8000E Series

8000E Series Software User’s Manual

8000E Series New Features

1. Virtual COM Technology
2. Ethernet I/O Technology
3. Web-server Technology
4. MiniOS7 & Xserver Inside
5. I/O Expansion Bus Inside
6. Time to market & Cost Effective Solution

$$\begin{tikzcd}
\text{8000E Series New Features} \arrow[r] & \boxed{\textbf{Your Powerful Tools}} \arrow[d] \\
 & \boxed{\textbf{Create New Ideas}} \arrow[d] \\
 & \boxed{\textbf{Create New Applications}}
\end{tikzcd}$$

Warranty

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

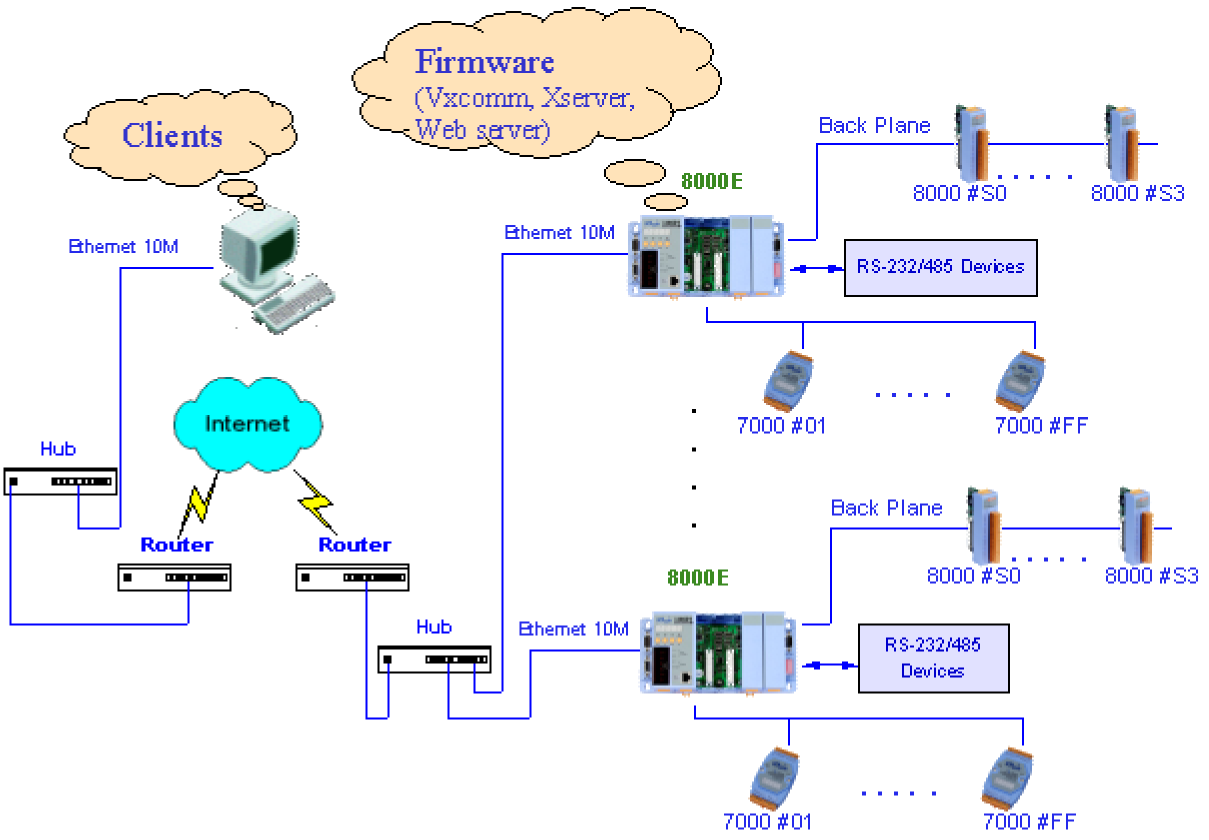
Warning

ICP DAS assume no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Table of Contents

1. Introduction

One 8000E series controller comes equipped with one RJ-45 connector, one I/O expansion bus and several serial COM ports. The 8000E is an embedded controller which, with aid of c language, can help you develop your own programs. Included with the 8000E are many demos and libraries. The 8000E can be used to access devices via Ethernet/Internet or RS-232/485/422. In addition, each 8000E controller has one I/O expansion bus. Our expansion boards can be mounted quickly and easily to implement various I/O functions,such as D/I, D/O, A/D, D/A, Timer/Counter, Flash memory and battery backuped SRAM.



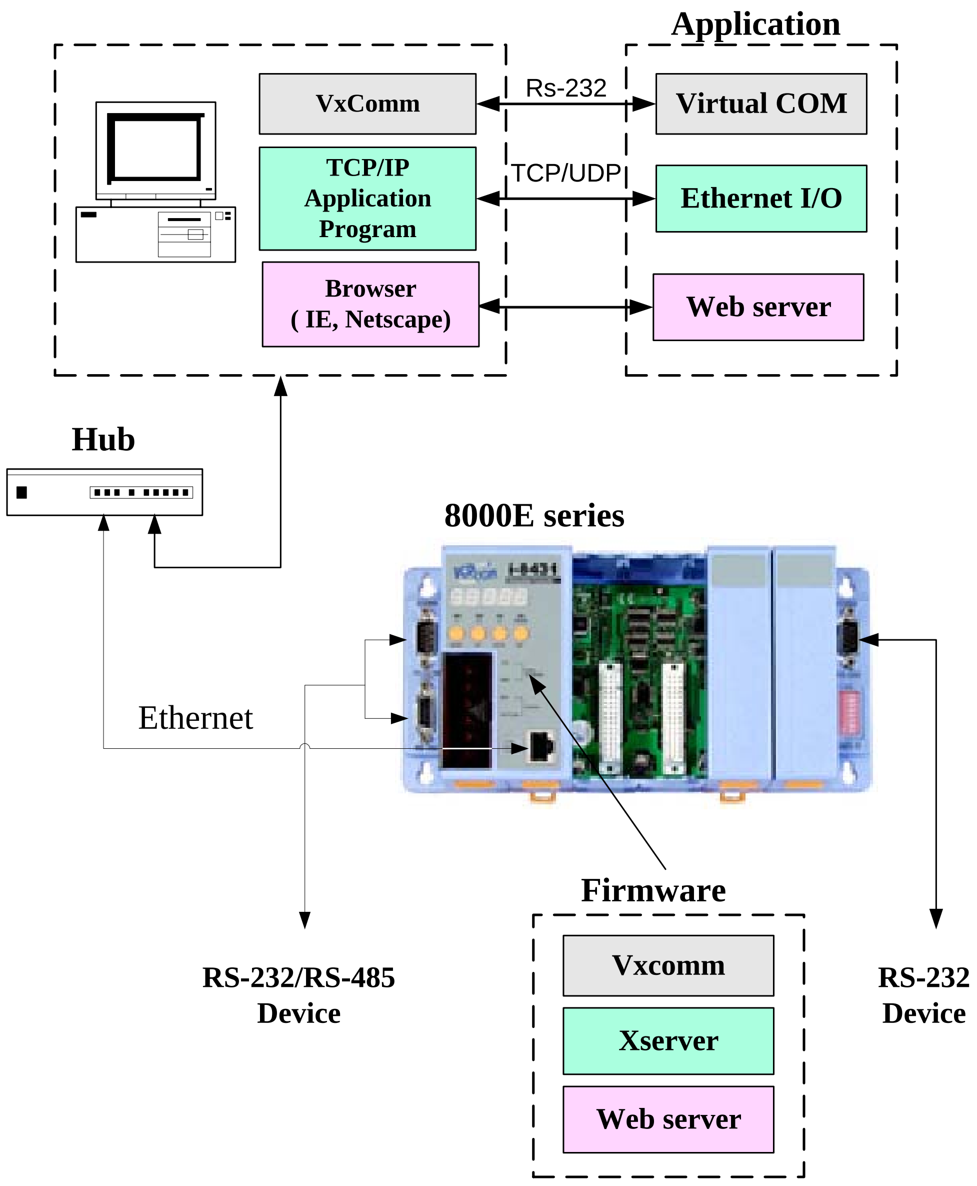
Note: Please refer to below documents for more information about hardware and how to getting start quickly.

- **8000Ehh.pdf**
- **8430_Quick_Start.pdf**
- **8431_Quick_Start.pdf**

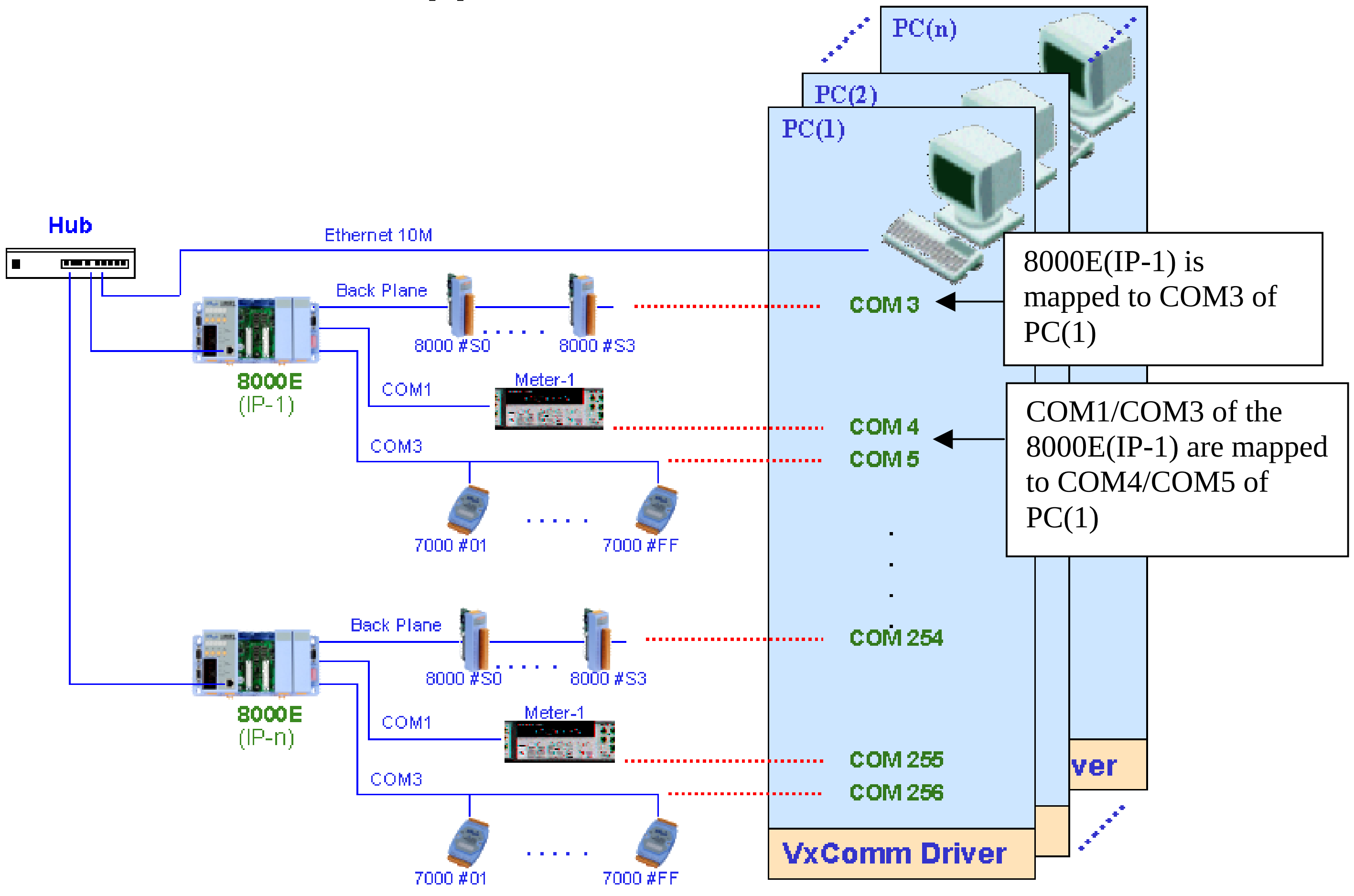
located in 8000\843x883x\Document

1.1 3 typical applications

8000E series controllers have 3 typical applications (Virtual COM, Ethernet I/O and web server implementation). These applications use different firmware and program styles. Users can choose any of the 3 applications they prefer.
Using the Virtual COM application, one PC can control 256 COM ports (including real COM ports). The VxComm firmware will turn your 7188E into a RS-232 to Ethernet/Internet converter.
Using the Ethernet I/O and Web Server applications, users can program the firmware of the 8000E (Xserver, Web server).



1.1.1 Virtual COM application

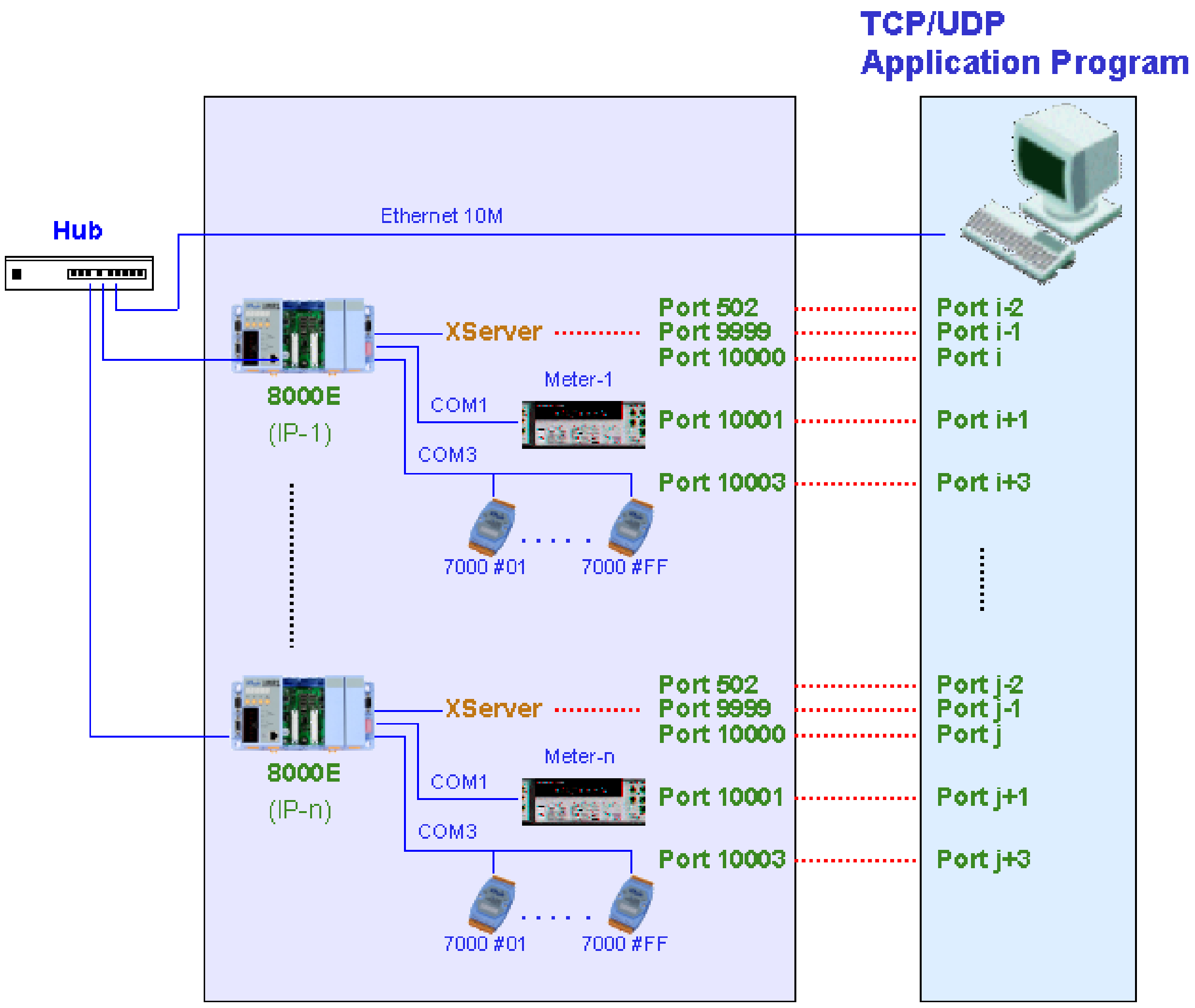


To use the Virtual COM application, first install the VxComm Driver. After installation, the VxComm Utility can map any 8000E remote COM port. These virtualized COM ports can be used by the PC to control devices directly, just as you would use COM1 (real COM port) to control devices. When using the Virtual COM application, one PC can use a maximum of 256 COM ports.
Users need not worry about network connections. The VxComm Driver will handle all Ethernet/Internet connections.

The advantages of VxComm:

1. Users can upgrade their systems to the Ethernet/Internet with increased ease as program code needs no modification.
2. The internal firmware of 8000E supports multiple clients. One 8000E can handle a maximum of 30-N socket connections simultaneously, with N being the number of the 8000E's COM ports.
   For example: If one PC uses two virtual COM ports connect to COM ports of one 8430. The 8430 allows a maximun of 14 PC connections.

1.1.2 Ethernet I/O application



The Xserver is a powerful program designed for Ethernet I/O appllications. It supplies the 8000E with a range of flexible options. Users can modify the Xserver to control all of the 8000E's hardware: COM ports, I/O expansion boards, a 7 Seg LED, or other relevant products.

The advantages of the Xserver:

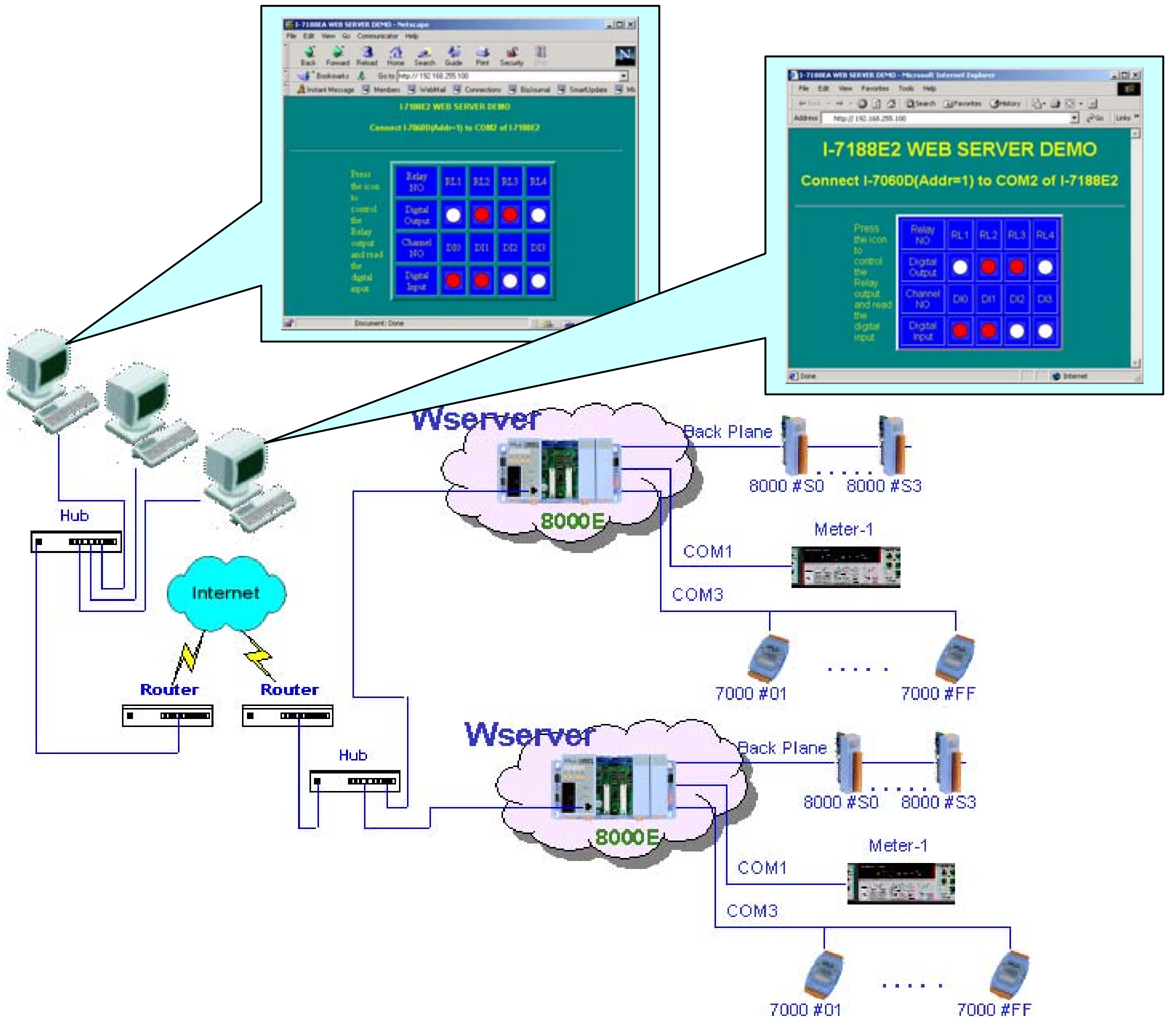
1. We design, maintain, update the Xserver for all users.
2. Xserver can be easily modified through the use of general development tools such as TC/BC/MSC. All relative libraries are coded in C language.
3. High running speed. The original Xserver (Demo4.exe) can run about 750 scan loops per second (version 2.6.14).
4. Most program code of the Xserver is finished. Ethernet/Internet communication and program loop control are all finished in VxComm.lib.

This vastly reduces user's developing time.

5. To modify Xserver, users need only to modify the 6 functions
6. The command protocol is designed to fit most of the 8000E's requirements.
7. Users can develop and extend their private command protocol very easily.
8. It supports multi-clients. The Xserver can handle a maximum of 30-N connections simultaneously, with N being the number of the 8000E's COM ports.
9. Auto wake up option. The Xserver will check on packet timeouts. If the software crashes, the Xserver will wake itself up automatically.
10. The Xserver demos (TC/BC/MSC) and Client demos (VB/VC) are included.

1.1.3 Web server application

With the help of Wserver (Web server), users are able to use standard browsers (such as IE or Netscape) to access the I/Os of the 8000 modules or devices connected to any of the 8000E's COM port.



