2G NPort Real TTY Driver Manager for UC-7400 Series

First Edition, January 2006

www.moxa.com/product

Moxa Systems Co., Ltd.
Tel: +886-2-8919-1230
Fax: +886-2-8919-1231
Web: www.moxa.com

MOXA Technical Support
Worldwide: support@moxa.com.tw
The Americas support@moxa.com

2G NPort Real TTY Driver Manager for UC-7400 Series

Copyright Notice

Trademarks

Disclaimer

Table of Contents

1

Introduction

About this Manual

In this manual, we describe how to install the “2G NPort Real TTY Driver Manager” on Moxa’s UC-7400 series of ready-to-run embedded computers. Once installed, the Real TTY Driver can be used to create a connection over a TCP/IP network between the UC-7400 embedded computer and the serial ports on Moxa’s 2G NPort serial device servers. Programmers will find the Real TTY Driver to be an invaluable tool for applications that use both UC-7400 and 2G NPort products.

UC-7400 Series Overview

The Moxa UC-7400 Series includes UC-7420, UC-7410, UC-7408, and UC-7402. These RISC-based ready-to-run embedded computers are ideal for embedded applications. UC-7400 features 8 RS-232/422/485 serial ports, dual 10/100 Mbps Ethernet ports, 8 digital input channels and 8 digital output channels, a PCMCIA interface for wireless LAN communication, a CompactFlash port for flash disk expansion, and USB ports for adding additional memory (such as a USB Flash disk).

This series use an Intel XScale IXP-422 266 Mhz RISC CPU. Unlike the X86 CPU, which uses a CISC design, the IXP-422’s RISC design architecture and modern semiconductor technology provide UC-7400 with a powerful computing engine and communication functions, but without generating a lot of heat. The built-in 32 MB NOR Flash ROM and 128 MB SDRAM give you enough memory to put your application software directly on UC-7400. And since the dual LAN ports are built right into the IXP-422 CPU, UC-7400 makes an ideal communication platform for Network Security applications. If your application requires placing UC-7400 at a site that is not located near an Ethernet LAN connection, you can connect to the network by using UC-7400’s PCMCIA port to attach a wireless LAN card.

The pre-installed Linux operating system provides an open software environment for your software program development. Software written for desktop PCs can be easily ported to the UC-7400 platform with a GNU cross compiler, without needing to modify the source code. All of the necessary device drivers, such as a PCMCIA Wireless LAN module and Keypad, LCM, and Buzzer control, are also included with UC-7400. The Operating System, device drivers, and the software you develop for your own application, can all be stored in UC-7400’s Flash memory.

UC-7400 Model Descriptions and Package Checklist

The basic features of each UC-7400 are described below:

UC-7420-LX
RISC-based Ready-to-Run Embedded Computer with 8 Serial Ports, Dual Ethernet, PCMCIA, CompactFlash, USB, Linux OS.

UC-7410-LX
RISC-based Ready-to-Run Embedded Computer with 8 Serial Ports, Dual Ethernet, Linux OS.

UC-7408-LX
RISC-based Data Acquisition Embedded Computer with 8 Serial Ports, 8 DI Channels, 8 DO Channels, Dual Ethernet, PCMCIA, CompactFlash, Linux OS.

UC-7402-LX
RISC-based Ready-to-Run Embedded Network Computer with Dual Ethernet, PCMCIA, CompactFlash, Linux OS.

All models of UC-7400 are shipped with the following items:

- 1 UC-7400 Series Embedded Computer
- Wall-Mounting Kit
- DIN-Rail Mounting Kit
- Quick Installation Guide
- Documentation & Software CD
- Cross-over Ethernet cable
- CBL-RJ45M9-150: 150 cm, 8-pin RJ45 to Male DB9 serial port cable (Except UC-7402)
- CBL-RJ45F9-150: 150 cm, 8-pin RJ45 to Female DB9 console port cable
- Power Adaptor
- Product Warranty Booklet

NOTE: *Notify your sales representative if any of the above items is missing or damaged.*

UC-7400 Product Features

- Intel XScale IXP-422 266 MHz Processor
- On-board 128 MB RAM, 32 MB Flash ROM
- Eight RS-232/422/485 serial ports (UC-7420/7410/7408 only)
- 8 digital input channels and 8 digital output channels (UC-7408 only)
- Dual 10/100 Mbps Ethernet
- USB 2.0 host for mass storage devices (UC-7420 only)
- PCMCIA/CompactFlash wireless LAN expansion (supports 802.11b/802.11g)
- LCM display and keypad for HMI (UC-7420/7410 only)
- Linux-ready communication platform
- DIN-Rail or wall mounting installation
- Robust fanless design

