wiring

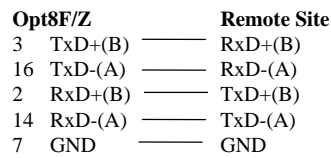
RS-422 connection boxes have been designed for *Smartio C168*, which are:

- Opt8J:** Connection box with 8 female RS-422/485 DB25 ports.
- Opt8F:** Connection box with 8 female RS-422 DB25 ports and Max. 500V DC isolation protection which can prevent damage caused by high potential voltage.
- Opt8Z:** Connection box with 8 female RS-422 DB25 ports but without isolation protection.

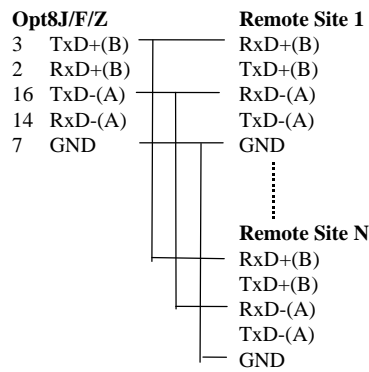
The RS-422/485 transmission distance can reach as long as 4000ft. at 100K bps baud. The connection box needs an external power adapter to supply 5V DC power. Either 110V or 220V AC power adapter is selectable.

The followings are operation modes for RS-422:

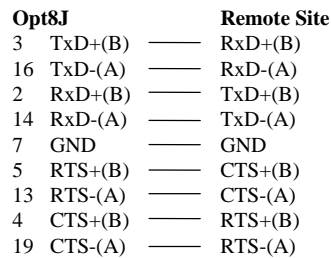
RS-422 Point-to-point



RS-422 Broadcasting



RS-422 Point-to-point full-duplex



For **Opt8J RS-422** communications, do set the [Tx, Rx] port switch on the side panel to [Off, Off] position and then use point-to-point full-duplex or broadcasting operation.

C.3 RS-485 Cable Wiring

RS-485 connection box has been designed for C168P, which is:

Opt8J: Connection box with 8 female RS-422/485 DB25 ports.

The RS-485 transmission distance can reach as long as 4000ft. The connection

box needs an external power adapter to supply 5V DC power. Either 110V or 220V AC power adapter is selectable.

The RS-485 standard is an enhanced version of the RS-422A balanced line standard. It allows multiple drivers and receivers in a multidrop systems. As many as 32 drivers and 32 receivers can be put on any multidrop system.

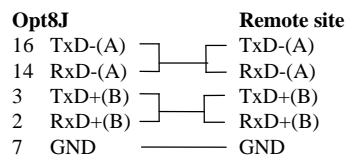
There are basically two operations for RS-485, **point-to-point** and **multidrop**. And each operation has **half** and **full-duplex** options. For half-duplex, there are still **echo** and **non-echo** modes. Since TxD+(B) and RxD+(B) are wired together (see pictures below), whatever data sent from TxD+(B) will be received by RxD+(B) of the local site and, of course, the RxD+(B) of the other site. If the local RxD+(B) is enabled to receive data, it is called **echo** mode. Otherwise, the local RxD+(B) is disabled and it is called **non-echo** mode. If the port switch is set to [On, On], only the non-echo mode is possible. See Appendix D "RS-485 Programming" for details.

The followings are operation modes for RS-485:

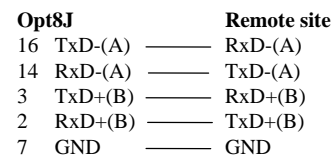
Point-to-point

Point-to-point communications means two devices located at two different places can be linked together to communicate via RS-485 interface. Depending on the wiring, half-duplex or full-duplex can be used.

Point-to-point RS-485 Half-duplex



Point-to-point RS-485 Full-duplex



In accordance with the point-to-point operation shown above, there are a couple of setting combinations for Tx and Rx:

1. Point-to-point full-duplex operation

Set [Tx, Rx] port switch to [Off, Off].