1.2 Directory tree of software and literature

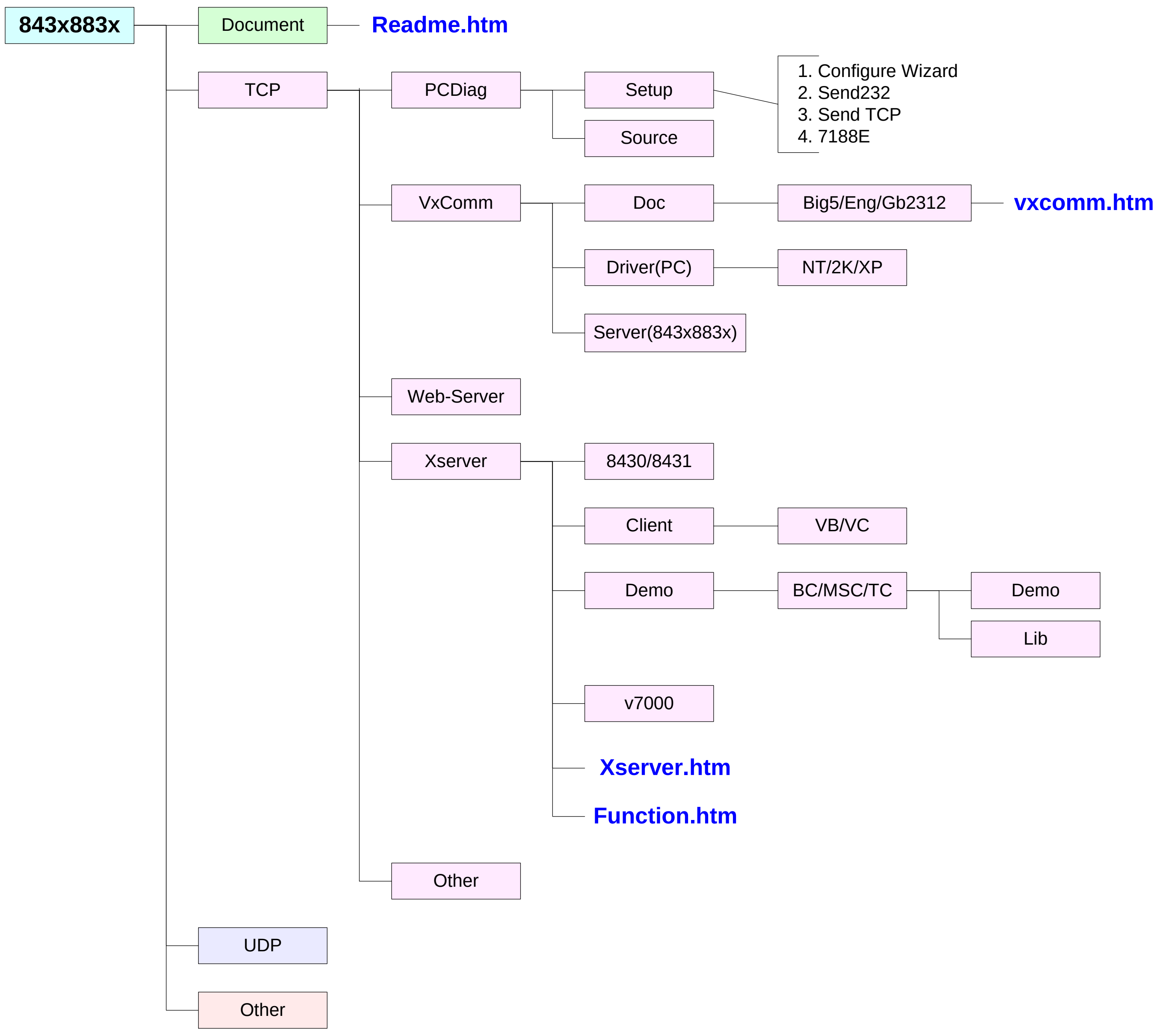
To help users reduce developing time, we support many software resources, including documents, drivers, libraries, diagnostic programs, client programs and many Xserver demos. You can quickly find which resources you need by using the directory tree.

The content of shipped CD:

```
AUTORUN.INF             → for CD auto run
README.TXT              → please read this file first
NAPDOS ─┬── Web         → Company web site
        ├── Ar4         → for Acrobat reader
        ├── Bin         → for autorun
        └── 8000 ─┬─ Index.htm      → please read this file first
                  ├─ 841x881x
                  ├─ 842x882x
                  ├─ 843x883x ─┬─ Document
                  │            ├─ TCP
                  │            └─ UDP
                  ├─ 844x884x
                  ├─ 87K
                  ├─ Catalog
                  └─ Common
```

Note: The software & manual are updated frequently, so the content of the companion CD is also updated frequently. The best way is to read every **README.TXT** located in every directory. All updated information is given in these files.

Sub directory tree of 843x883x:



1.3 Software Installation

1.3.1 Installation Steps

The installation steps are given as follows:

Step 1: Change the directory to the destination folder as follows.
For example (CD-ROM Drive is "D:")

```
C:\>d:
D:\>cd \NAPDOS\8000\843x883x
D:\NAPDOS\8000\843x883x\>_
```

Step 2: Make a new directory for the 8000E. Xcopy all files.

```
C:\>md 8000E
C:\>cd 8000E
C:\8000E\>xcopy d: c: /s /v
```

1.3.2 Installing 7188X.exe/7188XW.exe

The 7188X.exe/7188XW.exe is used to download and debug programs. Users should copy it to the PATH directory first. Then users can execute 7188X.exe/7188XW.exe in any directory on the host-PC. The installation steps are given as follow:

Step 1: Change the directory to the destination folder as follows.
For example (CD-ROM Drive is "D:")

```
C:\>d:
D:\>cd \NAPDOS\8000\Common\MiniOS7
D:\NAPDOS\8000\Common\MiniOS7\>_
```

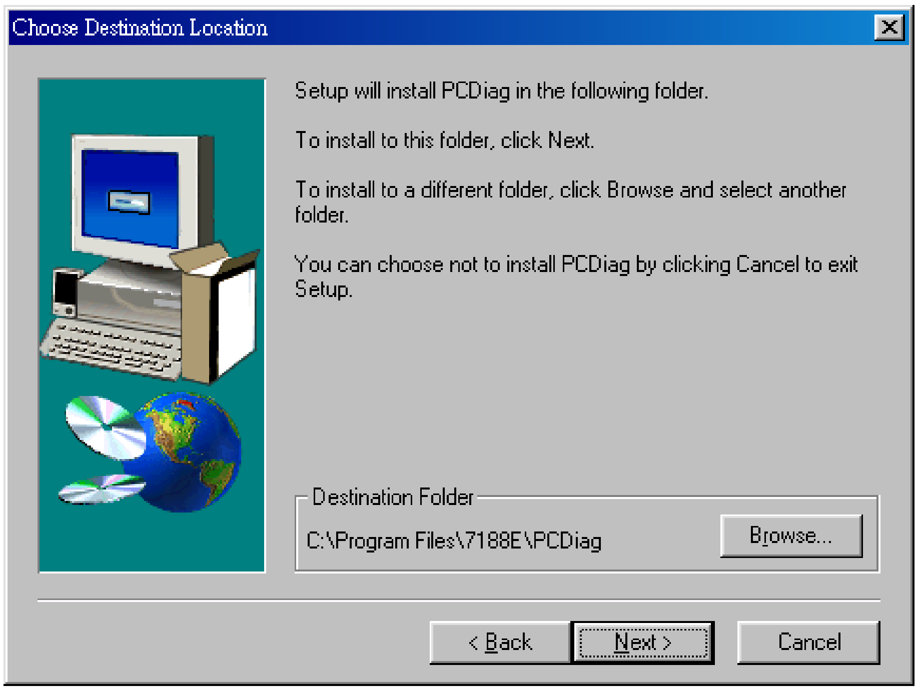
Step 2: Copy 7188X.EXE from CD to the PATH directory (defined in PATH).

```
C:\>cd windows
C:\windows\>copy d:7188x.exe
```

Note: 7188XW.exe is designed for win32 system. So it can be used for USB-RS232 or PCMCIA-RS232 port.

1.3.3 Installing PC diagnostic tools

Step1: Run Setup.exe from the 8000\843x883x\TCP\PCDiag\Setup directory.
Setp2: Choose destination folder.



Choose Destination Location

Setup will install PCDiag in the following folder.

To install to this folder, click Next.

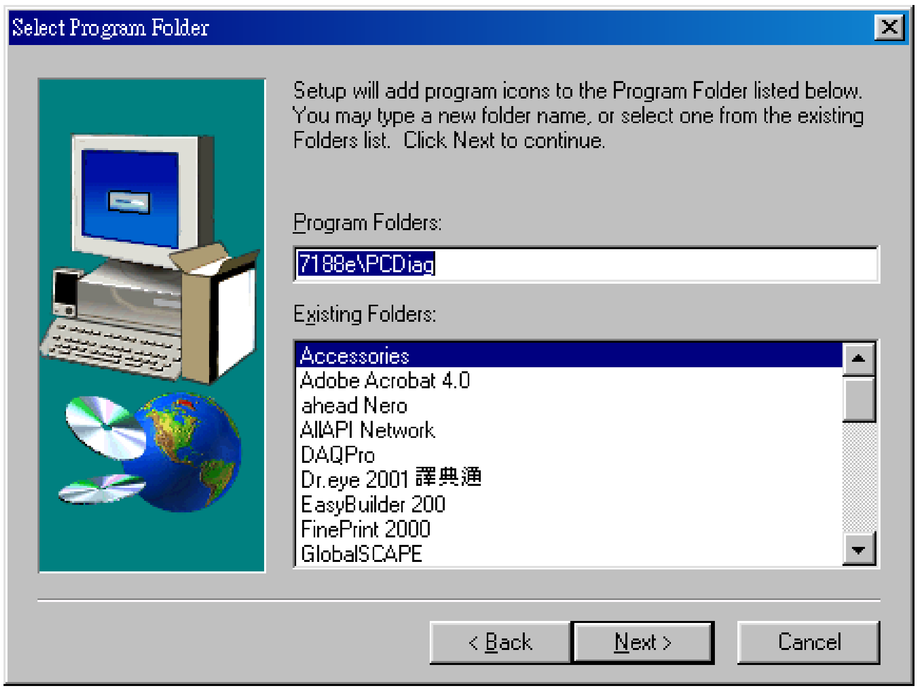
To install to a different folder, click Browse and select another folder.

You can choose not to install PCDiag by clicking Cancel to exit Setup.

Destination Folder

C:\Program Files\7188E\PCDiag

Step3: Select program folder.



Select Program Folder

Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing Folders list. Click Next to continue.

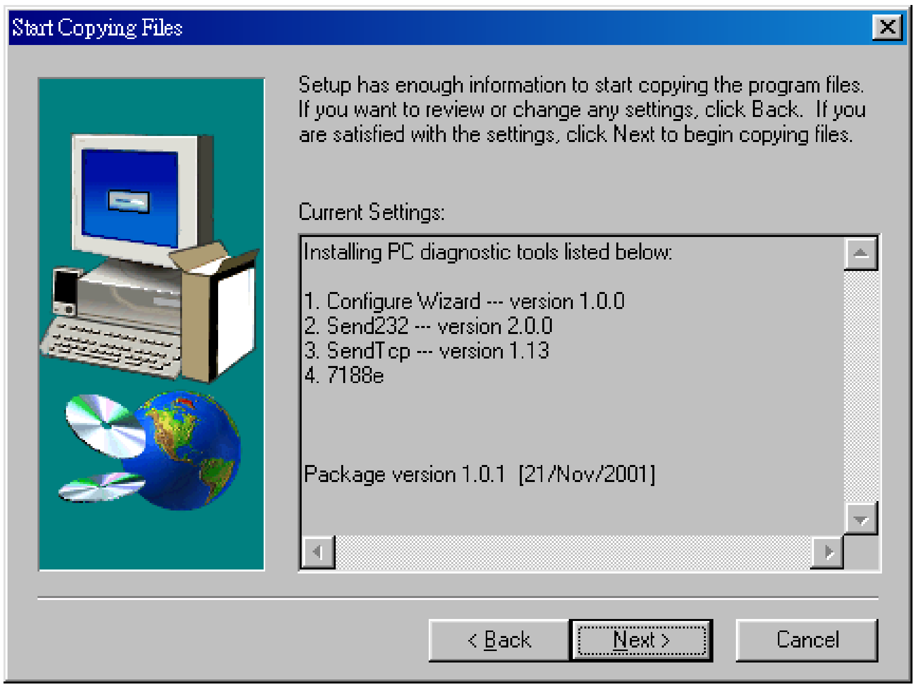
Program Folders:

7188e\PCDiag

Existing Folders:

Accessories
Adobe Acrobat 4.0
ahead Nero
AllAPI Network
DAQPro
Dr.eye 2001 譯典通
EasyBuilder 200
FinePrint 2000
GlobalSCAPE

Step4: Start copying files.



Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.

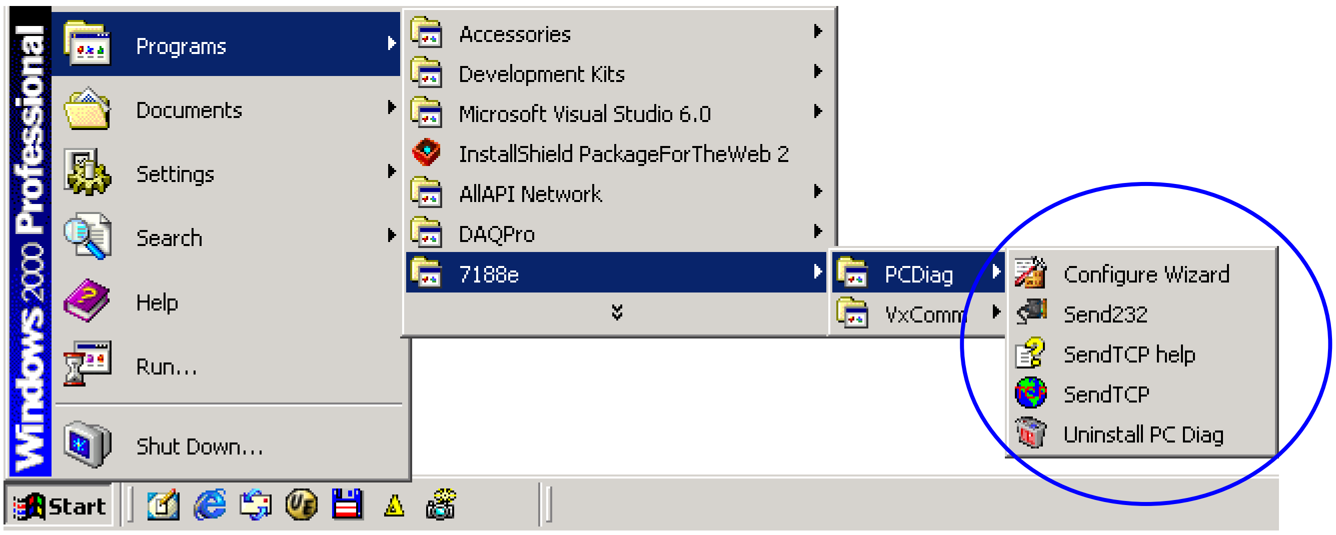
Current Settings:

Installing PC diagnostic tools listed below:

1. Configure Wizard --- version 1.0.0
2. Send232 --- version 2.0.0
3. SendTcp --- version 1.13
4. 7188e

Package version 1.0.1 [21/Nov/2001]

After finished installing, 5 items can be found in the PCDiag group.



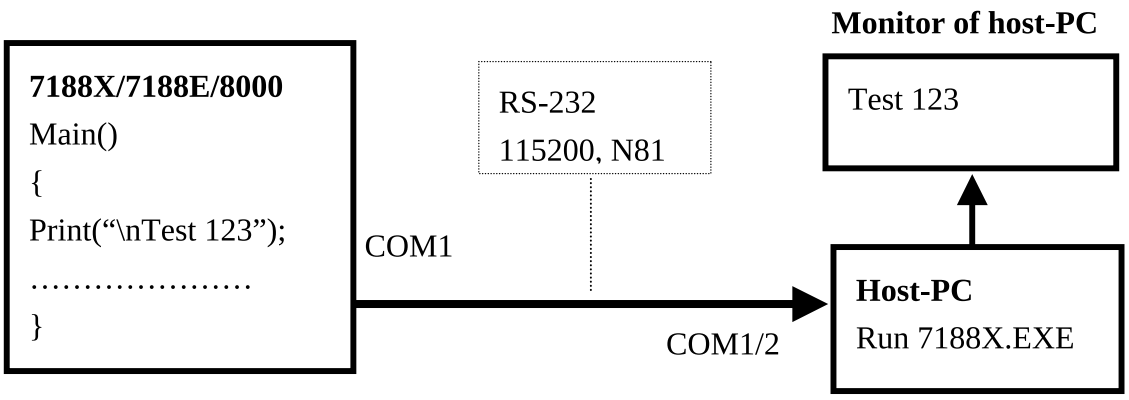
1.4 Software Utilities

1.4.1 7188X.EXE Utility for Host-PC

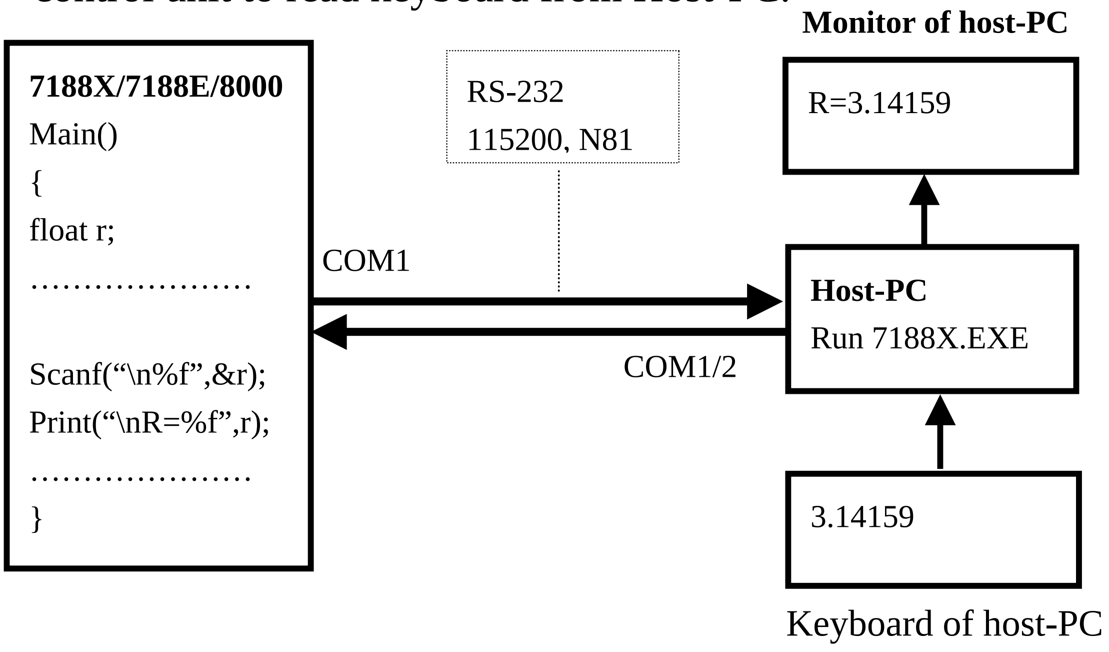
The utility program, 7188X.EXE, can be used as follows:

- **Downloads user's programs from host-PC into 7188, 7188X, 7188E and 8000 families.**

- **Shows debug string to monitor of host-PC**
  Three standard output libraries, Putch, Print & Puts, will allow main control unit to send output string to monitor of Host-PC.



- **Keys-in test data from keyboard of host-PC**
  Three standard input libraries, Getch, Scanf & LineInput, will allow main control unit to read keyboard from Host-PC.



Hot-key of 7188x.exe:

Command	**Description**
F1	Shows help messages of 7188x.exe
Alt_1	Uses PC's COM1
Alt_2	Uses PC's COM2
Alt_C	Switchs to command mode to change PC COM port's baudrate and data format. Press ENTER to confirm the setting. Press LEFT/RIGHT arrow key to select different field. Press Any key to switch different values. Press ENTER in the last field will stop this operation.
Alt_D	Sets the date of RTC to the PC's date.
Alt_T	Sets the time of RTC to the PC's time
Alt_E	For downloading files into memory. Only after the message "**Press ALT_E to download file!**" is shown on screen, can users press Alt_E.
Alt_L	Switchs normal/line mode. In line-mode, all characters-pressed will not send to COM until ENTER is pressed. It is designed for testing the 7000 series.
Alt_X	Quits 7188X.EXE.
F2	Sets the file name for download (without download operation).
Alt_F2	Sets multiple filenames for download. (10 files maximum. If set less then 10 files, add '*' to end.)
Ctrl_F2	Shows COM1 & COM2 messages (for easily COM port testing). Press ALT_X to return to the original mode. Press TAB to switch the cursor between these two windows.
F5	Runs the program specified by F2 and arguments set by F6.
F6	Sets the arguments of the execution file set by F2. (10 arguments maximum. If set less than 10 arguments, add '*' to end).
F8	F8=F9+F5.
F9	Downloads the file specified by F2 into FLASH memory.
Alt_F9	Downloads all files specified by ALT_F2 into FLASH memory.
F10	Downloads the file specified by F2 into SRAM and execute it.
F12	For 7521/7522/7523 to test RS-232.
… more …	… more …

1.4.2 7188XW.EXE Utility for Host-PC

7188xw.exe is **the Win32 version of 7188x.exe**. The difference beteween 7188x.exe and 7188xw.exe is:
7188x.exe: Uses standard COM ports (COM1/COM2).
7188xw.exe: Supports RS-232 COM ports using USB and PCMCIA interfaces.

Command line options of 7188xw.exe:

Option	**Description**
/c#	Uses PC's COM#
/b#	Sets baudrate of PC’s COM port (default is 115200)
/s#	Sets screen’s display-rows (default is 25, max. is 50)

Hot-key of 7188xw.exe:

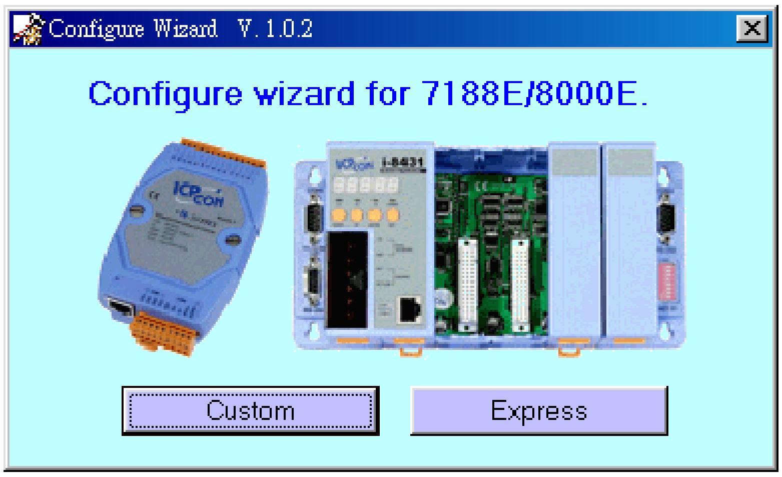
Command	**Description**
F1	Shows help messages of 7188xw.exe
Alt_F1	Shows the Chinese (Big5) help messages of 7188xw.exe
Ctrl_F1	Shows the Chinese (GB2312) help messages of 7188xw.exe
Alt_1	Uses PC's COM1
Alt_2	Uses PC's COM2
Alt_3	Uses PC's COM3
Alt_4	Uses PC's COM4
Alt_5	Uses PC's COM5
Alt_6	Uses PC's COM6
Alt_7	Uses PC's COM7
Alt_8	Uses PC's COM8
Alt_9	Uses PC's COM9
Alt_A	Switches between normal mode and ANSI-Escape-code-support mode
Alt_C	Switches to command mode. Supports commands: b#: sets new baudrate of PC’s COM ports. c#: Uses PC’s COM#. n/e/o: sets parity to none/even/odd. 5/6/7/8: sets data bits to 5/6/7/8. p#: sets PC’s working directory. q: quits command mode.