MOXA UC-7420

MOXA UC-7408

Product Hardware Specifications

	UC-7420	UC-7410	UC-7408	UC-7402
CPU	Intel Xscale IXP-422 266 MHz	Intel Xscale IXP-422 266 MHz	Intel Xscale IXP-422 266 MHz	Intel Xscale IXP-422 266 MHz
RAM	128 MB	128 MB	128 MB	128 MB
Flash	32 MB	32 MB	32 MB	32 MB
LAN	Auto-sensing 10/100 Mbps × 2 with built-in 1.5 KV magnetic isolation protection RJ45 Connector			
Serial Port	RS-232/422/485 × 8 RJ45 Connector	RS-232/422/485 × 8 RJ45 Connector	RS-232/422/485 × 8 RJ45 Connector	N/A
Serial Protection	15 KV ESD for all signals			N/A
Data Bits	5, 6, 7, 8			N/A
Stop Bits	1, 1.5, 2			N/A
Parity	none, even, odd, space, mark			N/A
Flow Control	RTC/CTS, XON/XOFF, RS-485 ADDC™			N/A
Speed	50 bps to 921.6 Kbps			N/A
Serial Console	RS-232 × 1 RJ45 Connector	RS-232 × 1 RJ45 Connector	RS-232 × 1 RJ45 Connector	RS-232 × 1 RJ45 Connector
DI/DO	N/A	N/A	DI × 8, DO × 8	N/A
USB 2.0 Hosts	2	N/A	N/A	N/A
USB 1.1 Client	1*	1*	1*	1*
PCMCIA	Cardbus × 1**	N/A	Cardbus × 1**	Cardbus × 1**
Storage Expansion	Compact Flash × 1***	N/A	Compact Flash × 1***	Compact Flash × 1***
LCM	128 × 64 dots	128 × 64 dots	N/A	N/A
Keypad	5	5	N/A	N/A
Real Time Clock	Yes	Yes	Yes	Yes
Buzzer	Yes	Yes	Yes	Yes
Rest Button	HW Reset × 1 Reset to Default × 1	HW Reset × 1 Reset to Default × 1	HW Reset × 1 Reset to Default × 1	HW Reset × 1 Reset to Default × 1
Power Input	12 to 48 VDC	12 to 48 VDC	12 to 48 VDC	12 to 48 VDC
Power Consumption	12W	10W	8W	7W
Dimension (W × D × H)	197 × 125 × 44 mm	197 × 125 × 44 mm	197 × 125 × 44 mm	197 × 125 × 44 mm
Weight	875 g	810 g	870 g	830 g
Operating temperature	-10 to 60°C (14 to 131°F), 5 to 95% RH			
Storage temperature	-20 to 80°C (-4 to 176°F), 5 to 95% RH			
Regulatory Approvals	EMC: CE Class A, FCC Class A Safety: UL, CUL, TUV			
Warranty	5 years	5 years	5 years	5 years

* USB Client function is reserved for future enhancement

** PCMCIA is designed for 802.11b/g wireless LAN card expansion

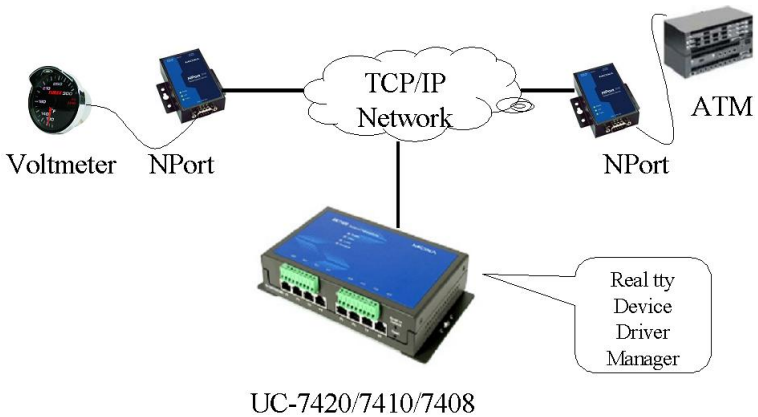
*** CompactFlash is designed for Flash memory card or Microdrive expansion

About 2G NPort Real TTY Driver Manager

The 2G NPort Real TTY Driver Manager for UC-7400 Series manages remote MOXA 2G NPort device servers operated in Real COM Mode. The Real TTY Driver Manager establishes a transparent connection between the UC-7400 and serial devices attached to the NPort device server by mapping an IP address and TCP port number of the Moxa 2G NPort's serial port to a local TTY port on the UC-7400 embedded computer. The remote serial device can be treated as a virtual TTY device on the UC-7400 embedded computer. The standard Linux serial programming interface can be used to control this virtual TTY device node in the same way that a remote serial device is controlled.