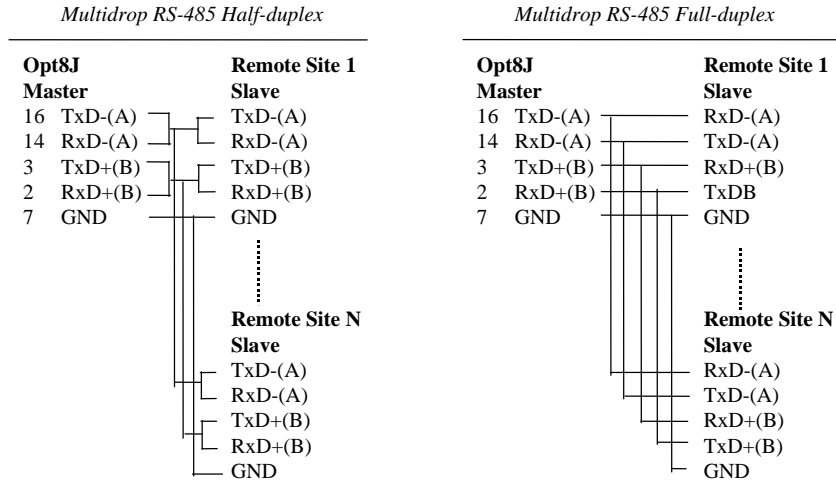
2. Point-to-point half-duplex operation

For echo mode, set [Tx, Rx] port switch to [On, Off]. For non-echo mode, set [Tx, Rx] port switch to [On, On].

Multidrop

Multidrop communication means that more than two devices can be linked together to communicate with one another via RS-485 interface. In this operation, one of the device serves as **master** device while the rest of the devices as **slaves**. Normally, master will send data to individual slave and the slave will receive the data according to its ID (or any site-specific identification) embedded in the data. This is how master communicate with each slave. However, **broadcasting** is also possible if the master sends data without any ID to all the slaves.

Depending on the wiring, half-duplex or full-duplex can be used.



In accordance with the multidrop operations shown above, there are a couple of setting combinations for Tx and Rx:

1. Multidrop full-duplex operation

Set [Tx, Rx] port switch of the master to [Off, Off]. Set [Tx, Rx] port switch of the slaves to [On, Off].

2. Multidrop half-duplex operation

For echo mode, set [Tx, Rx] port switch of the master and slaves to [On, Off].
For non-echo mode, set [Tx, Rx] port switch to [On, On].

C.4 RS-422/485 Impedance Matching

For RS-422/485 serial communications, when an electrical signal travels through two different resistance junctions in a transmission line, the mismatch will sometimes cause signal reflection. Signal reflection causes signal distortion, which in turn will contribute communication errors. The solution to this problem is to establish the same impedance at the line ends as in the line itself by terminating them with resistors.

The value of the termination resistor should equal the characteristic impedance of the transmission line. The resistors should be added near the receiving side.

Opt8J/F/Z			Remote Site	
3	TxD+(B)	_____	RxD+(B)	} Opt8J only
16	TxD-(A)	_____ξ	RxD-(A)	
2	RxD+(B)	_____ξ	TxD+(B)	
14	RxD-(A)	_____ξ	TxD-(A)	
5	RTS+(B)	_____ξ	CTS+(B)	
13	RTS-(A)	_____ξ	CTS-(A)	
4	CTS+(B)	_____ξ	RTS+(B)	
19	CTS-(A)	_____ξ	RTS-(A)	

Note:

1. ξ stands for termination resistor near the receiving side
2. The suggested termination resistor for AWG #26 cable is 100 ohm.
3. The suggested termination resistor for phone cable is 600 ohm.

C.5 DB62 Connector Pinouts

The following lists the pin assignments of the **DB62** connector on the bracket.

Pin no.	Signal	Pin no.	Signal	Pin no.	Signal
1	TxD1	22	RxD1	43	CTS1
2	DTR1	23	DSR1	44	RTS1
		24	DCD1	45	GND
3	RxD2	25	TxD2	46	CTS2
4	DSR2	26	DTR2	47	RTS2
5	DCD2				
6	TxD3	27	RxD3	48	CTS3
7	DTR3	28	DSR3	49	RTS3
		29	DCD3	50	GND
8	RxD4	30	TxD4	51	CTS4
9	DSR4	31	DTR4	52	RTS4
10	DCD4	32	GND		
11	RxD5	33	TxD5	53	CTS5
12	DSR5	34	DTR5	54	RTS5
13	DCD5			55	GND
14	TxD6	35	RxD6	56	CTS6
15	DTR6	36	DSR6	57	RTS6
		37	DCD6	58	GND
16	RxD7	38	TxD7	59	CTS7
17	DSR7	39	DTR7	60	RTS7
18	DCD7	40	GND		
19	RxD8	41	TxD8	61	CTS8
20	DSR8	42	DTR8	62	RTS8
21	DCD8				

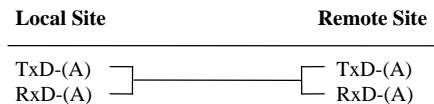
Appendix D

RS-485 Programming

If you are using RS-485 communication, such as Opt8J or A50/A51, please follow the RS-485 programming guide below and also refer to Appendix C.3 for more RS-485 operation details.

For RS-485 ports, each has one receiver (Rx) and one driver (Tx). Both driver and receiver can be either enabled or disabled in order to achieve either **point-to-point half/full-duplex** communication or **multidrop half/full-duplex** communication.