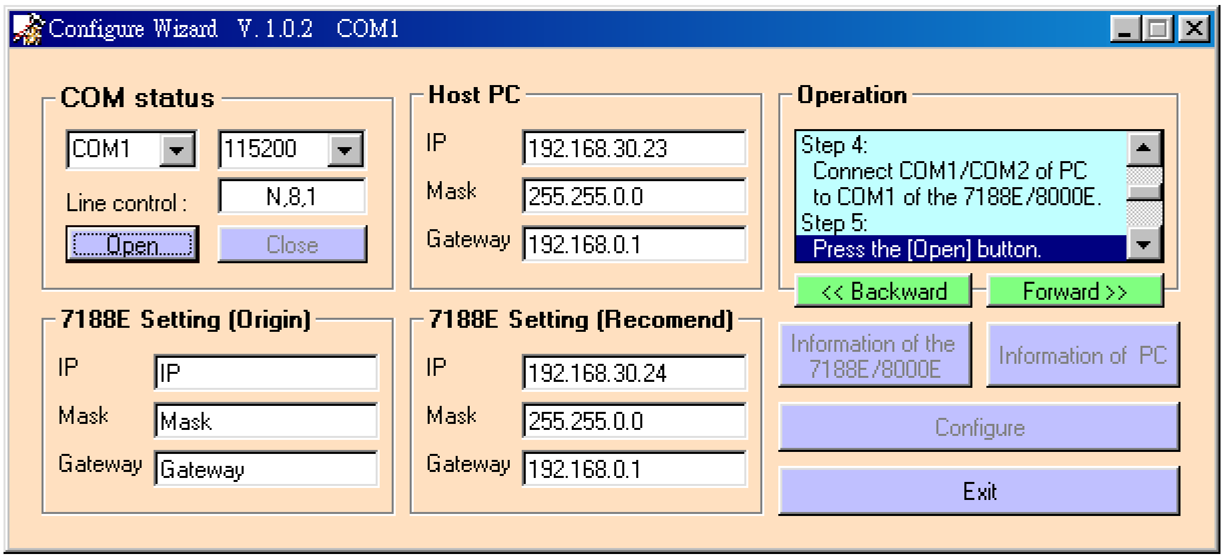
Alt_D	Sets the date of RTC to the PC's date.
Alt_T	Sets the time of RTC to the PC's time
Alt_E	For downloading files into memory. Only after the message “Press ALT_E to download file!” is shown on screen, can users press Alt_E.
Alt_H	Switchs Hex/ASCII display mode.
Alt_L	Switches normal/line mode. In line-mode, all characters-pressed will not send to COM until the ENTER is pressed. It is designed for testing the 7000 series.
Alt_X	Quits 7188X.EXE.
F2	Sets the file name for download (without download operation).
Alt_F2	
F5	Runs the program specified by F2 and arguments set by F6.
Alt_F5	Runs the program stored in SRAM.
F6	Sets the arguments of the execution file set by F2. (10 arguments maximum. If set less than 10 arguments, add ‘*’ to end).
Ctrl_F6	Clears screen.
F8	F8=F9+F5.
F9	Downloads the file specified by F2 into FLASH memory.
Alt_F9	Downloads all files specified by ALT_F2 into FLASH memory.
F10	Downloads the file specified by F2 into SRAM and execute it.
Alt_F10	Downloads the file specified by F2 into SRAM memory.
Ctrl_B	Sends a BREAK signal to the PC’s COM port that is used by 7188xw.exe.
… more …	… more …

1.5 PC Diagnostic tools

The PC Diagnostic tools include:

➢ **Configure Wizard:** guides users step by step in configuring the 7188E/8000E's network setting.




Configure Wizard V. 1.0.2 COM1

COM status: COM1, 115200, Line control: N,8,1, Open, Close

Host PC: IP 192.168.30.23, Mask 255.255.0.0, Gateway 192.168.0.1

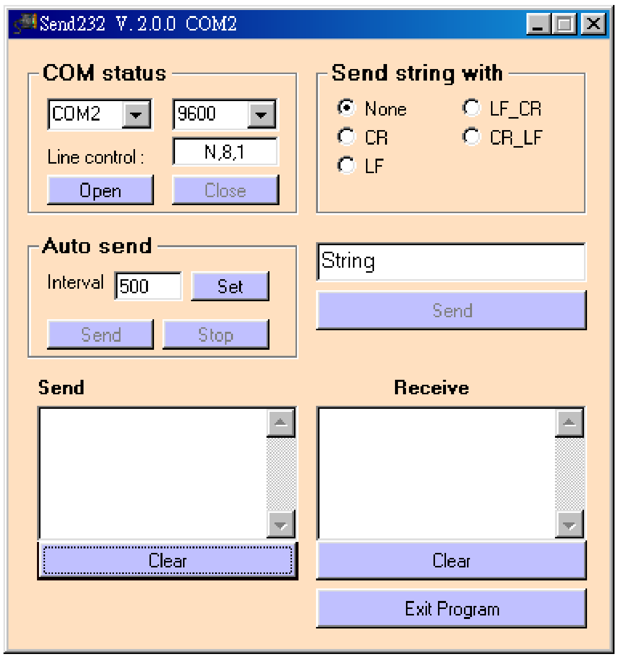
Operation:
Step 4:
Connect COM1/COM2 of PC to COM1 of the 7188E/8000E.
Step 5:
Press the [Open] button.

7188E Setting (Origin): IP IP, Mask Mask, Gateway Gateway

7188E Setting (Recomend): IP 192.168.30.24, Mask 255.255.0.0, Gateway 192.168.0.1

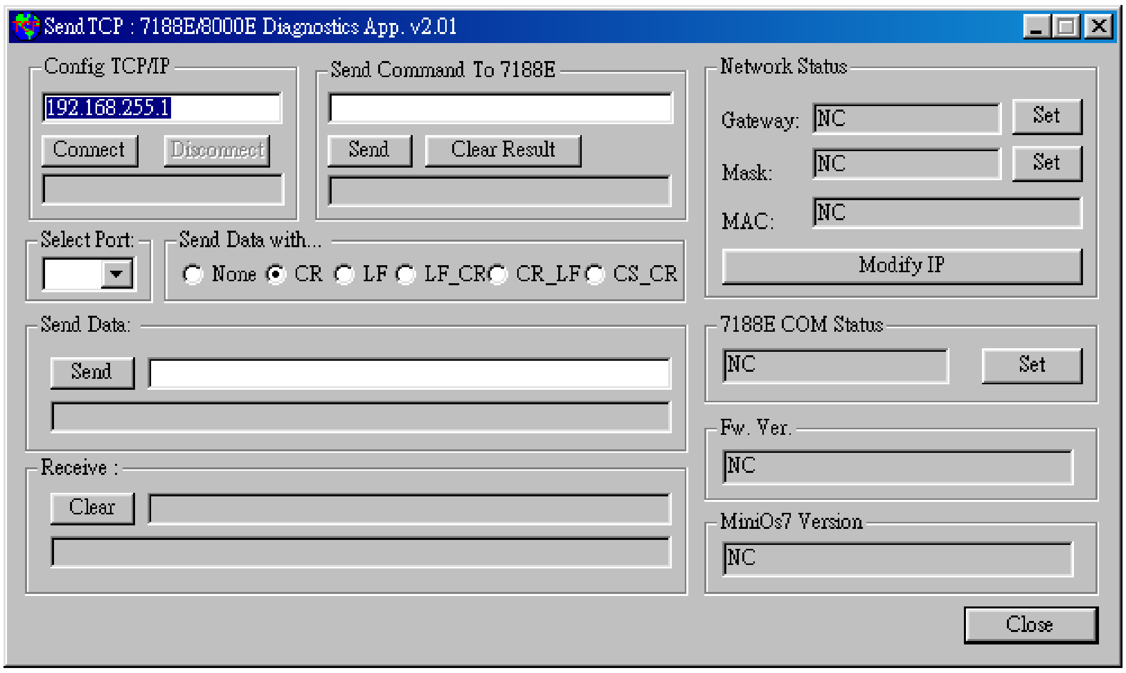
Please refer to 8000\843x883x\Document\8000Ehh.pdf for more operating details.

➤ **Send232:** uses serial port (RS-232) interface to communicate with devices. Can be used to test the Virtual COM technology.



Please refer to sec. 3.6 for more operating details.

➤ **SendTCP:** uses TCP protocol to communicate with the 7188E/8000E and devices which are connected to the 7188E/8000E's COM ports.



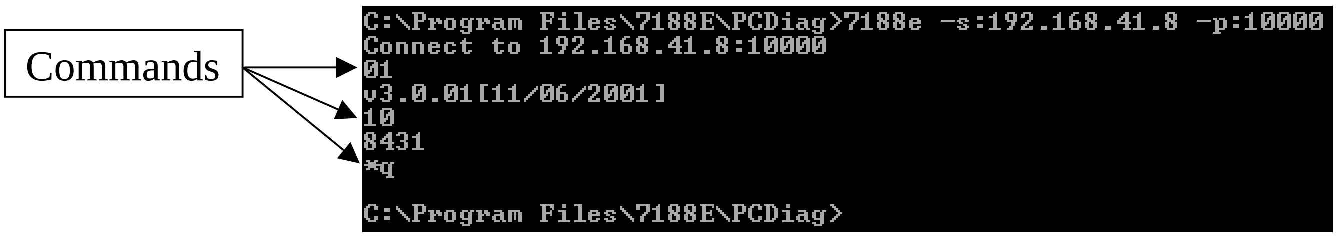
Please refer to sec. 4.6 for more operating details.

➢ **7188e:** Command-prompt mode program, used to send data to specific machines using TCP protocol.

Usage:

7188e [-S:IP] [-P:Port] → Connect to a device by using TCP protocol.

*Q → Quit program and disconect.



```
C:\Program Files\7188E\PCDiag>7188e -s:192.168.41.8 -p:10000
Connect to 192.168.41.8:10000
01
v3.0.01[11/06/2001]
10
8431
*q

C:\Program Files\7188E\PCDiag>
```

2. MiniOS7 of the 8000 Series

2.1 MiniOS7 for the 8000 Series

The MiniOS7 is an embedded O.S. designed for the following families:

- 7188XA/7188XB/7188XC series
- 7521/7522/7523 series
- 7188EA/7188EX/7188EX-256 series
- 7188E1/7188E2/7188E3/7188E4/7188E5/7188E8 series
- 8000 series.
- Iview-100 series
- More new embedded controller families

Several brands of DOS have been created by various companies. In all cases, DOS, whether PC-DOS, MS-DOS, or ROM-DOS, is a set of commands or code which tells the computer how to process information. DOS runs programs, manages files, controls information processing, directs input and output, and performs many other related functions. The MiniOS7 provides equivalent functions of ROMDOS and provides more specific functions for the 7188X/7521/8000 family.

Comparison between MiniOS7 and ROM-DOS:

Function	**MiniOs7**	**RomDos**
Power up time	0.1 sec	4 ~ 5 sec
Supports I/O expansion bus	Yes	No
Supports AsicKey	Yes	No
Supports hardware unique serial number	Yes	No
Supports MMI, Iview-100 series	Yes	No
Supports Ethernet 10M interface, 7188E & 8X3X series	Yes	No
Directly downloads executable programs into Flash ROM	Yes	No
O.S. updateable (downloadable)	Yes	No
Built-in hardware diagnostic functions	Yes	No
Directly controls 7000 series modules	Yes	No
Customers ODM functions	Yes	No
Free of charge	Yes	No

Note: We reserve the right to change the specifications of MiniOS7 without notice.

Command Sets of MiniOS7:

Command	**Description**
LED5 pos value	Shows a HEX value in the specified position of 5-digit LED.
USE NVRAM	Into the service routine for reading/writing NVARM.
USE EEPROM	Into the service routine for reading/writint EEPROM.
USE Flash	Into the service routine for reading/writing Flash-ROM.
USE COM0 /option	Into the service routine for sending/receiving to/from COM0 (8000's back plane: RS-232) to communicate with 87K modules.
USE COM2 /option	Into the service routine for sending/receiving to/from COM2 (RS-485) to communicate with 7000 and 87K modules.
DATE [mm/dd/yyyy]	Sets the date of RTC.
TIME [hh:mm:ss]	Sets the time of RTC.
MCB	Tests current memory block.
UPLOAD	The first step in updating the MiniOs7.
BIOS1	The last step in updating the MiniOs7.
LOAD	DOWNLOADs the user program into the Flash-Memory.
DIR [/crc]	Shows the information of all files downloaded into the Flash-Memory.
RUN [fileno]	Runs the file with file-number=fileno, no fileno→the last file.
Name	Runs the file with file-name=name.
DELETE (or DEL)	Deletes all files stored in the Flash-Memory. It will delete all files.
RESET	Resets the CPU.
DIAG [option]	Hardware Diagnostic.
BAUD baudrate	Sets the new value of communication-baudrate to baudrate.
TYPE filename [/b]	Lists content of the file.
REP [/#] command	Repeats execution of the same command # times.
RESERVE [n]	Reserves n Flash Memory sectors for USER's programs.
LOADR	Downloads a file into SRAM.
RUNR [param1 [param2...]]	Runs a program saved into SRAM (downloaded by command LOADR).
I/INP/IW/INPW port	Reads data from the hardware PORT.
O/OUTP/OW/OUTPW port value	Outputs to hardware PORT.
… more …	… more …

*** Refer to **8000\Common\MiniOS7\DOC\index.htm** for user's manual & demo programs for the MiniOS7 ***

There are some of the libraries included with the 8000E as follows:

8000L.LIB: CPU & I/O related library (Large model)
TCPIPL.LIB: TCP/IP related library (Large model)
XS8_NNNN.LIB: Xserver related library (Large model), with NNNN being the version of lib.

Some of the libraries supported by 8000L.LIB are given as follows:

Function description	Example
COM port	InstallCom1, InstallCom2, ……, InstallCom4 IsCom1, IsCom2,………………, IsCom4 ToCom1, ToCom2, ……, ToCom4 ReadCom1, ReadCom2, ………, ReadCom4
EEPROM	WriteEEP, ReadEEP, EnableEEP, ProtectEEP
NVRAM & RTC	ReadNVRAM, WriteNVRAM, GetTime, SetTime, GetDate, SetDate
LED & 5DigitLed	LedOn, LedOff, LedToggle, Init5DgitLed, Show5DigitLed, Show5DigitLedWithDot
Flash Memory	FlashReadId, FlashErase, FlashRead, FlashWrite
Timer & Watchdog Timer	TimerOpen, TimerClose, TimerResetValue, TimerReadValue StopWatchReset, StopWatchRead, StopWatchStop InstallUserTimer EnableWDT, DisableWDT, RefreshWDT
File	GeFileNo, GetFileName, GetFilePositionByNo, GetFilePositionByName
Connects to 7000	SendCmdTo7000, ReceiveResponseFrom7000
Programmable I/O	SetDio4Dir, SetDio4High, SetDio4Low, GetDio4
Others	Kbhit, Getch, Putch, LineInput, Scanf

Refer to **8000\843x883x\document\8000Ehh.pdf** &
8000\843x883x\document\TCPIPLib.pdf &
8000\843x883x\document\WebLib.pdf &
8000\Common\minios7\doc\index.htm for more information.

Refer to **8000\843x883x\document\TCPIPLib.pdf** for how to compile & link.

2.2 Demo Programs for the 8000E Series

We provide hundreds of demo programs for users. The source codes are all in the shipped CD. It is recommended to edit & modify these demo programs when starting user's special applications. The demo programs can be classed as follows:

Some of the demo programs designed for MiniOS7 are given as follows:

Demo	**Description**
Hello	Can run on PC or 8000, just use Print() to print: "*** Hello 8000 ***"
Hello1	Demo for using functions: Is8000, GetLibVersion, Print
Hello2	Demo for using C++ compiler.
FILE	Demo information in obtaining file information and file position in Flash memory. All file data is stored in Flash Memory. In MiniOS7, cannot use C's functions: fopen, fclose, fread, fwrite.
BATCH	An example of BATCH files (*.BAT).
SCANF	Demo for using LineInput and Scanf.
RUNPROG	Uses Ungetch to send commands to MiniOS7 to run another program
DEMO90-98	Demos for using TIMER functions.

Location: 8000\Common\MiniOS7\Demo\*.*

Some typical TCP/IP demo programs are given as follows:

Typical TCP/IP demo	**8000E**	**PC**
Ping demo	Client, ping.exe	None
Telnet server demo	Server, telserv.exe	Client, telnet.exe
Telnet server demo2	Server, telserv2.exe	Client, telnet.exe
Demo1: TCP/IP demo	Server, demo1.exe	Client, Client1.exe
Demo2: TCP/IP demo	Server, demo2.exe	Client, Client1.exe
Demo3: TCP/IP demo	Server, demo3.exe	Client, Client1.exe

Location: 8000\843x883x\TCP\Other\*.*

Refer to Sec. 4.7 for more TCP/IP demo program designed for Xserver.
Refer to **8000\843x883x\document\TCPIPLib.pdf**,
8000\common\minios7\doc\index.htm and
8000\843x883x\document\WebLib.pdf for more information.
Refer to **sec. 4.5** for how to compile and link.

3. VxComm Applications

- Overview
- Installing the VxComm Driver
- Adding a 7188E/8000E server and configuring the VxComm Driver
- Removing a 7188E/8000E server
- Uninstalling the VxComm Driver
- Diagnostics and Trouble Shooting
- FAQ

3.1 Overview

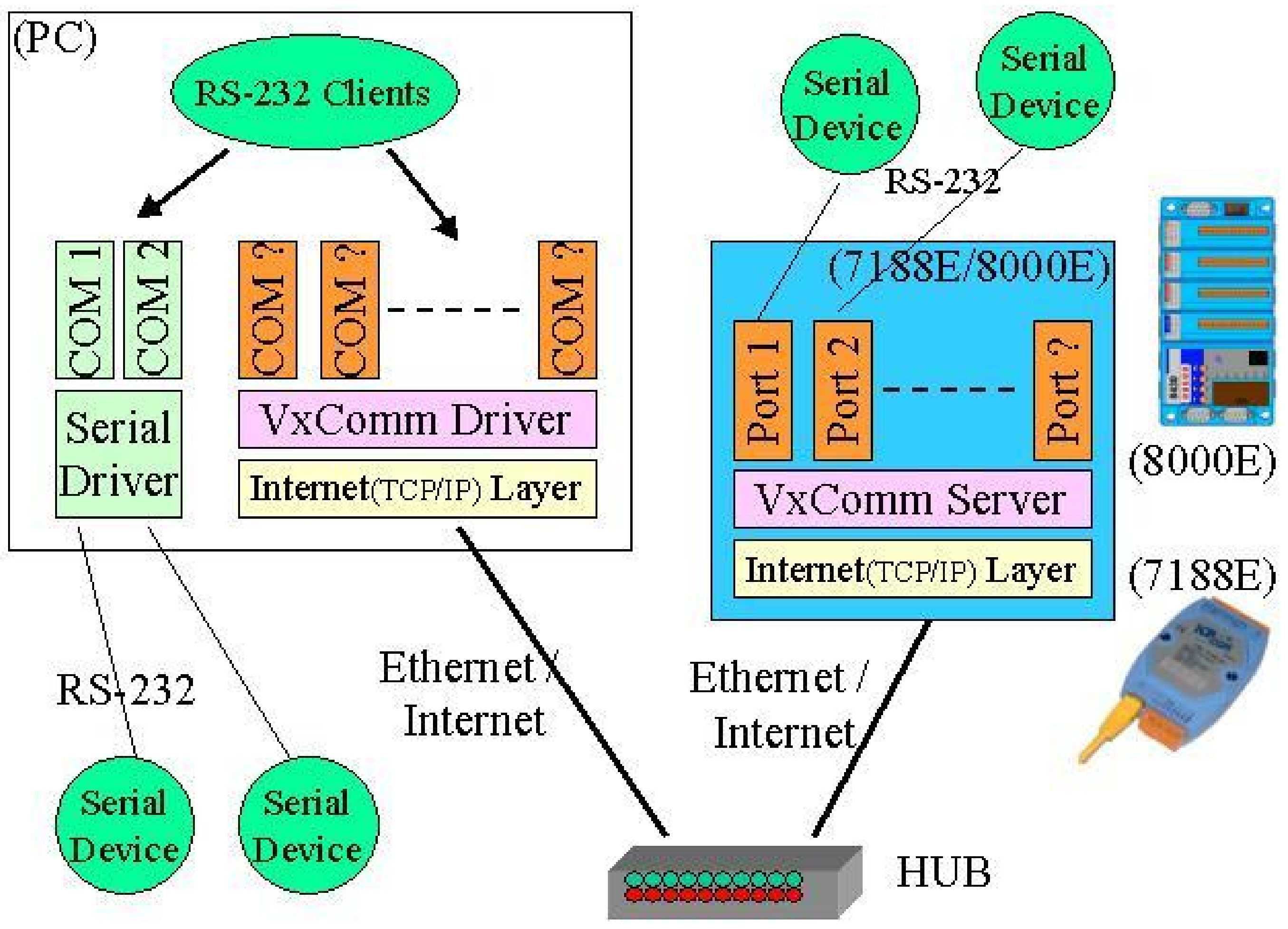
The VxComm (Virtual Comm) Driver and VxComm Utility are very easy to install and use. The first thing to do is to find the installation file in the included CD. The directory is:

- ➢ 8000\843x883x\TCP\VxComm\NT\VxCommNT.exe
  (for Windows NT 4.0) or
- ➢ 8000\843x883x\TCP\VxComm\2K\VxComm2K.exe
  (for Windows 2000, Windows XP).

This document shows how to install and configure the driver correctly. There are three parts to the quick start manual. The first part instructs users how to install the software. The second part shows how to add a 7188E/8000E server and configure a COM port. Finally, the third part teachs you how to remove a 7188E/8000E server.

3.1.1 Architecture

The VxComm Driver creates COM port(s) and maps them to the COM port(s) of the 7188E/8000E. The user's RS-232 client programs need only to change to the different COM port to access the serial devices that are allocated to the Internet or Ethernet network via the 7188E/8000E.



3.1.2 Port mapping

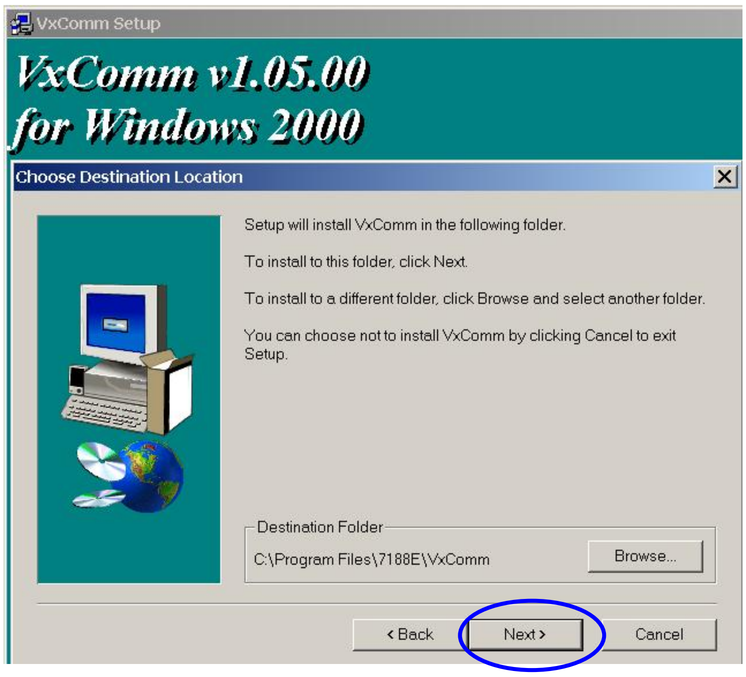
Vxcomm Driver/Utility supports Port 1 to Port 8 in accessing COM1 to COM8 of the 7188E/8000E. Another Port I/O is designed to access the I/O boards mounted on 7188E or 8000E.
With the help of the VxComm Driver/Utility, uses can map remote COM port and I/O boards to become a virtual COM port of PC. One PC can control a maximun of 256 COM ports (including COM1 and COM2).

Local COM Port (PC)	VxComm Driver/Utility (PC)	Remote COM port (7188E/8000E)
COM ?	Port 1	COM1
COM ?	Port 2	COM2
COM ?	Port 3	COM3
COM ?	Port 4	COM4
COM ?	Port 5	COM5
COM ?	Port 6	COM6
COM ?	Port 7	COM7
COM ?	Port 8	COM8
COM ?	Port I/O	I/O board

3.2 Installing the VxComm Driver

Step 1: Run VxComm2K.exe (for Windows 2000, Windows XP) or VxCommNT.exe(WindowsNT 4.0) in the packaged CD to start installing.

Step 2: Choose a destination folder.



Setup will install VxComm in the following folder.

To install to this folder, click Next.

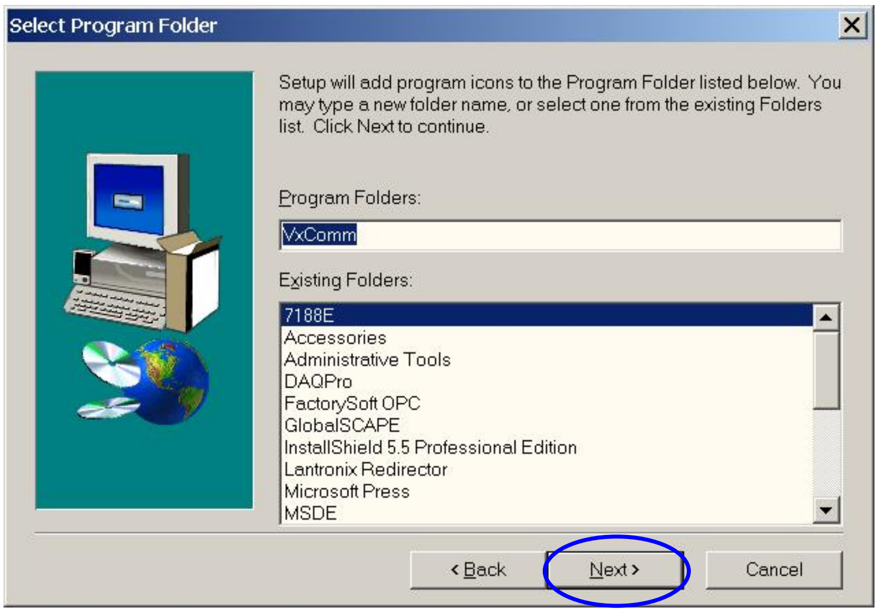
To install to a different folder, click Browse and select another folder.