UC-7400 Operating a MOXA 2G NPort in Real COM Mode

The Real COM Mode allows users to continue using RS-232/422/485 serial communications software that was written for pure serial communications applications. The 2G NPort Real TTY Driver Manager for UC-7400 intercepts data sent to the UC-7400's tty port, packs it into a TCP/IP packet, and then redirects it through the Ethernet interface. At the other end of the connection, the Moxa NPort device server accepts the Ethernet frame from the UC-7400, unpacks the TCP/IP packet, and then transparently sends it to the appropriate serial device attached to one of the Moxa NPort serial ports.



Application Using 2G NPort Real TTY Device Manager with UC-7400

In the example shown above, the UC-7400 manages MOXA 2G NPort Device Servers remotely via a TCP/IP network. The voltmeter and ATM are connected to the NPort units via an RS-232 serial port. By running the Real TTY Driver Manager on the MOXA UC-7400 embedded computer, data from the voltmeter and ATM can be converted to the TCP/IP network. The status of these remote serial devices can be seen easily from the MOXA UC-7400.

The MOXA 2G NPort Real TTY Driver Manager for UC-7400 gives programmers an easy means of expanding the number serial ports that UC-7400 can access, providing a complete solution for controlling serial devices.

2

Software Installation

In this chapter, we describe the software installation and configuration of the UC-7400 embedded computer and the remote Moxa 2G NPort serial device server. Refer to the **Check the Software Installation** section to check whether the software is properly installed on UC-7400. If the Real TTY Driver Manager has been installed, you can skip the **Installation/Uninstallation** topics, and go directly to the **Configuration** section to configure the port mapping on UC-7400.

Real TTY Driver Manager Setup

The installation and configuration methods are introduced in this section.

- Installation/Uninstallation
- Configuration
- Controlling remote serial ports on UC-7400

Installation/Uninstallation

1. The Real TTY Driver Manager is stored in tar.gz format. The UC-7400 Linux platform supports the tar utility, allowing you to unpack the file to the UC-7400. First, use FTP to upload the Real TTY Driver Manager software to the UC-7400 /tmp directory.

```
[root@Jared_7 npreal2]# ls
cpnpreal2  moxa  new  npreal2  npreal2.tar.gz  npreal2.tools  web
[root@Jared_7 npreal2]# ftp 192.168.14.26
Connected to 192.168.14.26 (192.168.14.26).
220 Moxa FTP server (Version wu-2.6.1(2) Mon Nov 24 12:17:04 CST 2003) ready
Name (192.168.14.26:root): root
331 Password required for root.
Password:
230 User root logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /tmp
250 CWD command successful.
ftp> put npreal2.tar.gz
local: npreal2.tar.gz remote: npreal2.tar.gz
227 Entering Passive Mode (192,168,14,26,106,221)
150 Opening BINARY mode data connection for npreal2.tar.gz.
226 Transfer complete.
253931 bytes sent in 3.43 secs (72 Kbytes/sec)
ftp> bye
221-You have transferred 253931 bytes in 1 files.
221-Total traffic for this session was 254383 bytes in 1 transfers.
221-Thank you for using the FTP service on Moxa.
221 Goodbye.
[root@Jared_7 npreal2]# _
```

2. Next, telnet to the UC-7400. The login name/password are root/root.

```
Moxa login: root
Password:
```

3. Next, untar the software by typing **npreal2.tar.gz**.

```
root@Moxa:/tmp# tar xzvf npreal2.tar.gz_
```

4. The untared file is extracted to the **/tmp/npreal2** directory. Change to the **/tmp/npreal2** directory and then call the **mxinst** program to install the software

```
root@Moxa:/tmp/npreal2# ./mxinst

================================================================================
Copyright (C) 2005  Moxa Technologies Co., Ltd.
All Rights Reserved.

MOXA NPort Server Real TTY Driver V2 Installation
================================================================================

Install gawk ... OK!
Copying configurations files ... OK!
Copying driver files ... OK!

Loading TTY Driver...
Using /lib/modules/2.4.18_mvl30-ixdp425/kernel/drivers/serial/npreal2.o
Complete.


================================================================================
Installation process is completed.
The all driver files are installed on /usr/lib/npreal2/driver.
Now you can cd /usr/lib/npreal2/driver and run ./mxaddsvr to add tty port.
================================================================================

root@Moxa:/tmp/npreal2# _
```