The principle for **RS-485 full-duplex programming** is same as that for RS-232/422 programming. However, for **RS-485 half-duplex programming**, remember that only one end of the signal line can be enabled when Tx and Rx of both ends are physically wired together. For example,



When the TxD-(A) of the local site is enabled to transmit data, the TxD-(A) of the remote site should be disabled and RxD-(A) of remote site should be enabled to receive data. Meanwhile, the RxD-(A) of the local site can be either enabled to receive data from TxD-(A) of the local site, which is called **echo** mode, or disabled, which is called **non-echo** mode.

In a **half-duplex** environment, **contention** problems (or **data collision**) will occur if two devices attempt to transmit data simultaneously. For example, if the TxD-(A) of both the local and the remote sites are enabled at the same time, the data sent will be garbled due to the contention problems. However, if the TxD-(A) of both the local site and the remote site are disabled, the line will be floating.

To control the enable/disable mode of the driver and receiver, there are 2 options: "**Always On**" and "**By RTS**" for the driver and 2 options: "**Always On**" and "**By /RTS**" for the receiver to be chosen.

1. Always On

If the **Tx** and **Rx** port switch is set to **Off** position ("**Always On**"), the driver/receiver is always enabled and ready for transmitting/receiving data.

2. Tx By RTS or Rx By /RTS

The **Tx** or **Rx** port switch of each port can be set to **On** position ("**By RTS**" for Tx or "**By /RTS**" for Rx). That is, the driver and receiver are controlled independently.

The driver is enabled if it is set to "**By RTS**" and the RTS signal is on. The driver is disabled if it is set to "**By RTS**" and the RTS signal is off.

The receiver is disabled if it is set to "**By /RTS**" and the RTS signal is on. The receiver is enabled if it is set to "**By /RTS**" and the RTS signal is off.

"**By RTS**" or "**By /RTS**" is suitable for any system that allows RTS control from application programs. Any one of the following statements will set RTS signal true and turn on the driver or turn off the receiver:

```
sio_RTS(port, 1);          /* turn on RTS */
or
sio_putb_x();             /* turn on RTS before data sent */
```

Any one of the following statements will clear the RTS signal and turn off the driver or turn on the receiver:

```
sio_RTS(port, 0);        /* turn off RTS */
or
sio_putb_x();            /* turn off RTS after data sent */
```

For details about API-232 function, **sio_RTS()** and **sio_putb_x()**, please

refer to the *PComm* Library on-line **Help** file and RS-485 example C program in the *PComm* diskette.

Appendix E Extended UNIX ioctl () Commands

The following lists the syntax of MOXA extended functions for both non-SVR4.x and SVR4.x UNIX. The variable `moxa_fd` is the returned file descriptor by `open()` a specific MOXA port. For example,

```
int moxa_fd;
moxa_fd = open("/dev/ttya11",O_RDWR);
```

1. MTCRTS

This function, only valid when hardware flow control is turned off (see MHWFLOW), is used to drive RTS on or off.

Syntax for SCO UNIX/OpenServer

```
#define MTCRTS          0x403
#define TurnON          1
#define TurnOFF         0

ioctl(moxa_fd, MTCRTS, TurnON);
ioctl(moxa_fd, MTCRTS, TurnOFF);
```

Syntax for UNIX SVR4.2

```
#include <sys/stropts.h>
#include <sys/sysmacros.h>
#define MTCRTS          0x403
#define TurnON          1
#define TurnOFF         0
struct strioctl
int                    ioc;
int                    setting;

setting = TurnON; /* or TurnOFF */
ioc.ic_cmd = MTCRTS;
ioc.ic_timout = 0;
ioc.ic_len = sizeof(int);
ioc.ic_dp = (char *)&setting;
ioctl(moxa_fd, I_STR, &ioc);
```

2. MTC DTR

This function, only valid when hardware flow control is turned off (see MHWFLOW), is used to DTR on or off.

Syntax for SCO UNIX/OpenServer

```
#define MTC DTR      0x404
#define TurnON      1
#define TurnOFF     0

ioctl(moxa_fd, MTC DTR, TurnON);
ioctl(moxa_fd, MTC DTR, TurnOFF);
```

Syntax for UNIX SVR4.2

```
#include <sys/stropts.h>
#include <sys/sysmacros.h>
#define MTC DTR      0x404
#define TurnON      1
#define TurnOFF     0
struct strioc l
int
setting;

setting = TurnON; /* or TurnOFF */
ioc.ic_cmd = MTC DTR;
ioc.ic_timeout = 0;
ioc.ic_len = sizeof(int);
ioc.ic_dp = (char *)&setting;
ioctl(moxa_fd, I_STR, &ioc);
```

3. MSTATUS

This function is used to know the RS-232 line status (CTS/DSR/DCD).