You can choose not to install VxComm by clicking Cancel to exit Setup.

Destination Folder

C:\Program Files\7188E\VxComm

Step 3: Choose a program folder.



Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing Folders list. Click Next to continue.

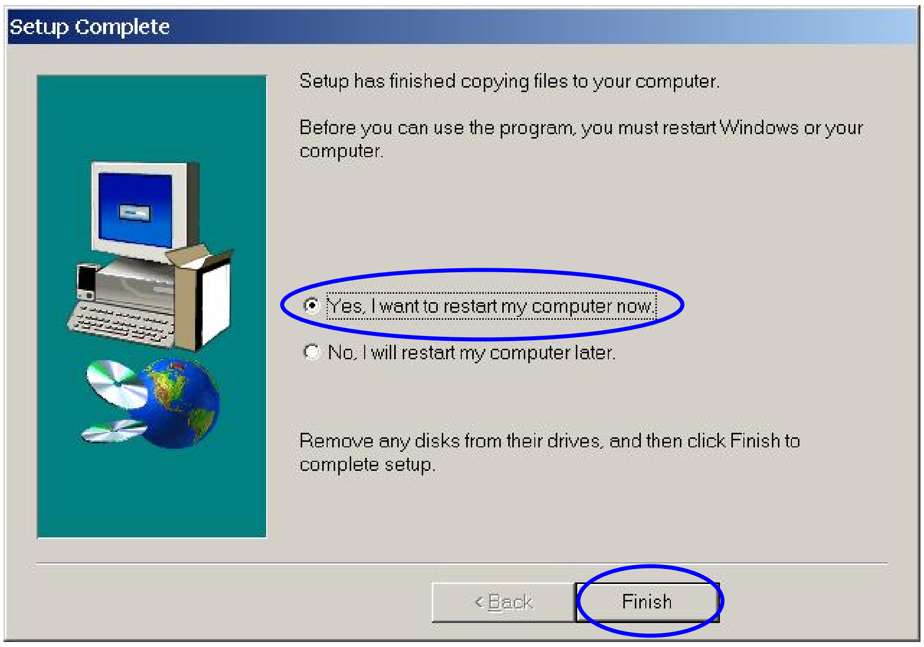
Program Folders:

VxComm

Existing Folders:

7188E
Accessories
Administrative Tools
DAQPro
FactorySoft OPC
GlobalSCAPE
InstallShield 5.5 Professional Edition
Lantronix Redirector
Microsoft Press
MSDE

Step 4: Select the "Yes, ... " option and click the "Finish" button to reboot your computer.


Setup Complete

Setup has finished copying files to your computer.

Before you can use the program, you must restart Windows or your computer.

[✓ Yes, I want to restart my computer now.]

No, I will restart my computer later.

Remove any disks from their drives, and then click Finish to complete setup.

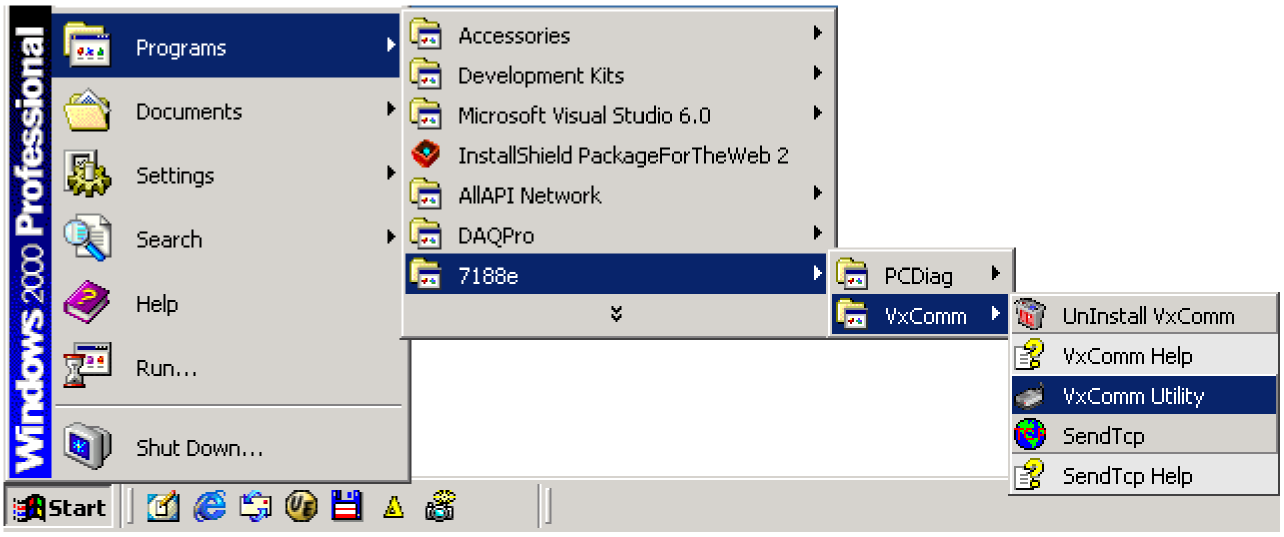
Step 5: After rebooting the computer, the VxComm Utility will ask you to configure the virtual COM port(s). Please refer to the next section for more information.

3.3 Adding a 7188E/8000E server and configuring the VxComm Driver

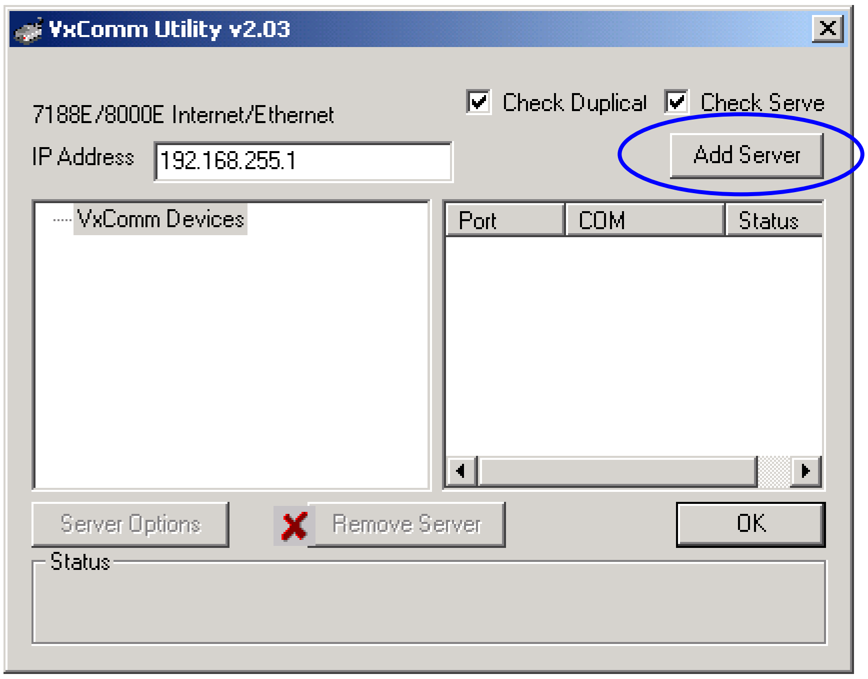
7188E/8000E's default IP address is 192.168.255.1.

Step1: Obtain the IP address of the 7188E/8000E. Either 7188x.exe, 7188xw.exe or Configure wizard can help you in obtaining the IP address of 7188E/8000E. Refer to 8000\843x883x\Document\8000Ehh.pdf for information regarding the use of these three tools.

Step 2: Select the "VxComm Utility".



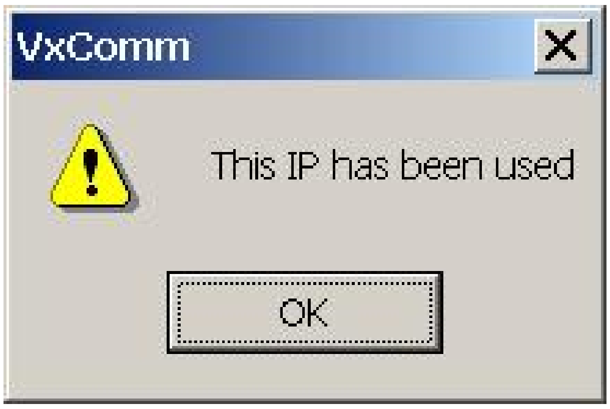
Step 3: Add a 7188E/8000E server IP address and Press the "Add Server" Button.



Note:

"Check Duplicate" option:

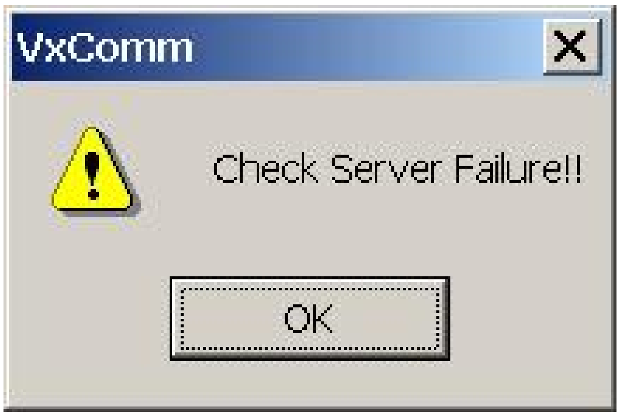
Checks whether the IP address is already listed in the server window (left-hand window). Default is automatically checked. The following window pops up if IP address is duplicated.


VxComm

This IP has been used

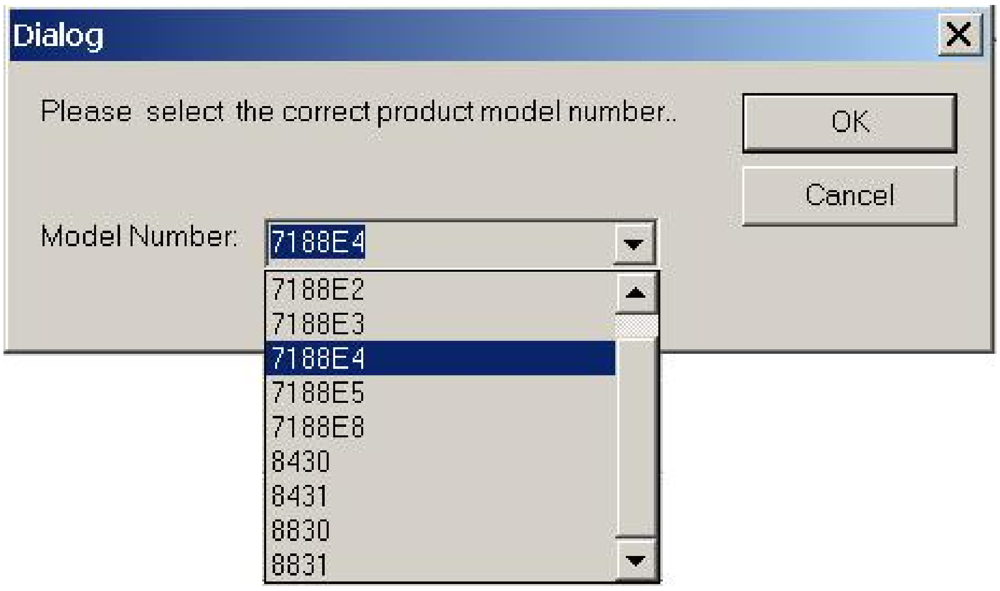
"Check Server" option:

Connects to the 7188E/8000E and gets the device's name before adding to the server window (left-hand window). Default is automatically checked. The following window pops up if the host fails to connect.



Check Server Failure!!

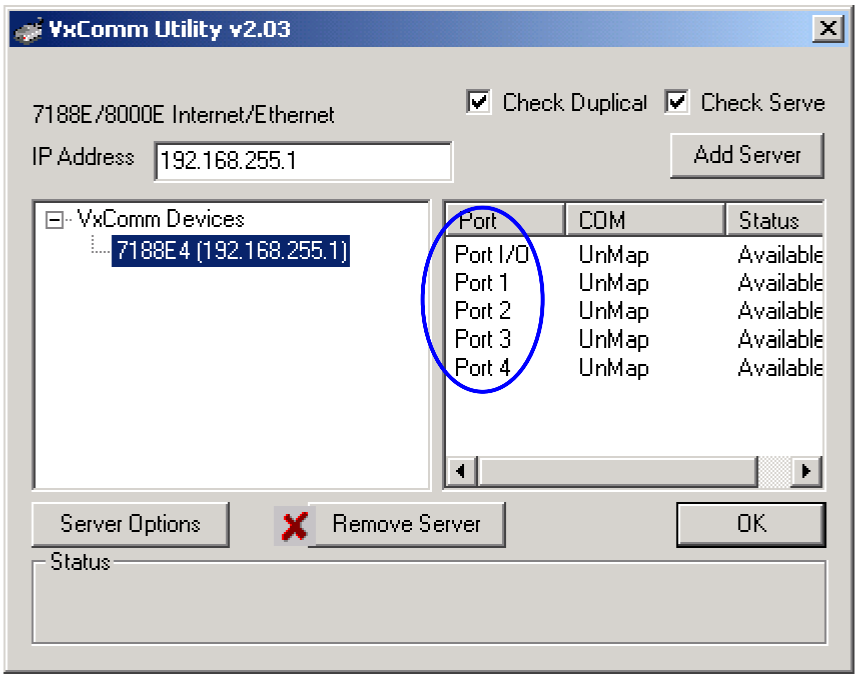
Step 4: The following window pops up if you uncheck the "Check Server" option before pressing the "Add Server" button. Please choose a suitable "Model Number" of 7188E/8000E and then click the "OK" button.



Please select the correct product model number..

Model Number: 7188E4

Step 5: Select one of the 7188E/8000E devices and configure the virtual COM port(s) by double clicking "Port1", "Port2" or etc..


VxComm Utility v2.03

7188E/8000E Internet/Ethernet

☑ Check Duplical ☑ Check Serve

IP Address 192.168.255.1

Add Server

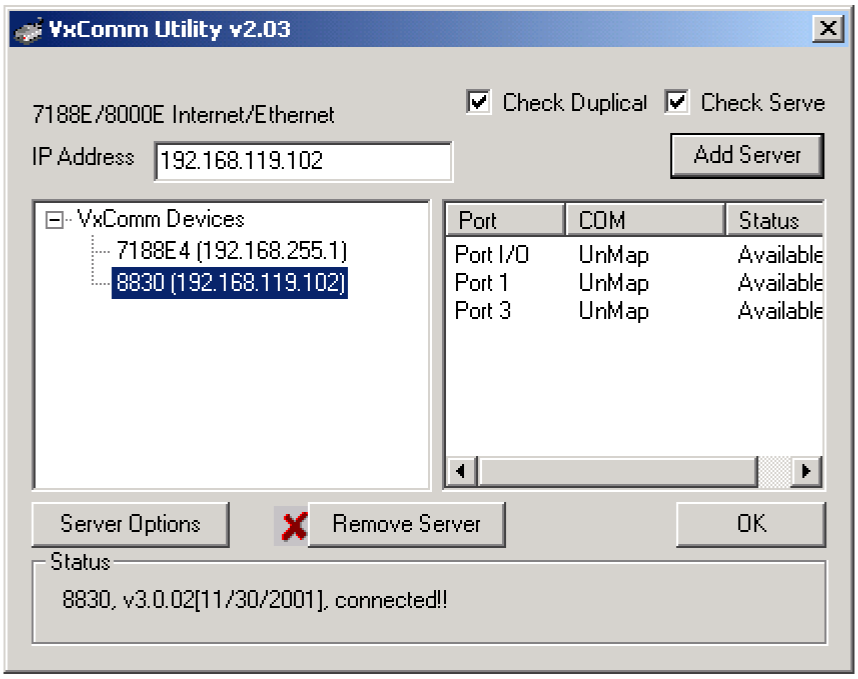
VxComm Devices
- 7188E4 (192.168.255.1)

Port	COM	Status
Port I/O	UnMap	Available
Port 1	UnMap	Available
Port 2	UnMap	Available
Port 3	UnMap	Available
Port 4	UnMap	Available

Server Options | Remove Server | OK

Status

Another example when adding the 8830 at IP address 192.168.119.102.


VxComm Utility v2.03

7188E/8000E Internet/Ethernet

☑ Check Duplical ☑ Check Serve

IP Address 192.168.119.102

Add Server

VxComm Devices
- 7188E4 (192.168.255.1)
- 8830 (192.168.119.102)

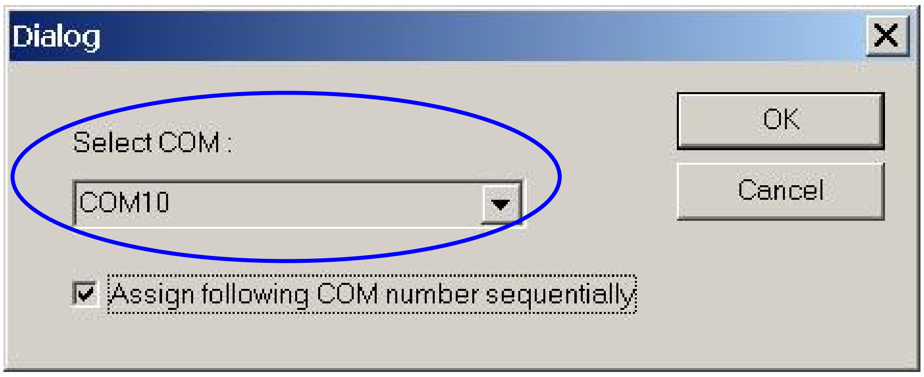
Port	COM	Status
Port I/O	UnMap	Available
Port 1	UnMap	Available
Port 3	UnMap	Available

Server Options | Remove Server | OK

Status

8830, v3.0.02[11/30/2001], connected!!

Step 6: Select an appropriate COM port number, and then click the "OK" button.

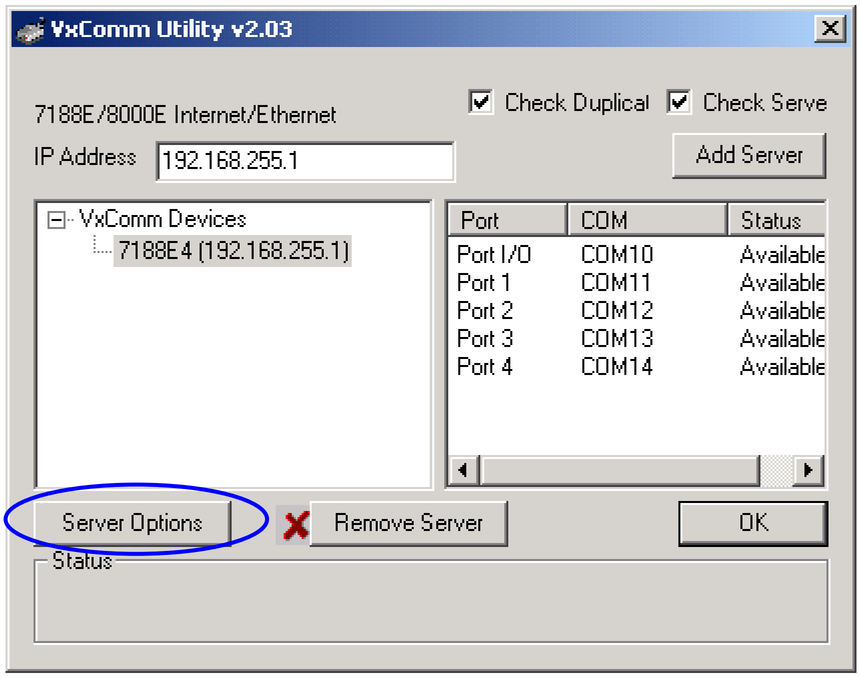


Note:

"Assign following COM number sequentially" option :

Assigns the following ports with the available COM port number sequentially and automatically.

Step 7: Select one of the 7188E/8000E devices and then click the "Server Options" button to configure the options.

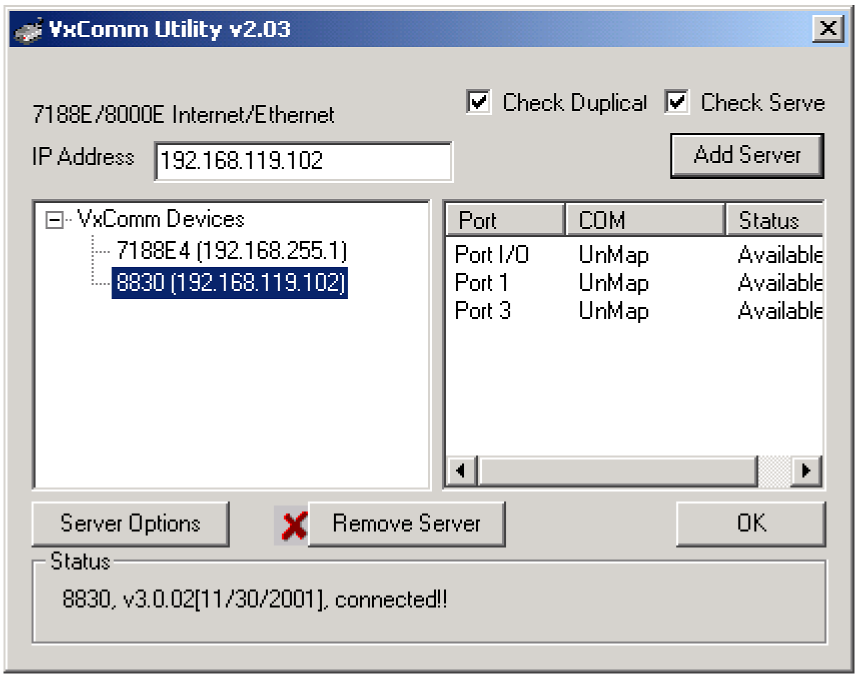


IP Address 192.168.255.1

7188E4 (192.168.255.1)

Port	COM	Status
Port I/O	COM10	Available
Port 1	COM11	Available
Port 2	COM12	Available
Port 3	COM13	Available
Port 4	COM14	Available

Another example of virtualizing the 8830s' COM1 to become PC's COM20.



7188E/8000E Internet/Ethernet

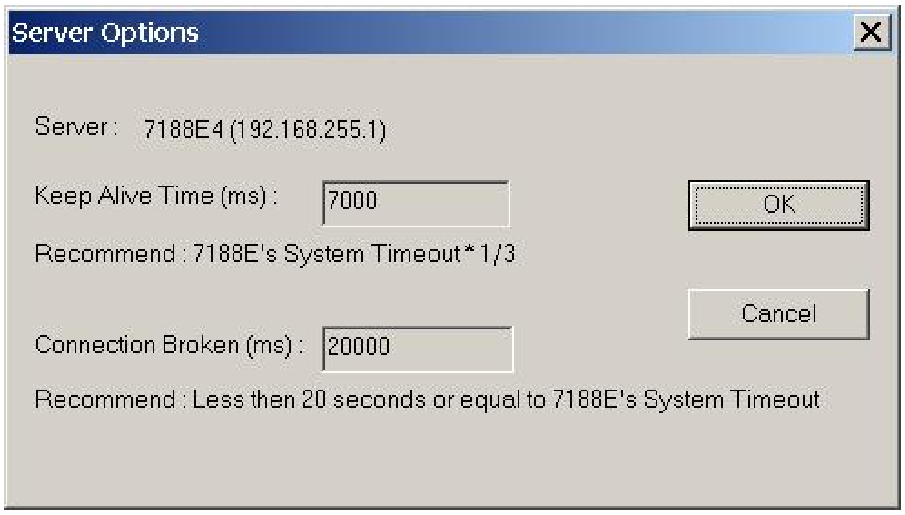
IP Address: 192.168.119.102

VxComm Devices
- 7188E4 (192.168.255.1)
- 8830 (192.168.119.102)

Port	COM	Status
Port I/O	UnMap	Available
Port 1	UnMap	Available
Port 3	UnMap	Available

Status: 8830, v3.0.02[11/30/2001], connected!!