5. If you want to uninstall the Real TTY Driver Manager from UC-7400, you need to change to the **/tmp/npreal2** directory first, and then call the **./mxuninst** program to uninstall the software.

```
root@Moxa:/tmp/npreal2# ./mxuninst
remove module loaded in kernle...OK!
remove /lib/modules/2.4.18_mvl30-ixdp425/kernel/drivers/serialnpreal2.o ...OK!
remove gawk...OK!
Complete

root@Moxa:/tmp/npreal2#
```

Check the Software Installation

1. After installation, check if the following files are installed.
   /usr/lib/npreal2/driver/mxcfmat
   /usr/lib/npreal2/driver/mxmknod
   /usr/lib/npreal2/driver/mxdelsvr
   /usr/lib/npreal2/driver/mxrmnod
   /usr/lib/npreal2/driver/npreal2d
   /usr/lib/npreal2/driver/mxaddsvr
   /usr/lib/npreal2/driver/mxloadsvr
   /usr/lib/npreal2/driver/mxuninst
   /usr/lib/npreal2/driver/npreal2d.cf
   /lib/modules/2.4.18_mvl30-ixp425/kernel/drivers/serial/npreal2.o
2. Check if following configuration files exist
   /etc/init.d/npreal2
   /etc/rc.d/rc3.d/S97npreal2 →/etc/init.d/npreal2
   /etc/rc.d/rc6.d/K97npreal2 →/etc/init.d/npreal2
   check if **export PATH=$PATH:/usr/lib/npreal2/driver** is appended to **/etc/profile**.

3. Type the following command to check if the module **npreal2.o** is loaded into kernel:
   $ lsmod

```
root@Moxa:~# lsmod
Module                  Size  Used by
npreal2               167240   0 (unused)
mxhw_cipher             9108   0 (unused)
ixp425_eth             18316   2
ixp400                727024   0 [mxhw_cipher ixp425_eth]
root@Moxa:~# _
```

The Installation Software Package List

After you untar the file npreal2.tar.gz and install the software, the following programs and configuration files will be installed on UC-7400:

Name	Location	Description
mxcfmat	/usr/lib/npreal2/driver/	This program is used by mxloadsvr to check the format of the configuration file, npreal2d.cf.
mxmknod	/usr/lib/npreal2/driver/	This program is used by mxloadsvr to create the device nodes listed in npreal2d.cf.
mxrmnod	/usr/lib/npreal2/driver/	This program is used by mxloadsvr to remove the device nodes listed in npreal2d.cf.
mxaddsvr	/usr/lib/npreal2/driver/	This program is used to add the port maps on UC-7400. Reference the section **Mapping TTY Ports**.
mxdelsvr	/usr/lib/npreal2/driver/	This program is used to remove the port maps on UC-7400. Refer to the section **Remove Mapped TTY Ports**.
mxloadsvr	/usr/lib/npreal2/driver/	This program is used by mxaddsvr/mxdelsvr to help create/remove the port map on UC-7400.
mxuninst	/usr/lib/npreal2/driver/	This program is used to uninstall the real tty device driver manager from UC-7400.
npreal2d.cf	/usr/lib/npreal2/driver/	This is te port mapping configuration file. Please do not modify this file directly. Use **mxaddsvr** and **mxdelsvr** to add or remove port maps from UC-7400. Refer to the sections **Mapping TTY Ports** and "**Remove Mapped TTY Ports**.
npreal2d	/usr/lib/npreal2/driver/	The real tty daemon creates the connection between remote Moxa NPorts and the local tty device nodes. You can invoke the daemon by using **npreal2d –t 1&** to start the daemon. The Moxa NPort real tty device driver manager will poll the Moxa NPort status every minute.
npreal2.o	/lib/modules/2.4.18_mvl30-ixdp425/kernel/drivers/serial/	These are the real tty device drivers. You can load the module into the Linux kernel by using **mxloadsvr module** or **modprobe mxloadsvr**.
profile	/etc/	check if **export PATH=$PATH:/usr/lib/npreal2/driver** is appended to **/etc/profile**

Name	Location	Description
npreal2	/etc/init.d/	Start/stop the real tty driver while the system is booting up or shutting down - /etc/init.d/npreal2 start - /etc/init.d/npreal2 stop
S97npreal2	/etc/rc.d/rc3.d/	Starts the Moxa NPort real tty device driver manager while the system is booting up. Link to /etc/init.d/npreal2
K97npreal2	/etc/rc.d/rc6.d/	Stops the Moxa NPort real tty device driver manager while the system is shutting down. Link to /etc/init.d/npreal2