Syntax for SCO UNIX/OpenServer

```
#define MSTATUS      0x407
int
status; /* status = RS-232 line status */
/* bit0»éCTS (1:on, 0:off)*/
/* bit1»éDSR (1:on, 0:off) */
/* bit2»éDCD (1:on, 0:off) */

ioctl(moxa_fd, MSTATUS, &status);
```

Syntax for UNIX SVR4.2


```

#define MSTATUS          0x407
#include                <sys/stropts.h>
#include                <sys/sysmacros.h>

struct strioc          ioc;
int                    status; /* status = RS-232 line status */
                        /* bit0»CTS (1:on, 0:off) */
                        /* bit1»DSR (1:on, 0:off) */
                        /* bit2»DCD (1:on, 0:off) */

ioc.ic_cmd = MSTATUS;
ioc.ic_timeout = 0;
ioc.ic_len = sizeof(int);
ioc.ic_dp = (char *)&status;
ioctl(moxa_fd, I_STR, &ioc);

```

4. MHWFLOW

This function is used to enable/disable hardware flow control. The first open() function of a port will set the hardware flow control bits on or off (RTSCTSFlowControlBits) depending on the [RTS/CTS Hardware Flow Control] configuration in **moxaadm**. However, users might want to control the DTR or RTS signal on their will, thus the hardware flow control should be turned off (HWFlowControlOff) in order to take over the control of DTR or RTS signal via function MTCRTS or MTCDTR. MTCRTS and MTCDTR can be effective only after the RTSCTS flow control bit of hardware flow control is turned off.

If the hardware flow control bit (CTSFlowControlBit) is turned on, the CTS hardware flow control is enabled. If the hardware flow control bits (RTSCTSFlowControlBits) is turned on, the RTS and CTS hardware flow control is enabled together. These are two hardware flow control methods allowed.

Syntax for SCO UNIX/OpenServer

```

#define MHWFLOW          0x40e
#define HWFlowControlOff 0x00
#define CTSFlowControlBitOn 0x01
#define RTSCTSFlowControlBitsOn 0x03
ioctl(moxa_fd, MHWFLOW, CTSFlowControlBitOn);
ioctl(moxa_fd, MHWFLOW, RTSCTSFlowControlBitsOn);

```

Syntax for UNIX SVR4.2

```

#include                <sys/stropts.h>
#include                <sys/sysmacros.h>

#define MHWFLOW        0x40e
#define HWFlowControlOff 0x00
#define CTSFlowControlBitOn 0x01
#define RTSCTSFlowControlBitsOn 0x03
struct strioctl
int                    setting;

setting = CTSFlowControlBitOn;
ioc.ic_cmd = MHWFLOW;
ioc.ic_timeout = 0;
ioc.ic_len = sizeof(int);
ioc.ic_dp = (char *)&setting;
ioctl(moxa_fd, I_STR, &ioc);

```

5. HighSpeedOn

This function let you check the high speed status for a port.

Syntax for SCO UNIX/OpenServer

```

#define HighSpeedOn    0x43d
int                    status;

ioctl(moxa_fd, HighSpeedOn, &status);

```

Syntax for UNIX SVR4.2

```

#include                <sys/stropts.h>
#include                <sys/sysmacros.h>
#define HighSpeedOn    0x43d
struct strioctl
int                    status;

ioc.ic_cmd = HighSpeedOn;
ioc.ic_timeout = 0;
ioc.ic_len = sizeof(int);
ioc.ic_dp = (char *)&status;
ioctl(moxa_fd, I_STR, &ioc);

```

Appendix F PC I/O Port Address Map

The following is the list of the I/O port addresses commonly used, which is for preventing I/O address conflict when configuring *Smartio* C104/C168 Family.

IO/ Address	Device
000-01F	DMA controller 1
020-03F	interrupt controller
040-05F	Timer
060-06F	Keyboard
070-07F	Real-time clock
080-09F	DMA page register
0A0-0BF	Interrupt controller 2
0C0-0DF	DMA controller
0F0-0FF	Math coprocessor
100-1EF	not usable
1F0-1F8	Fixed disk
200-207	Game I/O
278-27F	Parallel printer port 2 (LP2:)
2F8-2FF	Serial Port 2 (COM2:)
300-31F	Prototype card
360-36F	Reserved
378-37F	Parallel printer port 1 (LP1:)
3B0-3BF	Monochrome display
3C0-3CF	Reserved
3D0-3DF	Color graphics display
3F0-3F7	Diskette controller
3F8-3FF	Serial port 1 (COM 1:)