Step 8: Key in the value and then click the "OK" button to exit.



Server : 7188E4 (192.168.255.1)

Keep Alive Time (ms) : 7000

Recommend : 7188E's System Timeout * 1/3

Connection Broken (ms) : 20000

Recommend : Less then 20 seconds or equal to 7188E's System Timeout

Note 1: **Keep Alive Time (ms) field:**

After connecting to the 7188E/8000E, the VxComm Driver will automatically and periodically sends commands to keep the 7188E/8000E alive. The timer will be reset after each sending command or receiving data success. The Keep-Alive mechanism won't work until the next timeout.

The default setting of Keep-Alive time is about 7000 ms. It's recommended setting is (7188E/8000E's System Timeout * 1 / 3) or a smaller value.

Connection-Broken (ms) field:

The VxComm Driver will try to build a new connection when the connection is broken.

When clients send a message to the 7188E/8000E, the Internet (TCP/IP) layer may respond with a "Disconnect" event to the VxComm Driver if it sends the message failure in 20 seconds or later. Users can set a shorter Connection-Broken timeout [for example: 10000 ms (10 seconds)] to force the VxComm Driver to build the connection again and get a quicker response.

If there is no sending/receiving signal in the connection during the Connection-Broken time, the connection will be marked as broken. The VxComm Driver will build the connection again in Connection-Broken time. Thus, the Keep-Alive Time should be shorter than the Connection-Broken time to make the connection come on-line.

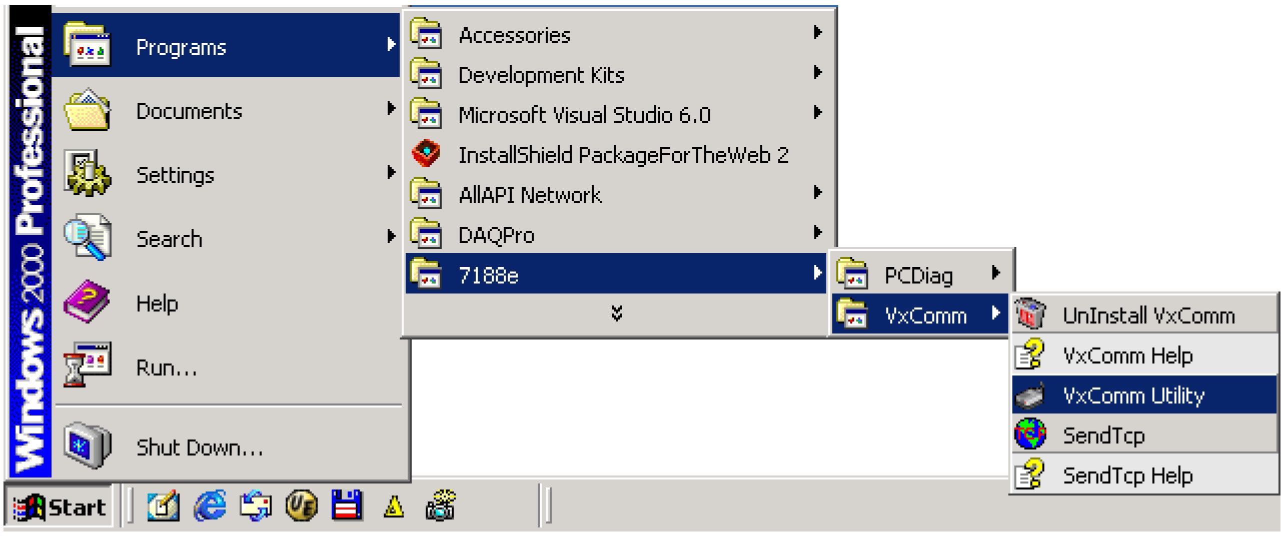
The default System Timeout (/STxxx) value of the 7188E/8000E is about 300 seconds. After client programs connect to the 7188E/8000E, the clients must send commands to keep the 7188E/8000E alive before timeout is up, otherwise the 7188E/8000E will reset itself and clients must build the connection again.

You can disable the Keep-Alive Time and the Connection-Broken mechanisms by setting their value to 0.

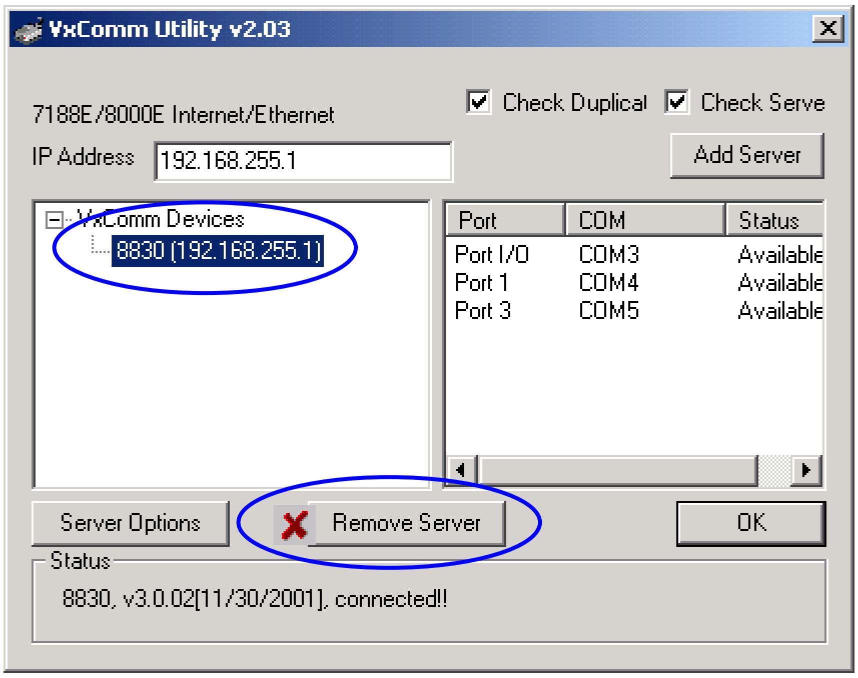
Step 10: Press the "OK" button to save the settings and exit the VxComm Utility.

3.4 Removing a 7188E/8000E server

Step 1: Select the "VxComm Utility".



Step 2: Click the server name you want to remove and press the "Remove Server" button.



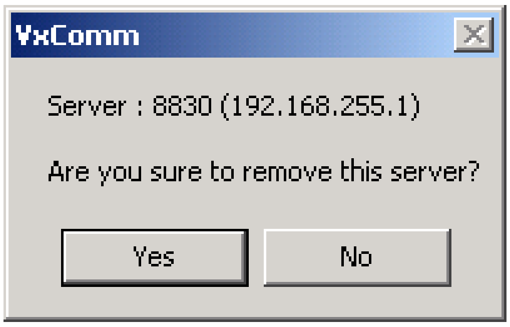
IP Address: 192.168.255.1

8830 (192.168.255.1)

Port	COM	Status
Port I/O	COM3	Available
Port 1	COM4	Available
Port 3	COM5	Available

8830, v3.0.02[11/30/2001], connected!!

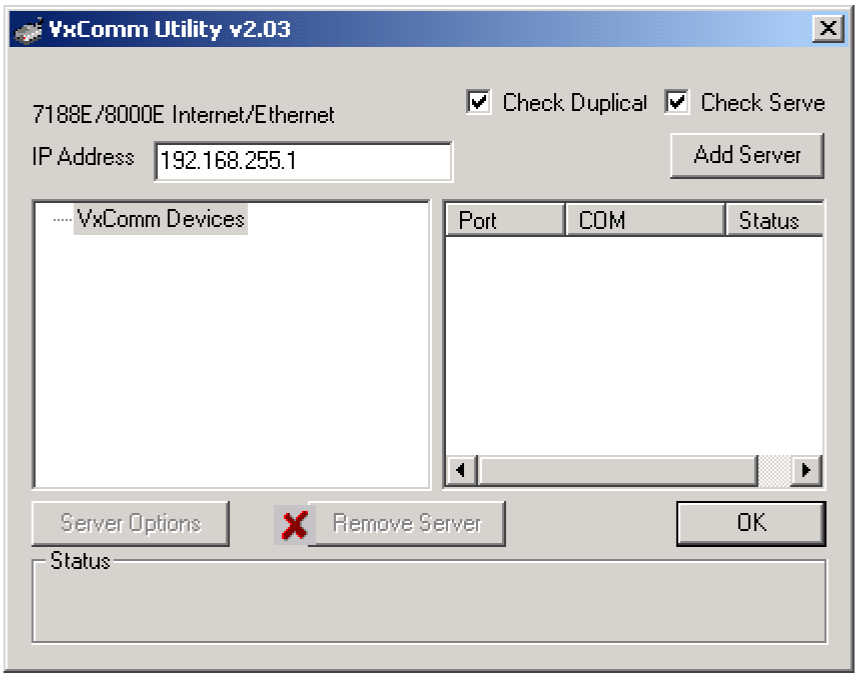
Setp 3: The following window will pop up, please make sure of your choice and press the "Yes" button to remove it.



Server : 8830 (192.168.255.1)

Are you sure to remove this server?

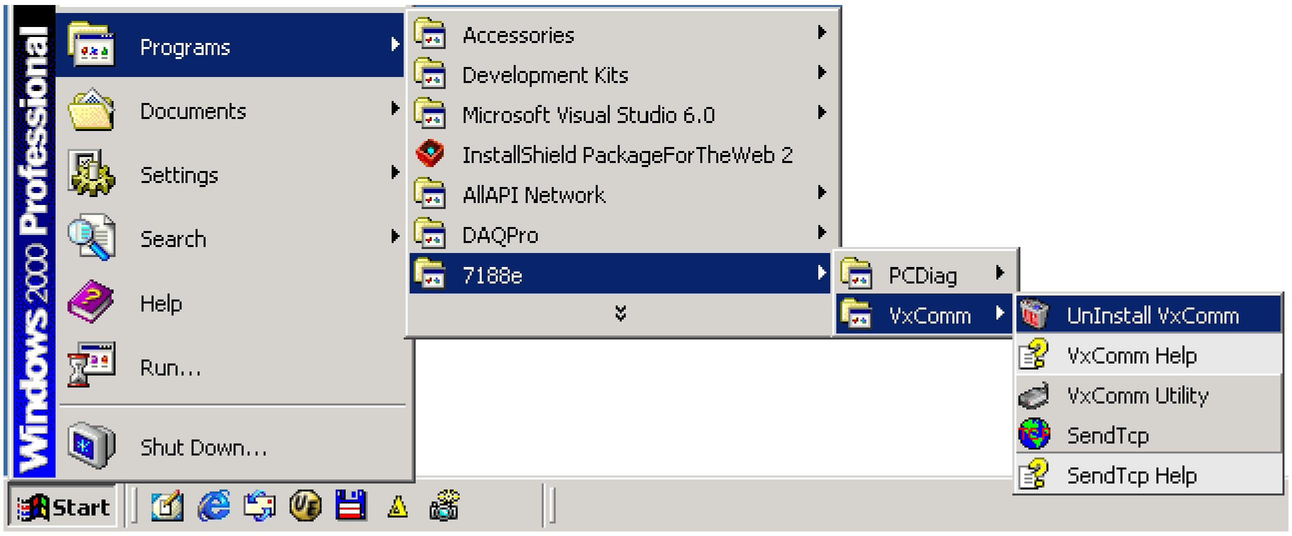
Step 4: Press the "OK" button to finish this utility.



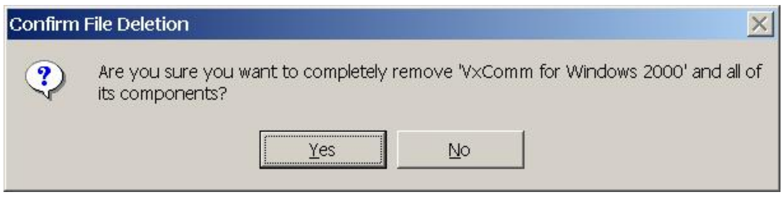
IP Address 192.168.255.1

3.5 Uninstalling the VxComm Driver

Step 1: Select the "UnInstall VxComm".

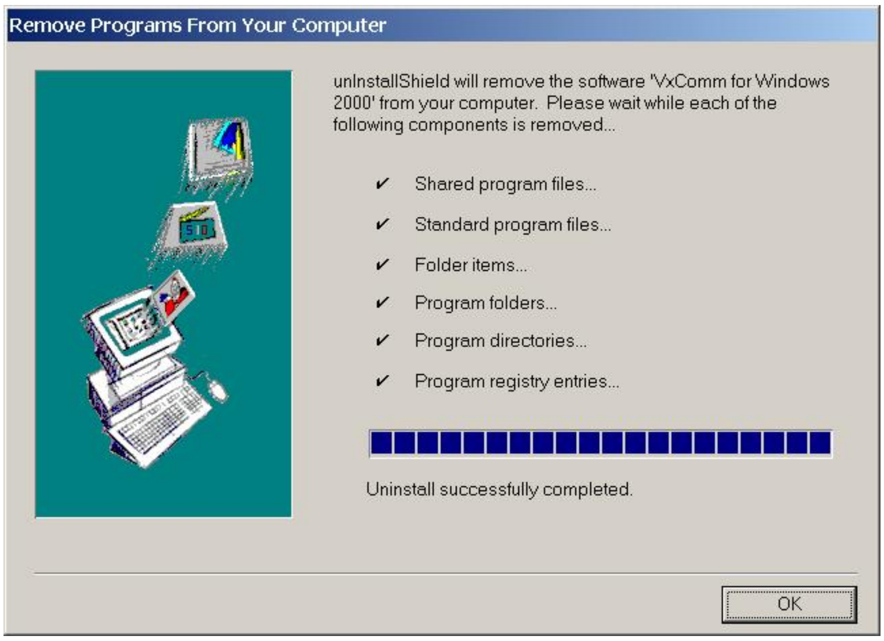


Step 2: Click the "Yes" button.


Confirm File Deletion

Are you sure you want to completely remove 'VxComm for Windows 2000' and all of its components?

Step 3: Click the "OK" button.


Remove Programs From Your Computer

unInstallShield will remove the software 'VxComm for Windows 2000' from your computer. Please wait while each of the following components is removed...

- ✓ Shared program files...
- ✓ Standard program files...
- ✓ Folder items...
- ✓ Program folders...
- ✓ Program directories...
- ✓ Program registry entries...

Uninstall successfully completed.

3.6 Diagnostics and Trouble Shooting

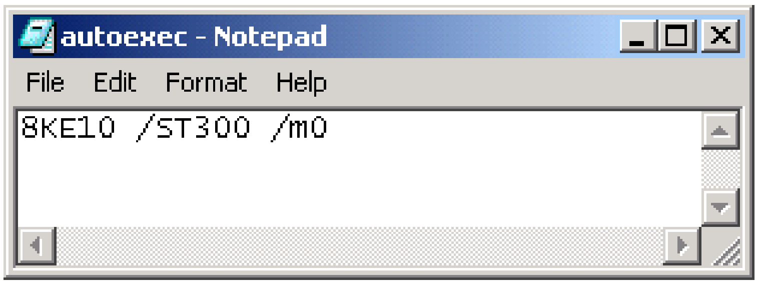
3.6.1 Diagnostics

After configuring the VxComm Driver by using the VxComm Utility, the VxComm Driver should work without error. However, users can use a simple test to make sure it's working properly.

Note: The test method depends on the user's devices and client programs.

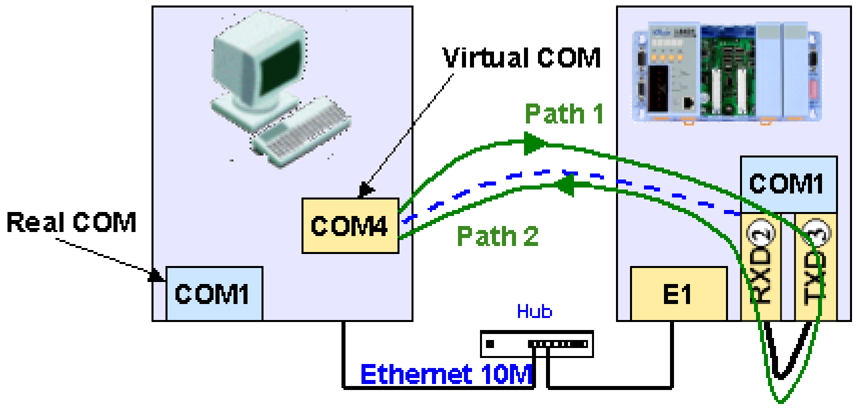
■ **Example 1: Loop-Back Testing**

Step 1: Make sure the VxComm Server is working in /m0 mode.
(Please refer to sec. 4.2.3 "Options of command line")

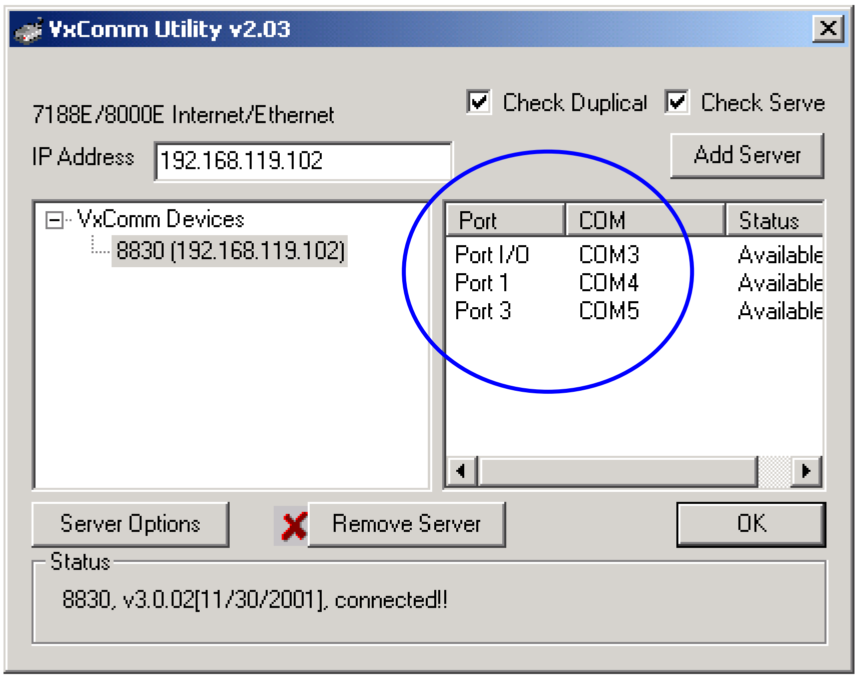


```
8KE10 /ST300 /m0
```

Step 2: Wire the TXD1 with the RXD1 (COM1) of the 7188E/8000E.



Step 3: Virtualize 7188E/8000E's COM1 to become PC's COM4 by using the VxComm Utility.



7188E/8000E Internet/Ethernet

IP Address 192.168.119.102

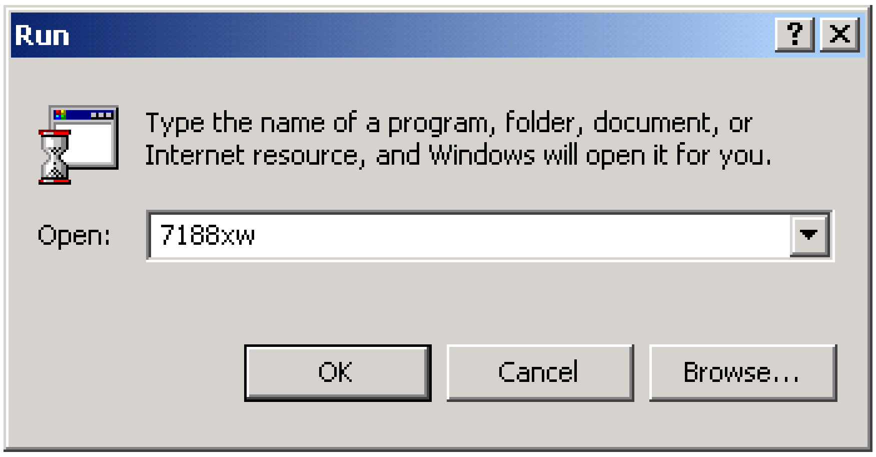
8830 (192.168.119.102)

Port	COM	Status
Port I/O	COM3	Available
Port 1	COM4	Available
Port 3	COM5	Available

Status

8830, v3.0.02[11/30/2001], connected!!

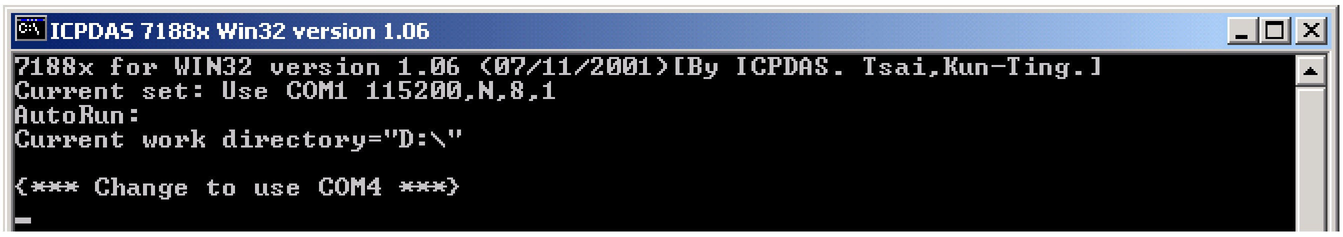
Step 4: Run the 7188xw.exe from the "Start / Run..." menu.



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: 7188xw

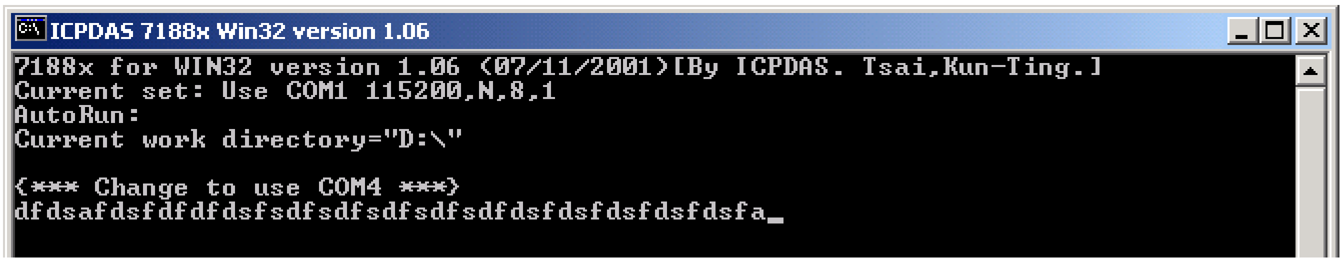
Step 5: Press the <Alt> + <4> keys to use PC's COM4.
It will show "{*** Change to use COM4 ***}" message after changed.



```
7188x for WIN32 version 1.06 (07/11/2001)[By ICPDAS. Tsai,Kun-Ting.]
Current set: Use COM1 115200,N,8,1
AutoRun:
Current work directory="D:\"

{*** Change to use COM4 ***}
_
```

Step 6: Type some characters in the 7188xw.exe window.
The characters will be sent from PC's COM4 to 7188E/8000E's COM1 (through Path1), and immediately returned from the 7188E/8000E's COM1 to the PC's COM4 (through Path2) then shown on the PC's minitor.