Configuration

Before controlling the remote serial ports on UC-7400, you need to create a port mapping between the remote serial ports and the local tty port on the UC-7400. This configuration file must be located at **/usr/lib/npreal2/driver/npreal2d.cf**. When the Real TTY Device Manager is executing, it references the configuration file to manage the remote serial ports on UC. MOXA Real TTY Device Manager provides the following utilities to add or remove port mapping on UC-7400:

- mxaddsvr
- mxdelsvr

Configuring Moxa NPort for Real COM Mode

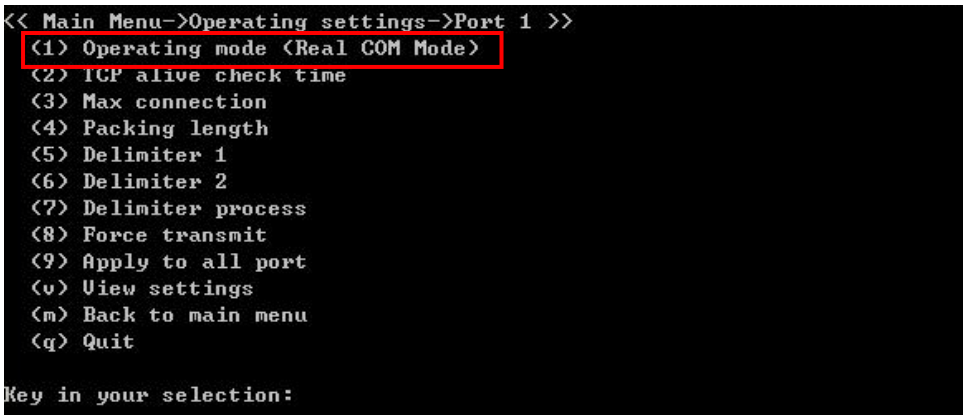
Before mapping tty ports, you must set the operation mode of the Moxa NPort to Real COM Mode. In the following example, we use telnet to configure the operating mode of NPort 5210's Port1 as Real COM mode.

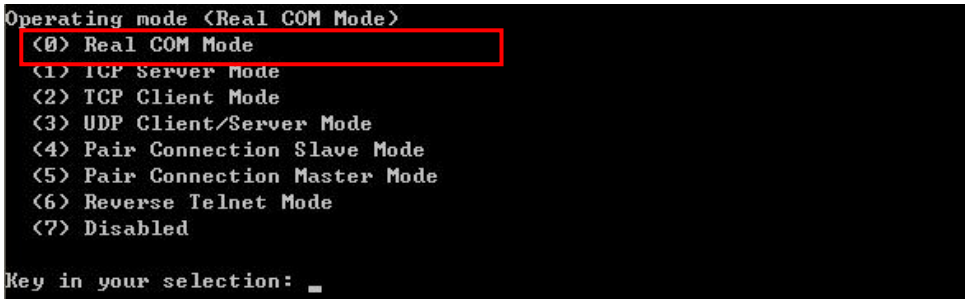
```
<< Main Menu >>
  (1) Basic settings
  (2) Network settings
  (3) Serial settings
  (4) Operating settings
  (5) Accessible IP settings
  (6) Auto warning settings
  (7) Monitor
  (8) Ping
  (9) Change password
  (a) Load factory default
  (v) View settings
  (s) Save/Restart
  (q) Quit

Key in your selection: 4
```

```
<< Main Menu->Operating settings >>
  (1) Port 1
  (2) Port 2
  (m) Back to main menu
  (q) Quit

Key in your selection: 1
```





After configuring the Moxa NPort to Real COM mode, we need to map the tty ports. Real TTY Device Manager provides two ways to map tty ports.

Mapping TTY ports

Mapping TTY ports automatically

After logging in as a super user, enter the directory **/usr/lib/npreal2/driver** and then execute **mxaddsvr** to map the target Moxa NPort serial port to the host tty ports. The syntax of **mxaddsvr** is:

mxaddsvr [NPort IP Address] [Total Ports] ([Data port] [Cmd port])

Example 1: Map two ports of the remote NPort (192.168.14.9) on UC's /dev/ttyr00 and /dev/ttyr01

```
# cd /usr/lib/npreal2/driver
# ./mxaddsvr 192.168.14.9 2
```

```
Adding Server...

Added server: 192.168.14.9

rm -f /dev/ttyr00
rm -f /dev/cur00
rm -f /dev/ttyr01
rm -f /dev/cur01
mknod -m 666 ttyr00 c 33 0
mknod -m 666 cur00 c 38 0
mknod -m 666 ttyr01 c 33 1
mknod -m 666 cur01 c 38 1
Complete.

#=========================================================#
#   This configuration file is created by Moxa NPort      #
#   Administrator Program automatically, please do not    #
#   modify this file by yourself.                         #
#=========================================================#
ttymajor=33
calloutmajor=38
#[Minor] [ServerIP]     [data]  [cmd]   [FIFO] [ttyName] [coutName]
0       192.168.14.9    950     966     1       ttyr00  cur00
1       192.168.14.9    951     967     1       ttyr01  cur01
root@Moxa:/usr/lib/npreal2/driver#
```