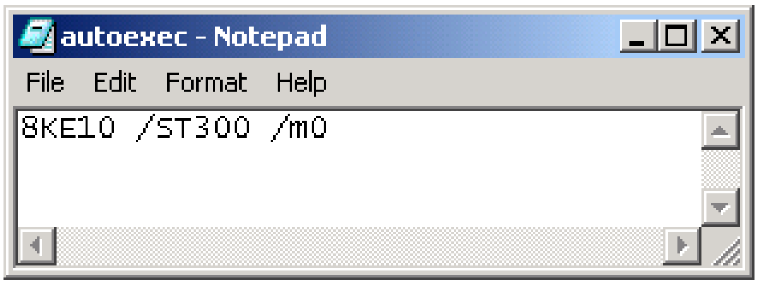


```
7188x for WIN32 version 1.06 (07/11/2001)[By ICPDAS. Tsai,Kun-Ting.]
Current set: Use COM1 115200,N,8,1
AutoRun:
Current work directory="D:\"

{*** Change to use COM4 ***}
dfdsafdsfdfdfdsfsdfsdfsdfsdfsdfdsfdsfdsfdsfdsfa_
```

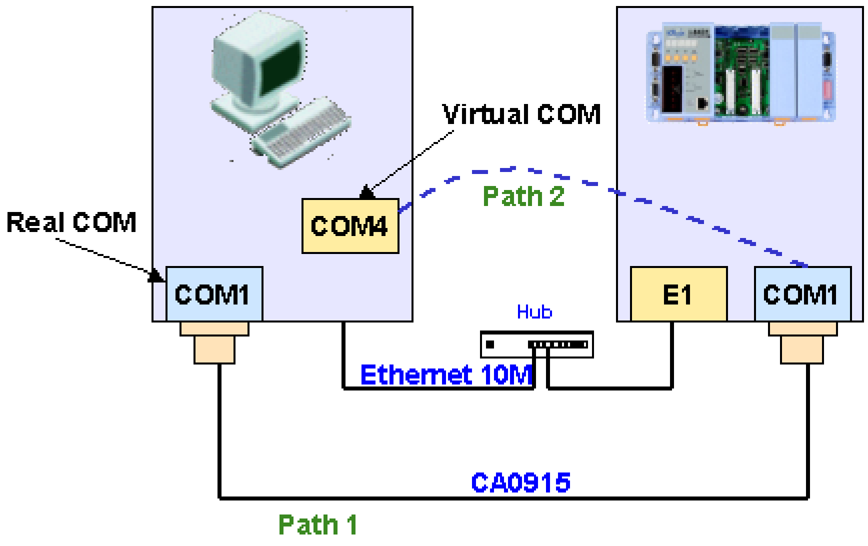
■ Example 2: Close-Loop Testing

Step 1: Make sure the VxComm Server is working in /m0 mode.
(Please refer to sec. 4.2.3 "Options of command line")

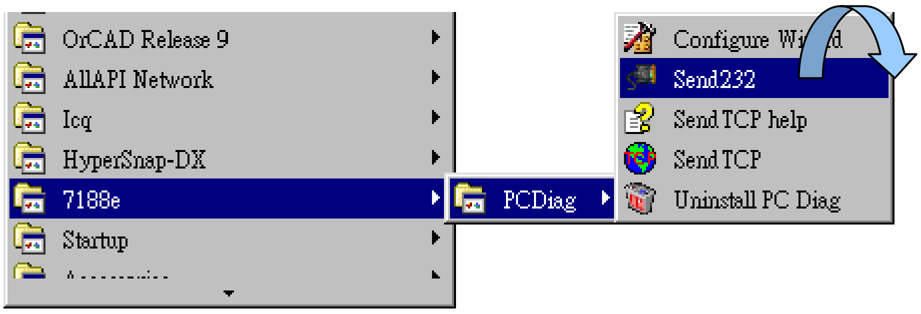


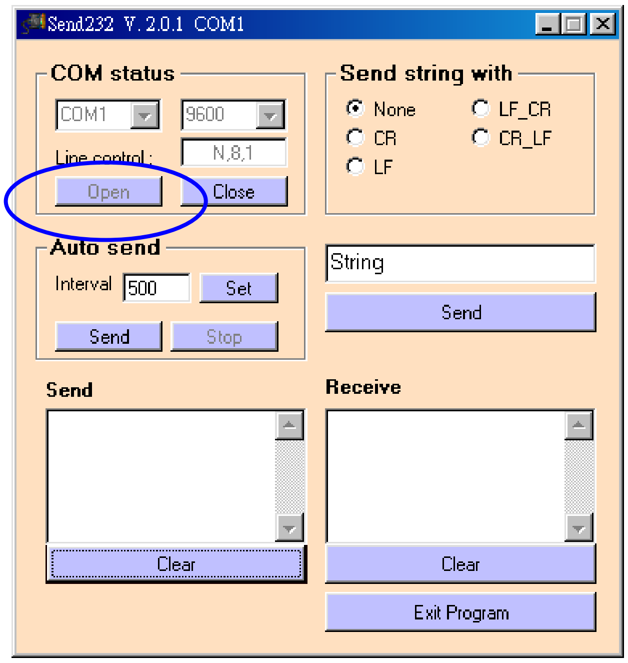
```
8KE10 /ST300 /m0
```

Step 2: Build connection as follows:



Step 3: Run Send232 and then open PC’s COM1.

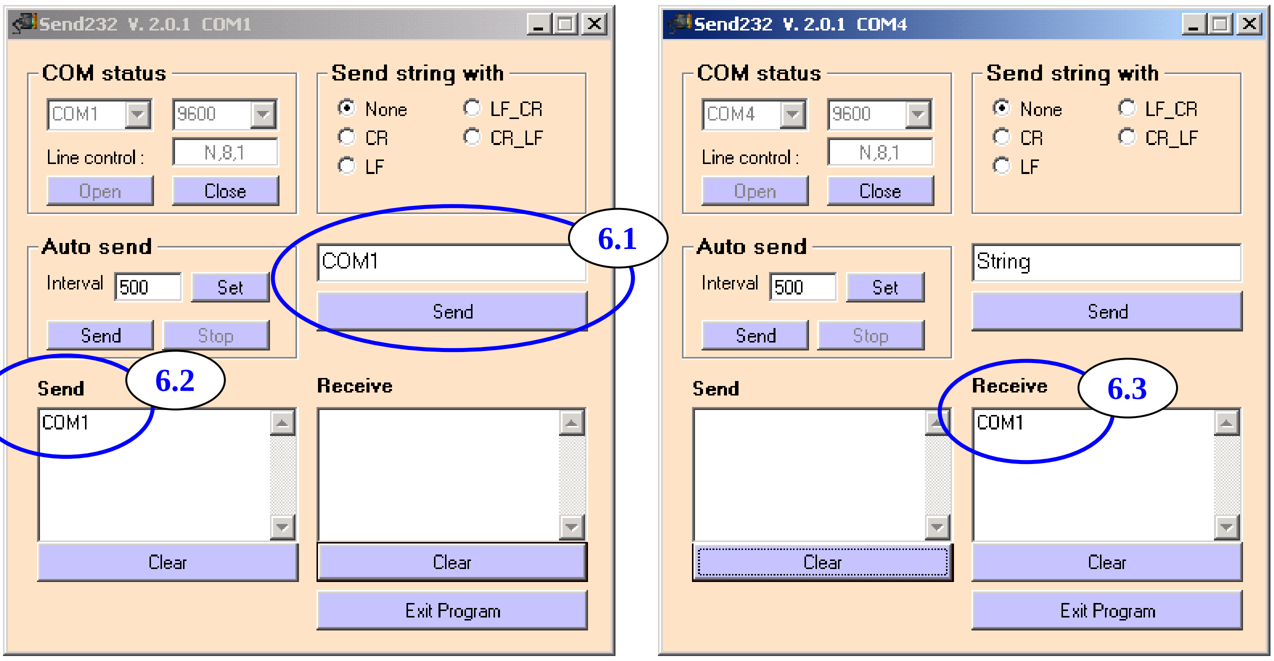




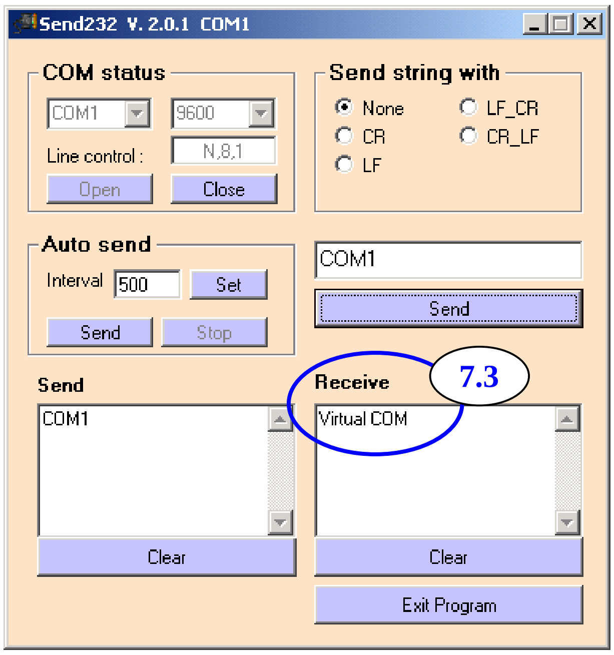
Step 4: Virtualize 7188E/8000E's COM1 to become PC's COM4 by using the VxComm Utility.

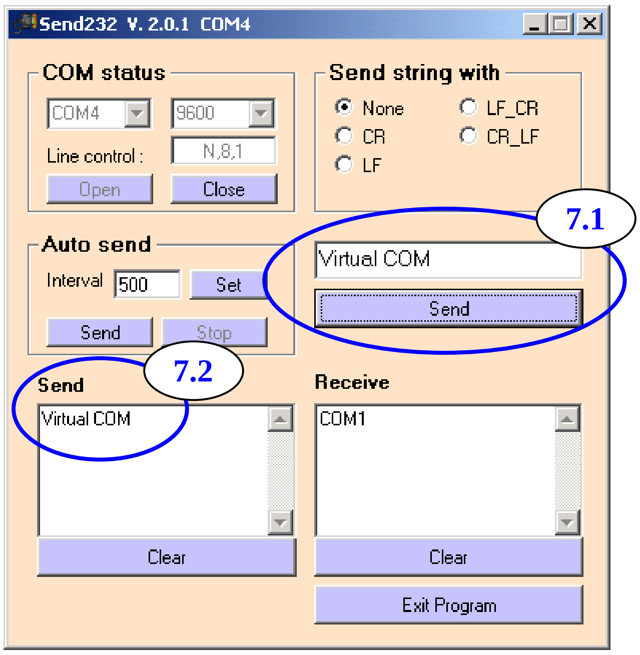
Step 5: Run another Send232 and open PC's virtual COM4.

Step 6: Type "COM1" in left hand window, and press "Send".
Data will be sent from PC's COM1 through Path1 to 7188E/8000E's COM1 and immediately returned through Path2 to PC's COM4.



Step 7: Type "Virtual COM" in right hand window, and press "Send".
Data will be sent from PC's COM4 through Path2 to 7188E/8000E's COM1 and immediately returned through Path1 to PC's COM1.

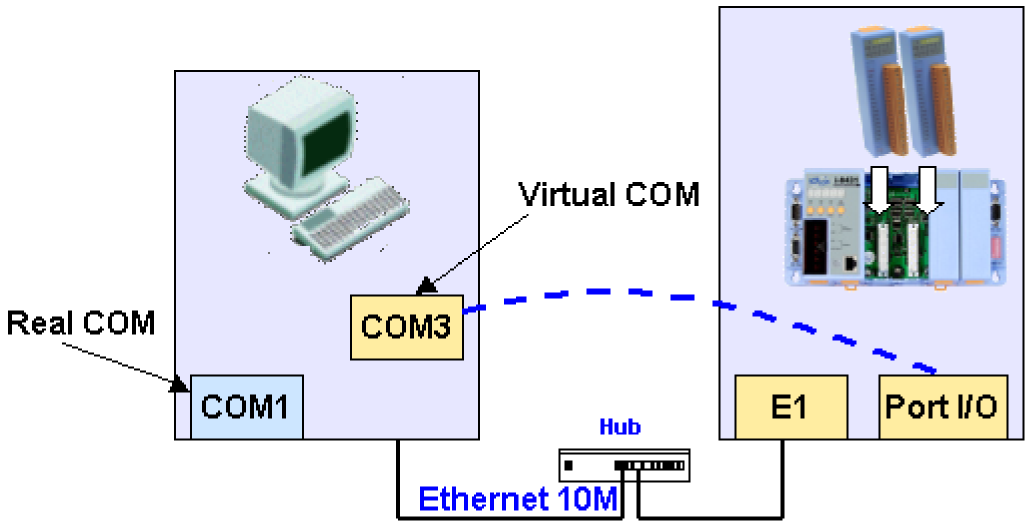




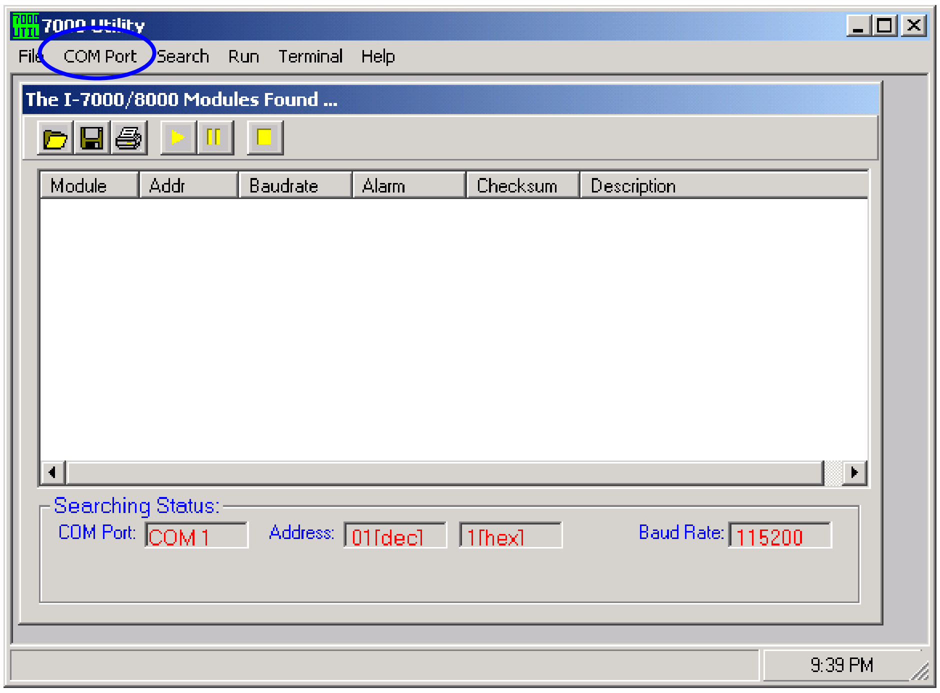
Example 3: Internal-Devices Testing

Step 1: Plug several 8000 modules or 87k modules into a 8000E.

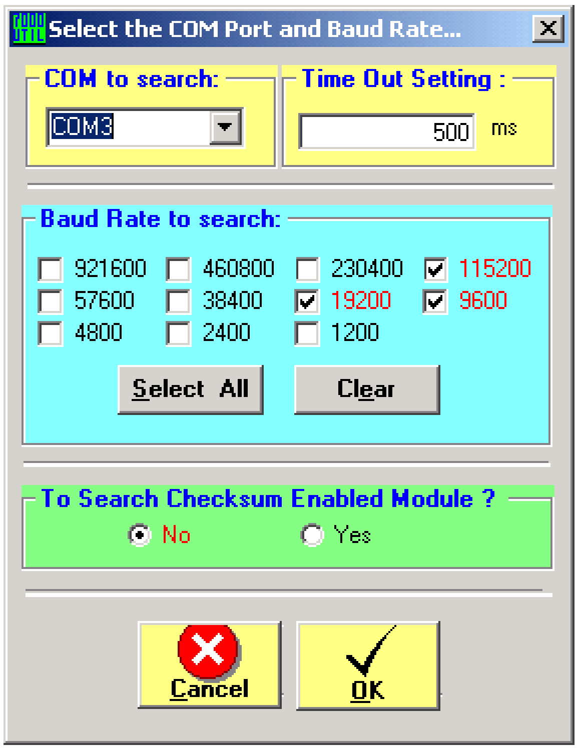
Step 2: Build a connection as follows:



Step 3: Click the menu item "COM Port" to choose the COM port number, baudrate, and checksum. For example: COM3, 115200, 19200, 9600 and No-Checksum. (These settings depend on the 8000 series module's settings.)



COM Port: COM 1 Address: 01[dec] 1[hex] Baud Rate: 115200



COM to search: COM3

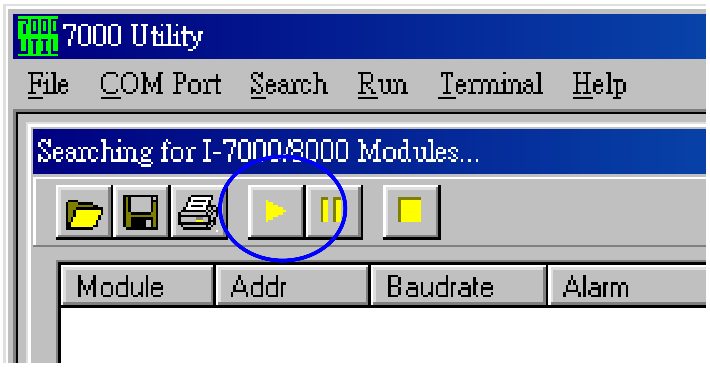
Time Out Setting : 500 ms

Baud Rate to search:

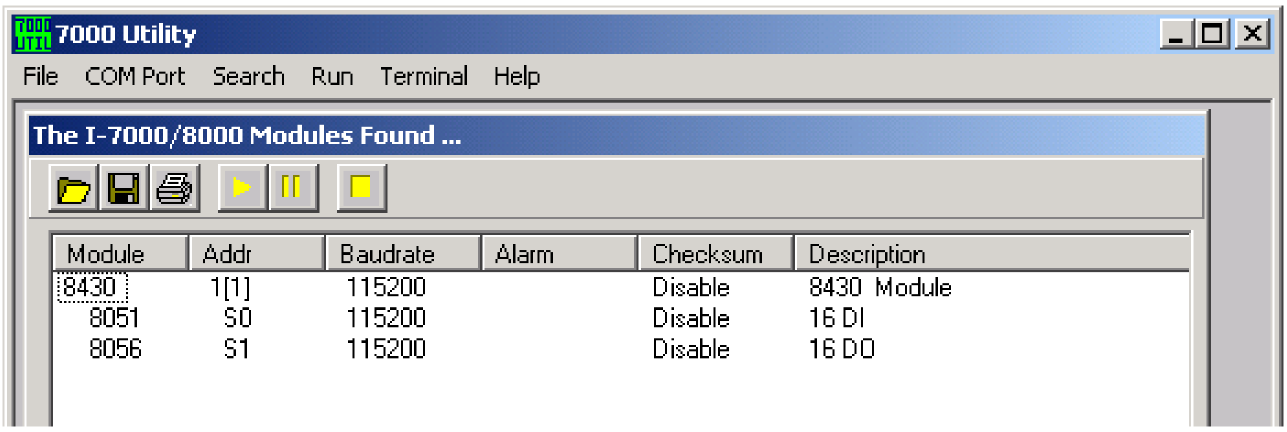
☐ 921600	☐ 460800	☐ 230400	☑ 115200
☐ 57600	☐ 38400	☑ 19200	☑ 9600
☐ 4800	☐ 2400	☐ 1200	

To Search Checksum Enabled Module ? [✓ No] Yes

Step 4: Click the search icon.


7000 Utility
File COM Port Search Run Terminal Help
Searching for I-7000/8000 Modules...
Module Addr Baudrate Alarm

Step 5: If the VxComm Driver works well, 7000 Utility can search the module(s) plugged in the 8000E.


7000 Utility
File COM Port Search Run Terminal Help
The I-7000/8000 Modules Found ...

Module	Addr	Baudrate	Alarm	Checksum	Description
8430	1[1]	115200		Disable	8430 Module
8051	S0	115200		Disable	16 DI
8056	S1	115200		Disable	16 DO

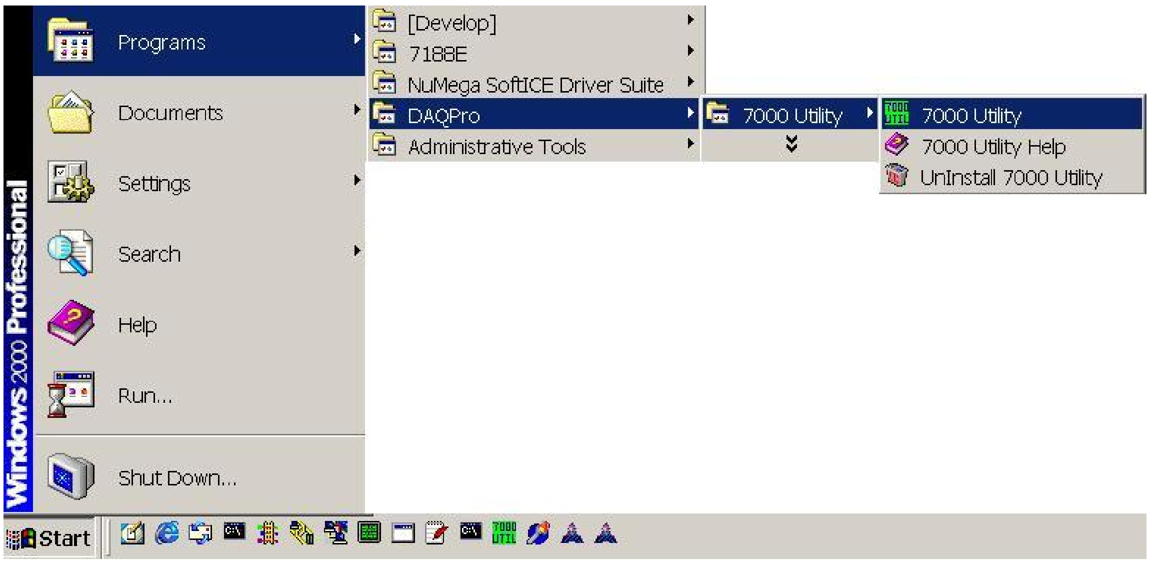
Example 4: External-Devices Testing

Connect 7000 series modules to 7188E/8000E's COM1. Use VxComm Driver to virtualize the 7188E/8000E's COM1 to become PC's COM10. Thus, we can use the 7000 Utility to search the 7000 series module through COM10.

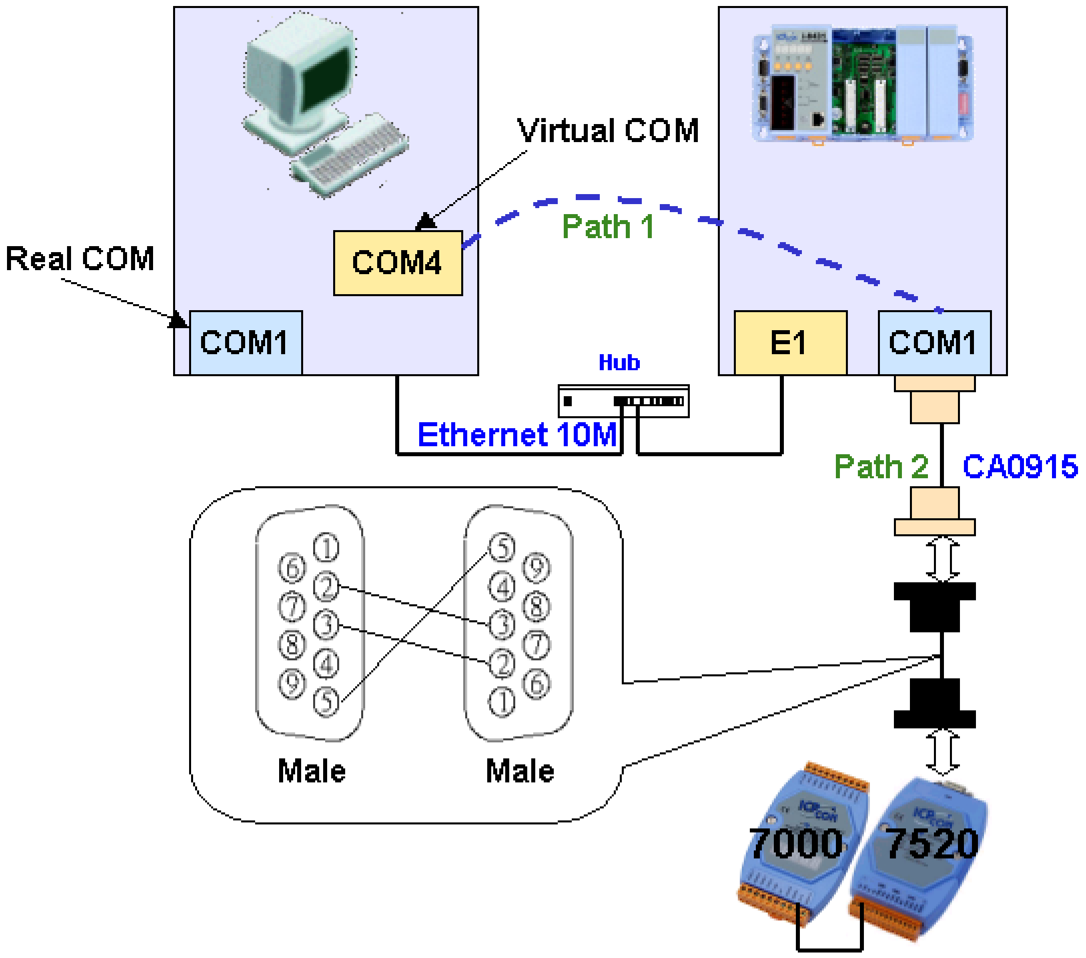
Note: Users must first install the 7000 Utility by runing

CD-ROM Drive:\Napdos\7000\7000Util\setup.exe

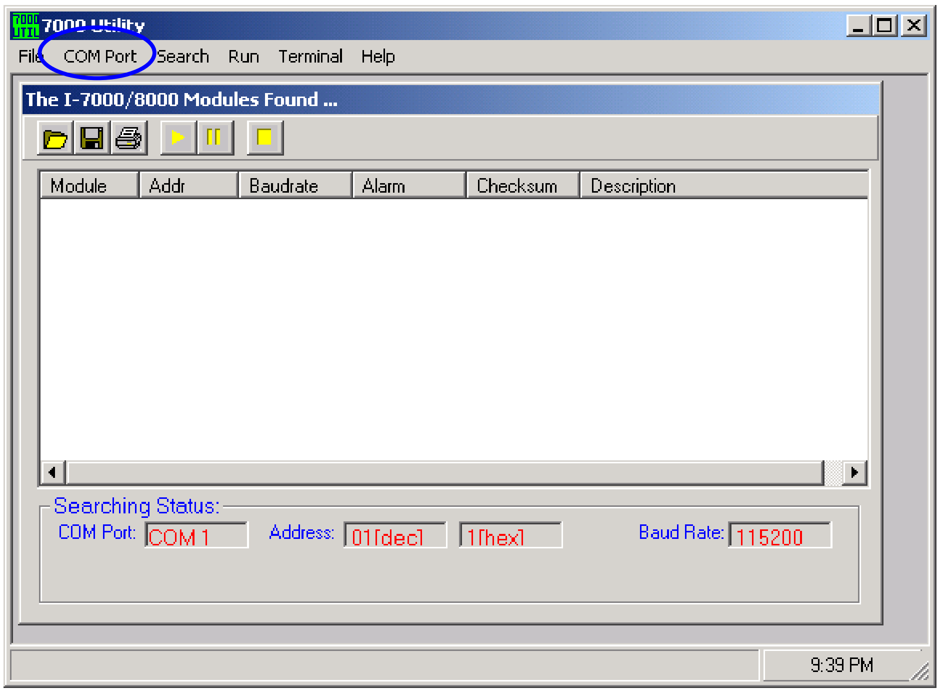
Step 1: Run the 7000 Utility.



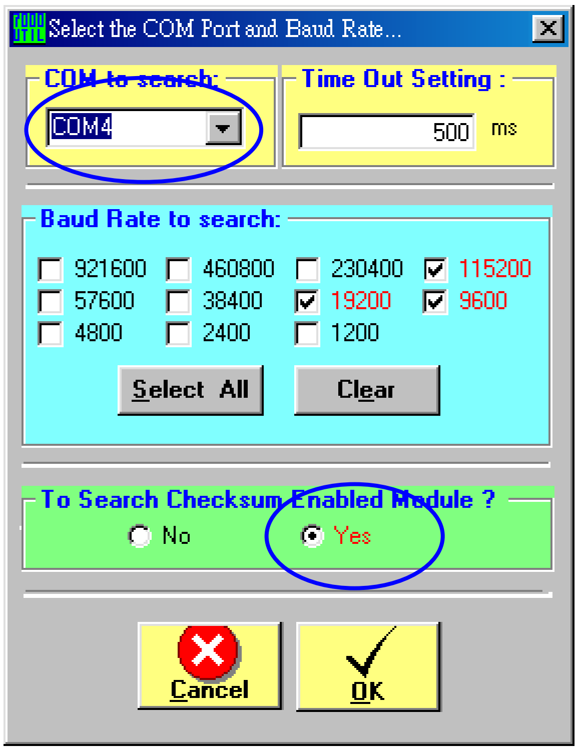
Step 2: Build connection as follows:



Step 3: Click the menu item "COM Port" to choose the COM port number, baudrate, and checksum. For example: COM4, 115200, 19200, 9600 and No-Checksum. (These settings depend on the 7000 series module's settings.)



Searching Status:
COM Port: COM 1
Address: 01[dec] 1[hex]
Baud Rate: 115200



COM to search: COM4

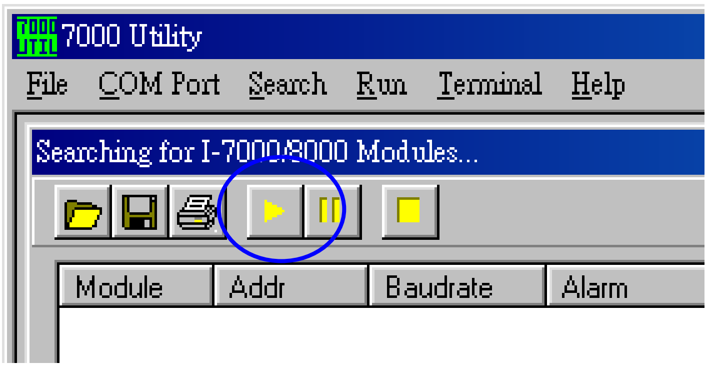
Time Out Setting : 500 ms

Baud Rate to search:

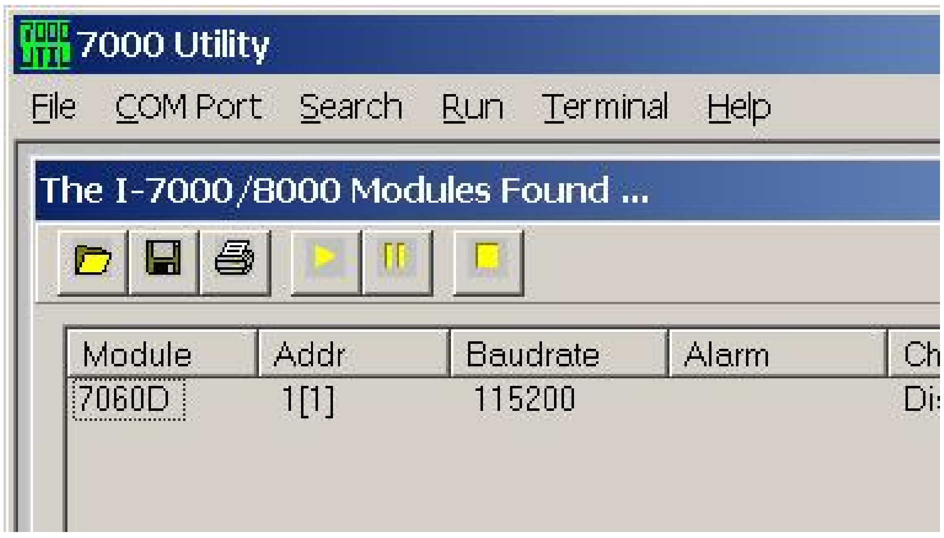
☐ 921600	☐ 460800	☐ 230400	☑ 115200
☐ 57600	☐ 38400	☑ 19200	☑ 9600
☐ 4800	☐ 2400	☐ 1200	

To Search Checksum Enabled Module ? No [✓ Yes]

Step 4: Click the search icon.


7000 Utility
File COM Port Search Run Terminal Help
Searching for I-7000/8000 Modules...
Module Addr Baudrate Alarm

Step 5: If the VxComm Driver works well, the 7000 Utility can search the module(s) connected to the 7188E/8000E's COM1.


7000 Utility
File COM Port Search Run Terminal Help
The I-7000/8000 Modules Found ...

```
Module   Addr   Baudrate   Alarm   Ch
7060D    1[1]   115200             Di
```

3.6.2 Trouble Shooting

Problem: Client program fail to open the COM port that was created by the VxComm Driver.

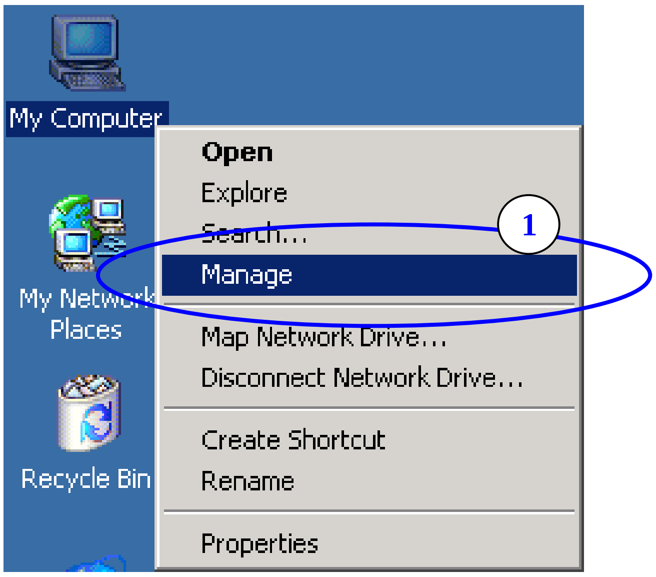
Check:

7188E/8000E's power supply, network cable, IP address, subnet-mask, and gateway. (Please refer to the 7188E/8000E user's manual for more information.)

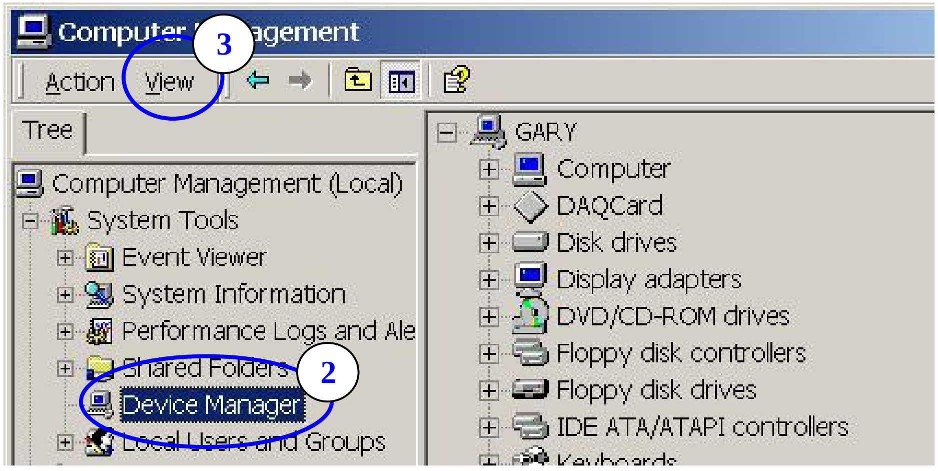
Problem: Client program still fails to open the COM port.

Check:

Step 1: Right click the "My computer" icon and select the "Manage" option.

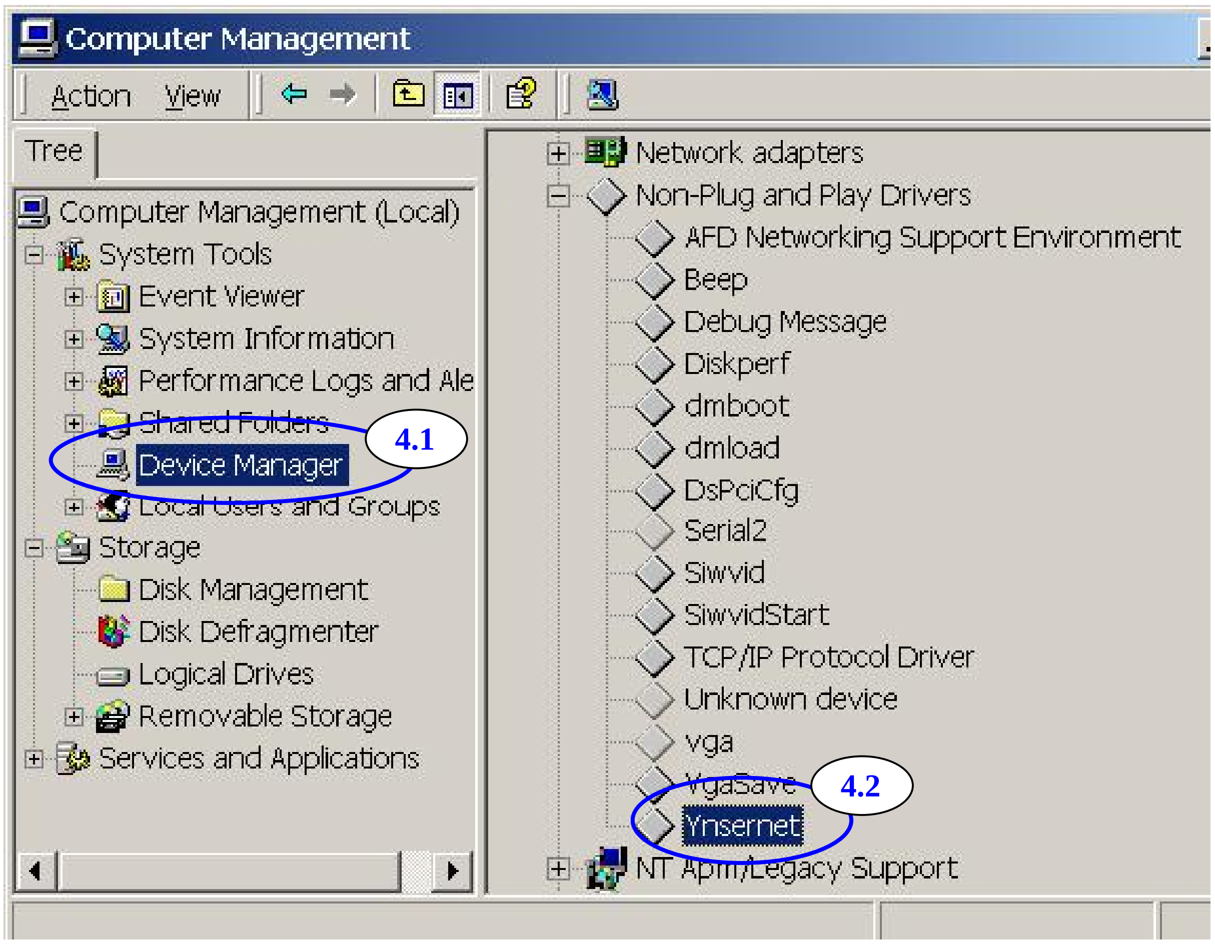


Step 2: Select the "Device Manager" icon from the "Computer Management" program.

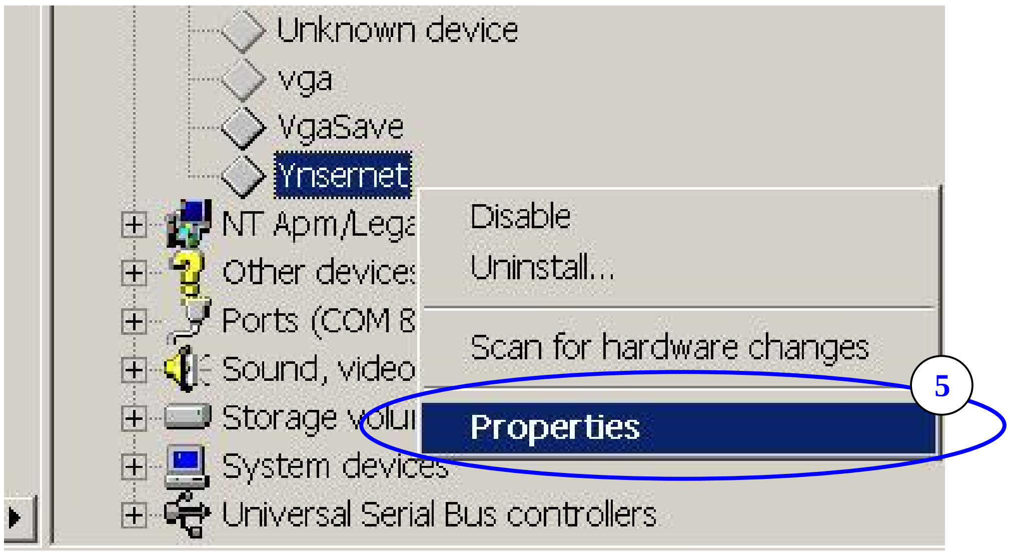


Step 3: Click the menu item "View / Show hidden devices".

Step 4: Select the item "Non-Plug and Play Drivers / Ynsernet".

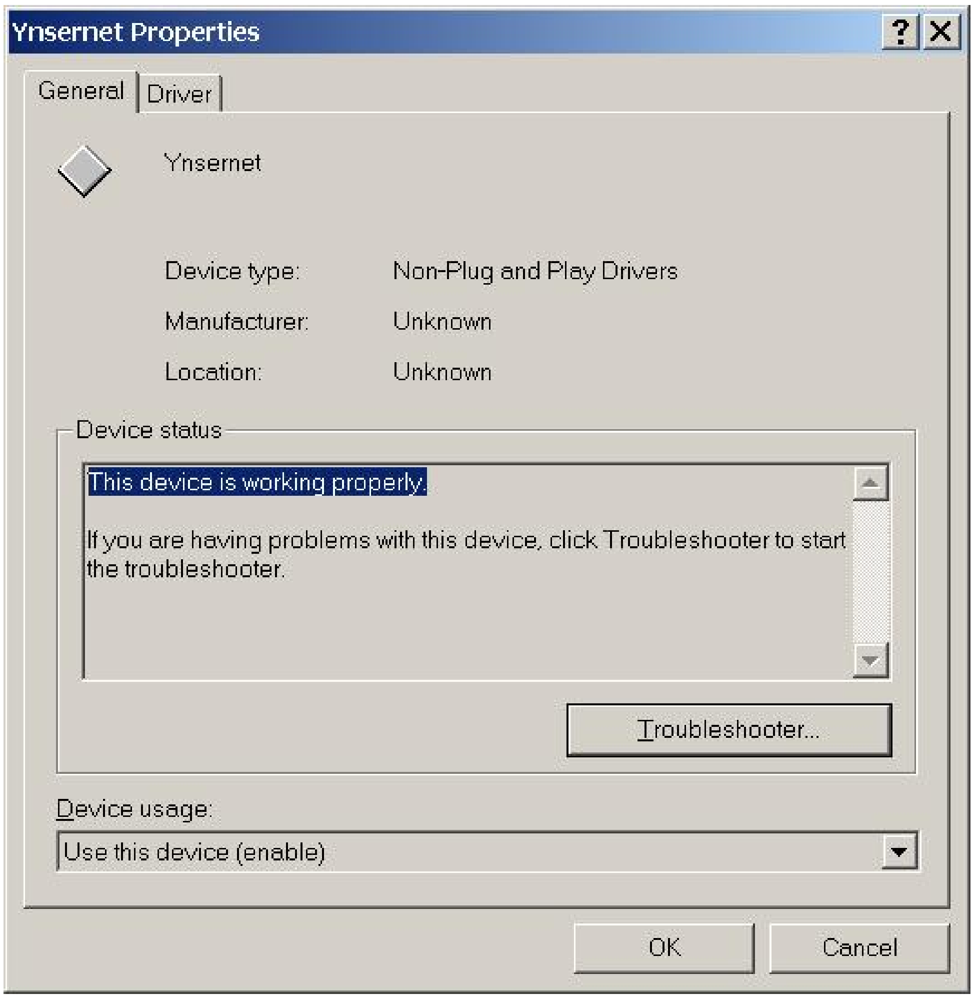


Step 5: Right click the mouse button on the "Ynsernet" item and select the "Properties" menu item.



Step 6: Check if it shows the message "This device is working properly."

If the driver does not work properly, please remove it and then re-install and configure it again.


Ynsernet Properties

General | Driver

Ynsernet

Device type: Non-Plug and Play Drivers

Manufacturer: Unknown

Location: Unknown

Device status

This device is working properly.

If you are having problems with this device, click Troubleshooter to start the troubleshooter.

Troubleshooter...

Device usage:

Use this device (enable)

OK | Cancel

Problem: Client programs open the COM port with success, but fail to access the device.

Check:

Check the device's power supply and wiring (RS-232: RXD, TXD; RS-485: D+, D- ; GND ...).

3.7 FAQ

Q: Which modules are supported by VxComm Driver(PC) ?

A: 7188EA, 7188EX, 7188E1, 7188E2, 7188E3, 7188E3-232, 7188E4, 7188E5, 7188E8, 8430/8830, and 8431/8831.

Q : Does the VxComm Driver(PC) v2.00 work with the VxComm Server (7188E/8000E) v2.6.00 ?

A : No, please upgrade the VxComm Server to version 2.6.14 or lastest version.

The VxComm Server(7188E/8000E) v2.6.00 uses the "06" and "07" command to change the baudrate and the data format and then saves these configurations in the EEPROM.

The newer version adds the "02" and "03" command to change the baudrate and data format without saving. These two commands improve the performance when changing baudrate and data format.

The VxComm Driver(PC) also changed to use the new commands. Thus, users have to upgrade their VxComm Server(7188E/8000E) to the lastest version.

Q: Does the VxComm Driver (PC) support auto-reconnection after fixing a network break?

A: Yes, the VxComm Driver (PC) supports the auto-reconnection mechanism after version 2.00.

The VxComm Utility allows users to set the server-options that include Keep-Alive Time (ms) and Connection-Broken time (ms). Please refer to the section: Adding a 7188E/8000E server and configuring the VxComm driver.

4. Ethernet I/O Applications

4.1 Operation Principle of the Xserver

The typical TCP/IP mechanism is a standard tool but very complicated for a software engineer. It takes long time for a software engineer to develop a programs using TCP/IP protocol.

The command protocol designed for a TCP/IP system can be based on user's applications without any limitations. So every software engineer can design his special protocol without any pre-defined standard. This will cause some of the troubles given as follows:

- Is this protocol reliable?
- Does this protocol fit all requirements?
- How to maintain this protocol by another software engineer?
- Time to market?
- Engineering cost to design & debug this protocol?

The Xserver is designed to solve all problems mentioned above as follows:

- ➢ We design & maintain the reliable, original Xserver for all users.
- ➢ The protocol is designed to fit all requirements of the 7188E series.
- ➢ The protocol is OPEN & expandable to reduce user's design cost.
- ➢ An easy-use interface is designed for user's special applications.
- ➢ Standard design and maintaince for all engineers using this protocol.

The features of the Xserver are given as follows:

- ➢ **The Xserver** is an embedded firmware designed for the 8000E series in the default shipping (8KE10.exe for 8430/8830, Demo36.exe for 8431/8831).
- ➢ Supports Virtual COM applications
- ➢ Supports Ethernet I/O applications
- ➢ Supports I/O expansion bus
- ➢ Supports 8430, 8431, 8830, 8831 and etc.
- ➢ TCP/IP protocol & command protocol is open & expandable.
- ➢ Provides easy-use interface for user's special programs.