In Example 1, 16 tty ports are added, all with IP 192.168.3.4, but with data ports equal to (950, 951, ..., 965), and command ports equal to (966, 967, 968, ..., 981).

Example 2: Map two ports of remote Moxa NPort (192.168.14.9) on UC's /dev/ttyr00 and /dev/ttyr01, the data ports are numbered from 4001, and the command ports are numbered from 966.

cd /usr/lib/npreal2/driver

./mxaddsvr 192.168.3.4 16 4001 966

```
root@Moxa:/usr/lib/npreal2/driver# ./mxaddsvr 192.168.14.9 4001 966
```

```
Adding Server...

Added server: 192.168.14.9

rm -f /dev/ttyr00
rm -f /dev/cur00
rm -f /dev/ttyr01
rm -f /dev/cur01
mknod -m 666 ttyr00 c 33 0
mknod -m 666 cur00 c 38 0
mknod -m 666 ttyr01 c 33 1
mknod -m 666 cur01 c 38 1
Complete.

#=========================================================#
#   This configuration file is created by Moxa NPort      #
#   Administrator Program automatically, please do not    #
#   modify this file by yourself.                         #
#=========================================================#
ttymajor=33
calloutmajor=38
#[Minor] [ServerIP]     [data]  [cmd]   [FIFO] [ttyName] [coutName]
0       192.168.14.9    4001    966     1       ttyr00  cur00
1       192.168.14.9    4002    967     1       ttyr01  cur01
```

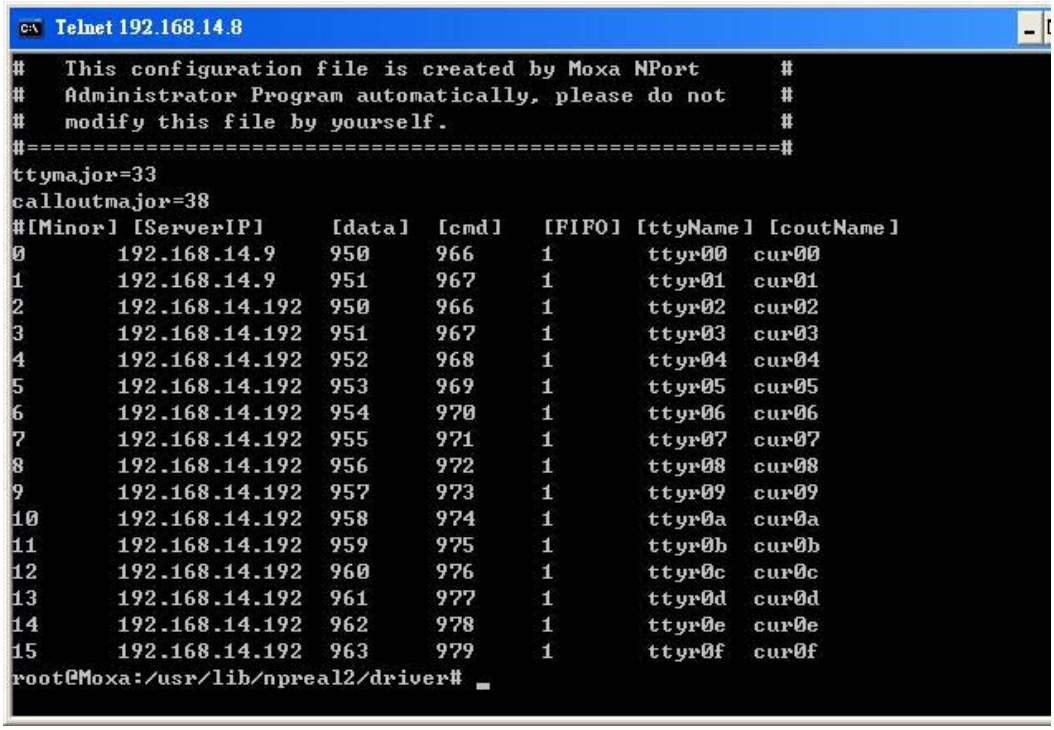
In Example 2, 16 tty ports will be added, all with IP 192.168.3.4, but with data ports equal to (4001, 4002, ..., 4016), and command ports equal to (966, 967, 968, ..., 981).

Mapping TTY ports manually

After entering the directory **/usr/lib/npreal2/driver**, you can modify **npreal2d.cf** to map Moxa NPort serial ports to tty ports, and then execute **mxloadsvr** to activate the modifications.

The following tasks will be performed:

- Modify **npreal2d.cf**
- Create tty ports in the directory **/dev** with major & minor number configured in **npreal2d.cf**
- Stop and then restart the driver.



Remove Mapped TTY ports

Remove the mapped TTY ports automatically

After logging in as root, enter the directory **/usr/lib/npreal2/driver** and then execute **mxdelsvr** to delete a server. The syntax of **mxdelsvr** is:

mxdelsvr [IP]

Example: Remove the two port maps of the remote Moxa NPort (192.168.14.2)

cd /usr/lib/npreal2/driver

./mxdelsvr 192.168.14.2

```
Delete Server ...
Deleted server: 192.168.14.9

Complete.
```

If you don't include the IP address in the command line, a numbered list of IP addresses for servers currently installed, along with the total number of ports for each server, will be listed on the screen. To remove the tty ports for a particular server, type the number next to the server's IP address and then hit Enter. The following tasks will be performed:

- Modify the npreal2d.cf
- Remove the relevant tty ports in directory **/dev**
- Stop and then restart the driver.

```
Delete Server ...

[Index] [Server IP]     [Port(s)]
  1     192.168.14.9      2

Select: 1
```

Remove the mapped tty ports manually

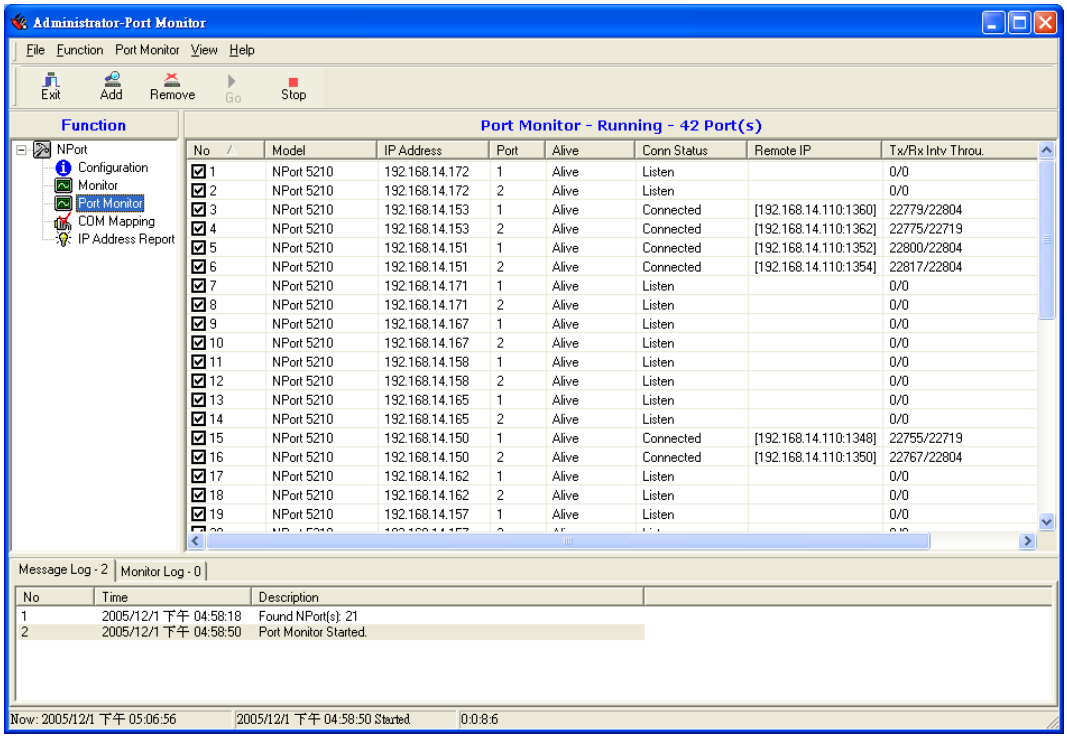
After entering the directory **/usr/lib/npreal2/driver**, you can modify **npreal2d.cf** to delete servers and ports manually, and then execute **mxloadsvr** to activate the modifications.

Check the Port Mapping

Refer to the section **Check the Software Installation** to check if the installation is successful or not. Then you can follow these steps check whether the port mapping is successful.

- Check the network interface, ixp0 or ixp1, configurations
  - ifconfig
- After verifying the IP address configuration, add the port mapping between UC and Moxa NPort.
  For example,
  - mxaddsvr 192.168.14.9 2
- Read the configurations of /dev/ttyr00
  - stty -a < /dev/ttyr00
- Modify the baud rate of Moxa NPort serial port
  - stty 1200 < /dev/ttyr00
- Read the settings of /dev/ttyr00 after the baud rate is modified. If the baud rate is changed to 1200, this implies that you can configure the remote serial port.
  - stty -a < /dev/ttyr00

You can also check the connection status with NPort Administrator, which can be downloaded from MOXA's website.



Configuring the Remote Serial Ports on UC-7400

UC-7400 is an embedded Linux computing and communication platform. The remove serial port is mapped as a real tty port on UC-7400. We can use the Linux stty utility to change or print the terminal line setting of the remote serial ports on UC-7400.

- Print all current settings in /dev/ttyr00
  - stty -a < /dev/ttyr00
- Enable xon and set baud to 1200
  - stty ixon 1200 < /dev/ttyr00
  - stty -a < /dev/ttyr00

For more information about the stty utility, read the **stty man** page.

The mapped tty ports can also be controlled by the standard Linux serial programming interface. Appendix A lists all Linux Serial Programming API lists supported by Real TTY Device Manager.

3

Serial Programming API Support List

ssize_t read(int fd, void *buf, size_t count);

Function Name	Parameters
read	---

ssize_t write(int fd, const void *buf, size_t count);

Function Name	Parameter
write	---

int ioctl(int d, int request, ...);

Function Name	int request
ioctl	TCGETS
	TIOCEXCL
	TIOCNXCL
	TIOCMGET
	TIOCMSET
	TIOCMBIS
	TIOCMBIC
	TCSBRK
	TCSBRKP
	TIOCGSOFTCAR
	TIOCSSOFTCAR
	TIOCGSERIAL
	TIOCSSERIAL
	TIOCSERGETLSR
	TIOCMIWAIT
	TCXONC
	TCFLSH

Notes:

- If you want to set the tty as exclusive mode, do not use open( ) with O_EXCL flag. You should open the tty and then use the ioctl( ) with TIOCEXCL parameter to set the exclusive mode.
- The parameter of TCXONC ioctl command only supports TCOOFF and TCOON, which can be used to suspend/resume the output.
- TCGETA and TCSETA are not supported. Use the tcgetattr( ) and tcsetattr( ) as a substitution.
  - Ex: ioctl(Pa.fd,TCSETA,&tio); → tcsetattr(Pa.fd,TCSANOW,&tio);
  - Ex: ioctl(Pa.fd,TCGETA,&tio1); → tcgetattr(Pa.fd,&tio1);

int select(int n, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct timeval *timeout);

Function Name	Parameters
Select	---

int tcgetattr(int fd, struct termios *termios_p);

Function Name	struct termios *termios_p
tcgetattr	---

int tcsetattr(int fd, int optional_actions, struct termios *termios_p);

Function Name	struct termios *termios_p
tcsetattr	c_iflag
	BRKINT
	IGNBRK
	ISTRIP
	INLCR
	IGNCR
	ICRNL
	IUCLC
	IXON (set with IXOFF)
	IXOFF(set with IXON)
	c_oflag
	OPOST
	OLCUC
	ONLCR
	OCRNL
	ONLRET

	c_cflag
	CSIZE
	CSTOPB
	CREAD
	PARENB
	PARODD
	HUPCL
	CLOCAL
	CRTSCTS
	c_lflag
	ISIG
	ICANON
	ECHO
	ECHOE
	ECHOK
	ECHONL
	NOFLSH
	IEXTEN
	c_cc
	VINTR
	VQUIT
	VERASE
	VKILL
	VEOF
	VMIN
	VEOL
	VTIME
	VLNEXT
	int optional_actions
	TCSANOW

Notes:

- c_iflag: IXON and IXOFF must be set together.

int tcsendbreak (int fd, int duration);

Function Name	int duration
tcsendbreak	0

int tcflush (int fd, int queue_selector);

Function Name	queue_selector
tcflush	TCIOFLUSH
	TCOFLUSH
	TCIFLUSH

References

TTY_IOCTL:
http://annys.eines.info/cgi-bin/man/man2html?tty_ioctl+4

Serial Programming HOWTO:
http://www.tldp.org/HOWTO/Serial-Programming-HOWTO/intro.html

Serial Programming Guide for POSIX System
http://digilander.libero.it/robang/rubrica/serial.htm#CONTENTS