With the help of Xserver, a software engineer can design a robust Xserver in one day. We will provide about 50 ~ 100 typical real world applications for user's reference. From these demos, a software engineer can start easily with a cost-friendly time to market. Refer to Sec. 4.3 for more information.

4.2 Command Protocol of Xserver

4.2.1 IP and port configuration

Before developing Ethernet I/O applications for your PC, you must first know the IP address and the Ethernet port number. The 7188E/8000E and all COM ports of the 7188E/8000E use the same IP address, but different Ethernet port number. They are listed below:

Function	IP address	Port number
Modbus TCP	192.168.255.1	502
Virtual 7000 (I/O boards)	192.168.255.1	9999
7188E/8000E configuration	192.168.255.1	10000
COM1 of the 7188E/8000E	192.168.255.1	10001
COM2 of the 7188E	192.168.255.1	10002
COM3 of the 7188E/8000E	192.168.255.1	10003
COM4 of the 7188E/8000E	192.168.255.1	10004
COM5 of the 7188E	192.168.255.1	10005
COM6 of the 7188E	192.168.255.1	10006
COM7 of the 7188E	192.168.255.1	10007
COM8 of the 7188E	192.168.255.1	10008

192.168.255.1 is the default IP address of the 7188E/8000E. You can change the IP address to suit your requirements. Contrary to the IP address, the Ethernet port is fixed. You must use the port number as defined above.

4.2.2 Command set of the Xserver

Cmd	**Explain**	**Instruction Format**	**Example:** Sends to Xserver	**Example:** Receives from Xserver
01	Version	<01>	01	V2.6.14[10/04/2001]
02	Sets baudrate (Doesn't store the setting to EEPROM)	<06,Port(1),Baud>	0619600	OK
03	Sets data format (doesn't store the setting to EEPROM)	<07,Port(1),LineControll(3)	0718N1	OK
04	Gets system	<04,Client(2)>	0414	141(First asking from client

	reset status			14 after system reset) 140(Not first asking from client 14 after system reset)
05	RTS	<05,Port(1),Set(1)>	0511	OK(COM Port RTS on)
06	Sets baudrate (Stores the setting to EEPROM)	<06,Port(1),Baud>	0619600	OK
07	Sets data format (Stores the setting to EEPROM)	<07,Port(1),LineControll(3)	0718N1	OK
08	Sets IP	<08,IP(12)>	08192168255001	OK..Reconnect
10	Server Name	<10>	10	7188E2
11	Diag	<11,String(<=80)>	11Hello	Hello
12	Sets Gateway	<12,GatewayIP(12)>	12192168000001	OK..Reconnect
13	Gets Gateway	<13>	13	192.168.0.1
14	Sets Mask	<14,Mask(12)>	14255255000000	OK..Reconnect
15	Gets Mask	<15>	15	255.255.0.0
16	Gets COM Status	<16,Port(1)>	161	9600,8,N,1
17	Digital Input	<17,Addr_Hex(4)>	1703f8	F8
18	Digital Output	<18,Addr_Hex(4),Data_Hex(2)>	1803f855	OK
19	**Bypass User Defined Command**	**<19,Command>**	**19(User defined)**	**(User defined)**
20	Enable 5 DigitLED	<20,Enable(1)>	201	OK (Enable 5 DititLED show information)
21	Gets Mac	<21>	21	00:80:30:39:9f:e2
22	Gets MiniOS Version	<22>	22	V1.2 (2000/06/17)
23	**Calls VcomUserCmd**	**<23,String>**	**23(User definded)**	**(User defined)**
24	Sets feedback command No.	<24,Enable(1)>	240 241	OK 24OK

Note 1: The number inside () of instruction format is parameter size (byte).

Note 2: Don't insert any space between parameters (except user defined command).

Note 3: All command (except user defined command) responses will add a termanial char CR (0x0d).

Note 4: Refer to vxcomm.htm to get more information about Xserver command protocol and parameter setting in 8000\843x883x\Tcp\Vxcomm\Doc\

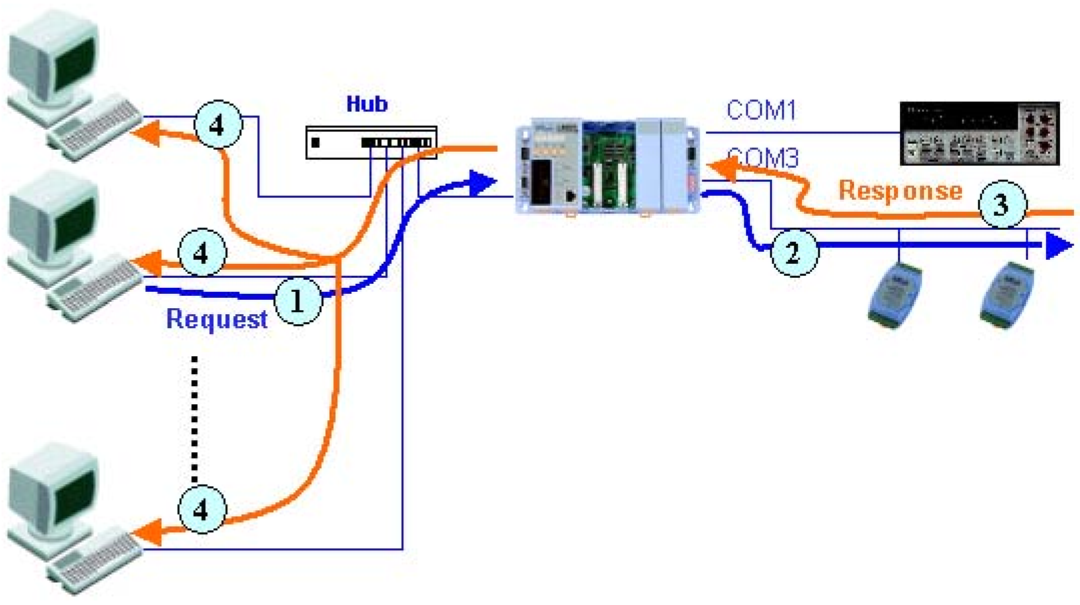
4.2.3 Options of command line

Options of the command line		
Vxcomm.exe [/Option]		
Options	**Explanations**	**Notes**
/1	Recognizes 7188E1	
/2	Recognizes 7188E2	
/3	Recognizes 7188E3	After version 3.0.0
/4	Recognizes 7188E4	After version 3.0.0
/5	Recognizes 7188E5	After version 3.0.0
/8	Recognizes 7188E8	After version 3.0.0
/X	Recognizes 7188EX	After version 3.0.0
/A	Recognizes 7188EA	After version 3.0.0
/M0	Multi-echo mode. Echoes data from the 7188E/8000E's COM ports to every client which is connected to the 7188E/8000E.	
/M1	Single-echo mode. Echoes data from the 7188E/8000E's COM ports to the specific client which requested the service.	After version 2.6.12
/Wxxx	Timeout of building a socket connection. If timeout is up, Vxcomm.exe/Xservder gives up building a socket connection. xxx: timeout Time unit: sec Default: 0 xxx=0: disable option /W	
/STxxx	System timeout between two packets from network to 7188E/8000E. If timeout is up, Vxcomm.exe/Xserver reboots system itself.	

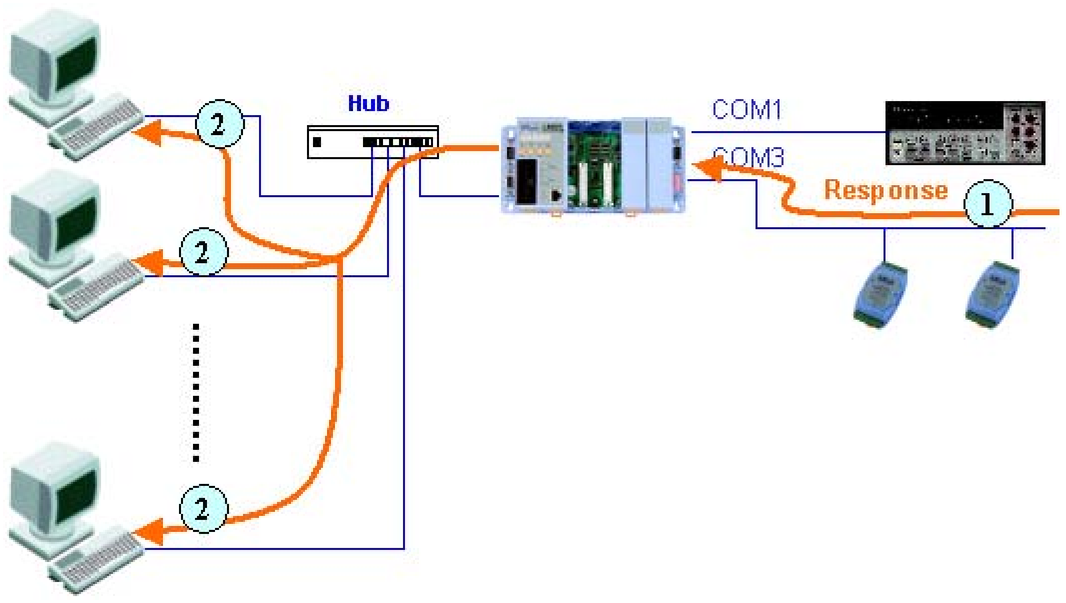
	XXX: timeout Time unit: sec Default: 0 xxx=0: disable option /ST	
/Txxx	Timeout between the 7188E/8000E sending command to COM ports completed and beginning receiving data from the COM ports. If timeout is up, Vxcomm.exe/Xserver gives up receiving data. Time unit: ms Default: 100 ms xxx=0: disable option /T	Acts in M1 (Single-echo mode)

/M0: Multi-echo mode

Condition 1: One client sends a request to Xserver to access devices. The Xserver echoes data from devices to every client which is connected to the 7188E/8000E.

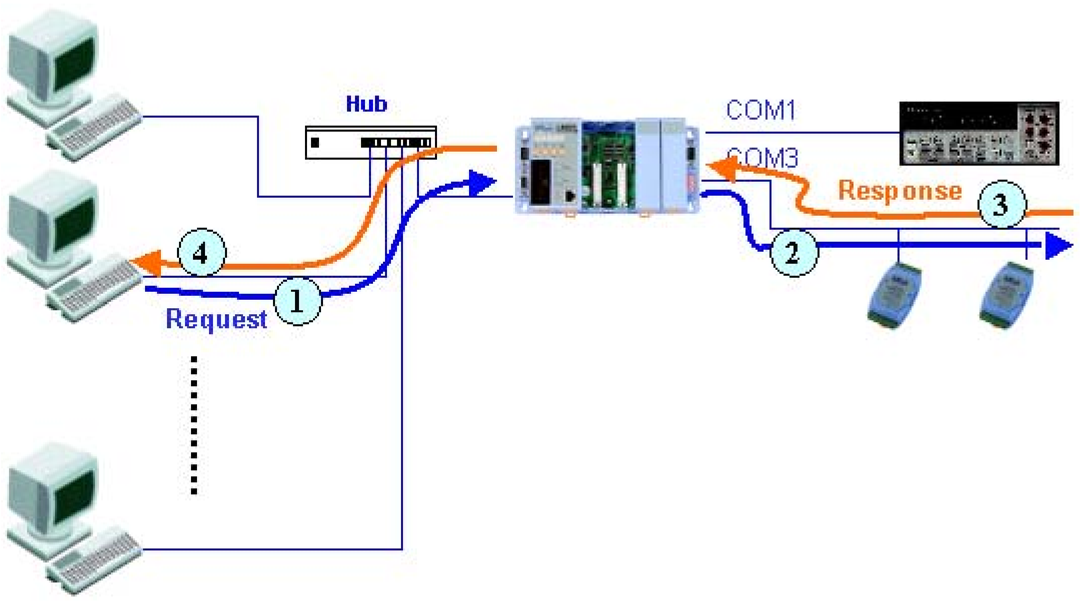


Condition 2: No clients send a request to Xserver to access devices. The Xserver echoes data from devices to every client which are connected to the 7188E/8000E.

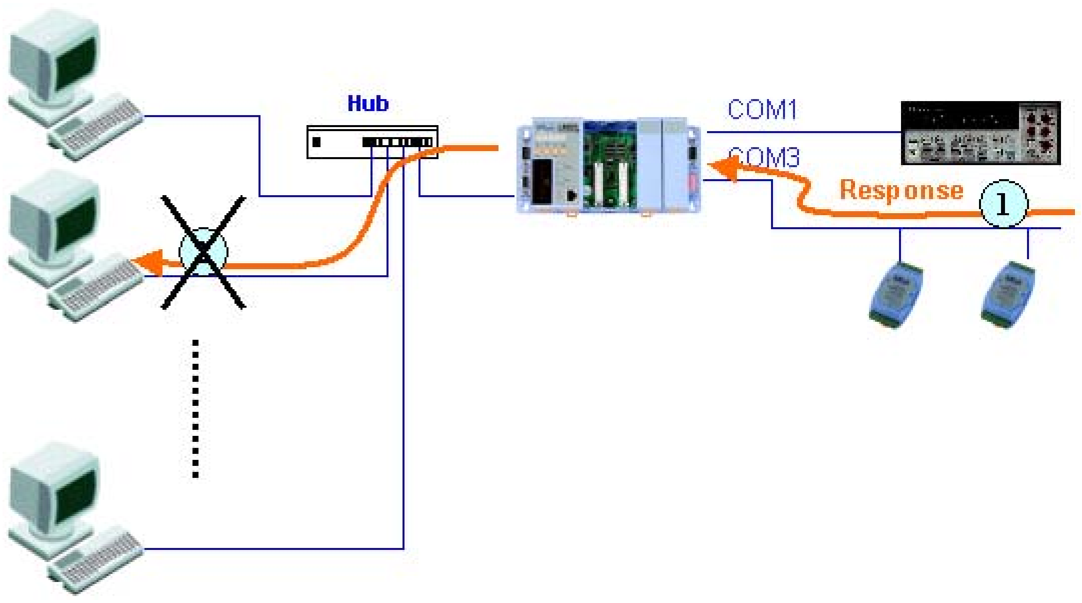


/M1: Single-echo mode

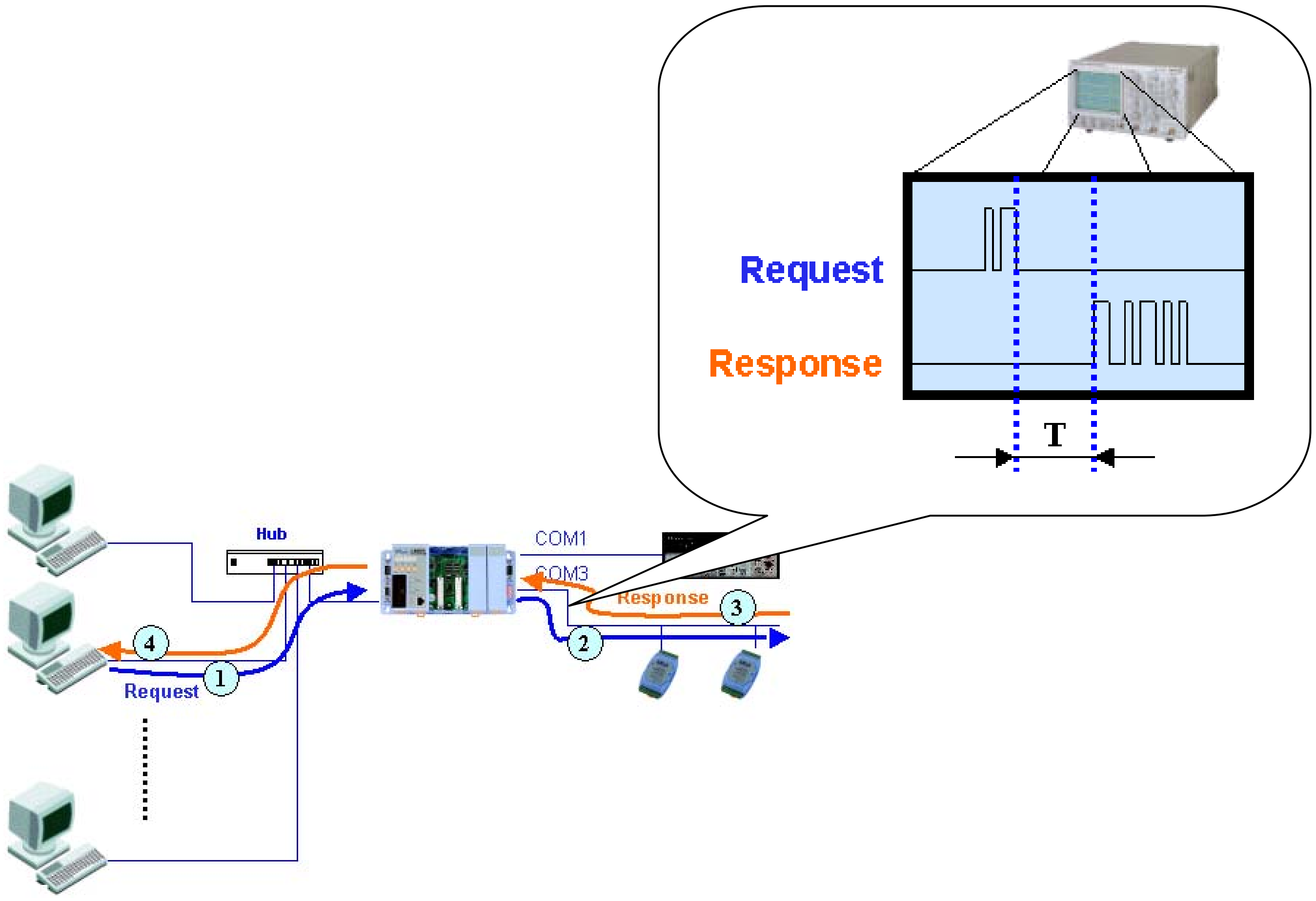
Condition 1: One client sends a request to Xserver to access devices. The Xserver echoes data from devices to the client which requested the service.



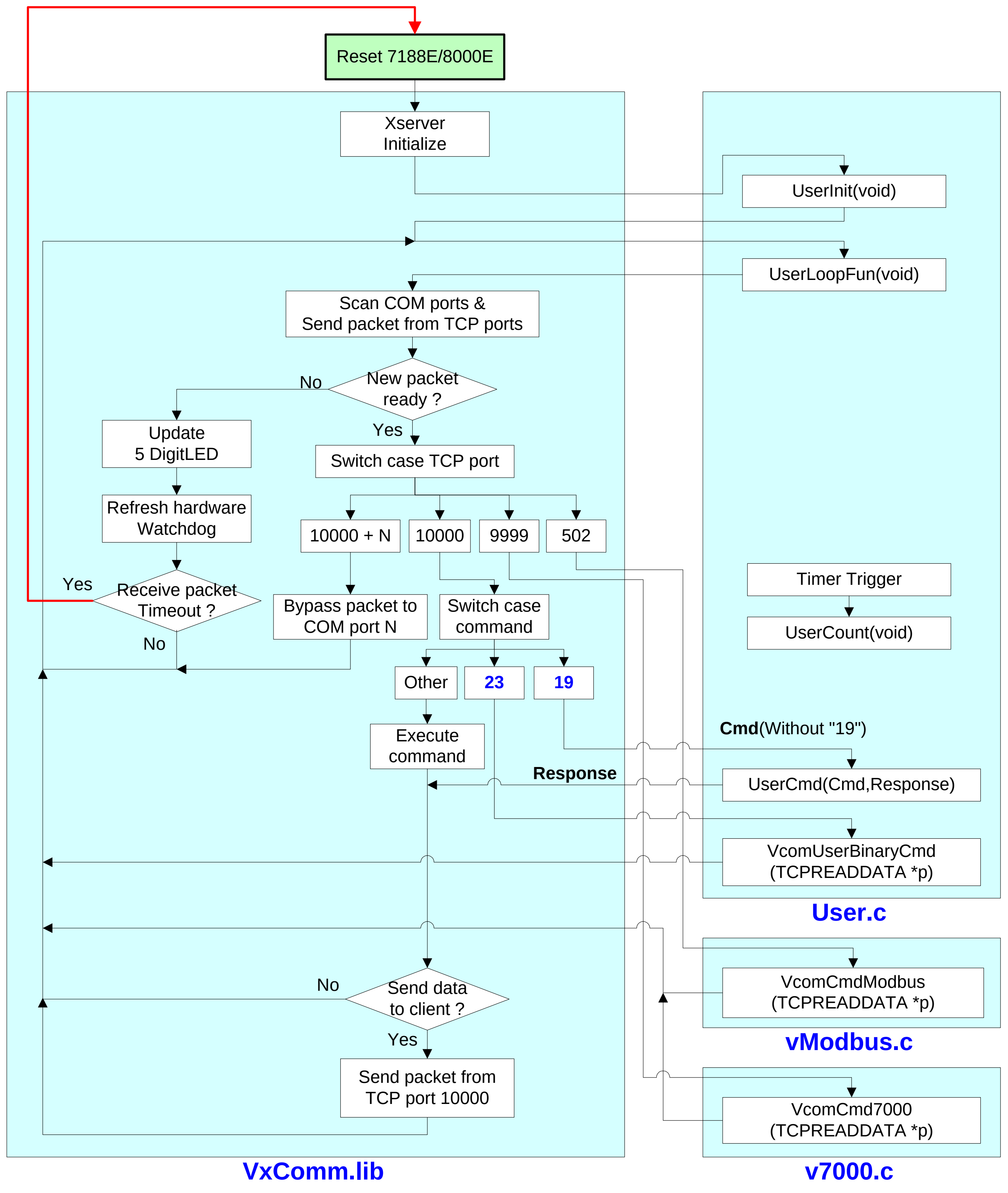
Condition 2: No clients send any request to Xserver to access devices. The Xserver doesn't echo data from devices to any client.



/Txxx:



4.2.4 Flow chart of Xserver



It is very difficult to develop an enbedded controller program with Ethernet/Internet communication. But by using the Xserver, users can do that quickly and easily. Users need only modify 7 functions in User.c, vModbus and v7000.c. Users build their own code in the 7 functions to make the Xserver work as they desire. The features of the 7 functions are listed below:

- **UserInit (void):**
  Xserver executes this function once as soon as the 7188E/8000E is turned on. Therefore all variables of initial values or initial status must be set in this function.

- **UserLoopFun (void):**
  Xserver executes this function every scan loop. One Xserver scan loop completes in a short time, so real time work should be executed by this function. **See Demo12.**

- **UserCount (void):**
  This function will be triggered when the time interval, set in AddUserTimerFunction, is up. For best result, call AddUserTimerFunction in UserInit to let the Xserver call the UserCount period. Longest time interval is 65.535 seconds (2^16–1 ms). **See Demo9.**

- **UserCmd (Cmd, Response):**
  Xserver executes this function when client program sends the command "**19**" to port 10000 of the Xserver. This command is defined in UserCmd function by users. When the Xserver receives packets from **port 10000**, the Xserver checks the data. If the data begins with **"19"**, the Xserver trims "19" and passes the other data (not including "19") to be the first parameter **Cmd** of function UserCmd.

  User can define his own command protocol in UserCmd. For example, define <19,Read/Write(1),address(4),[value(2)]> to replace command 17 and 18, then one can send "19R03f8" to read values form address 03f8; sends "19W03f85a" to write 5a to address 03f8.
  Users can decide whether or not any other characters are needed between parameters. Any command protocol format will be accepted, because it is user defined.
  At the end of UserCmd, copy the results to the second parameter **Response**

Then the XServer will send the string to the Client program by port 10000. **See Demo4.**

- **VcomUserBinaryCmd (TCPREADDATA *p):**
  Xserver executes this function when client program sends command "**23**" to port 1000 of the Xserver. This function is similar to UserCmd. When the client program sends command "23", VcomUserCmd will receive TCPREADDATA type information. The TCPREADDATA is declared as below:
  ```
  Type define t_TcpReadData{
  Int Comport;
  int Socket;
  int Length;
  char* ReadUartChar;
  } TCPREADDATA;
  ```
  p->ReadUartChar: the buffer where command data is stored(include "23")
  p->Length: the command data length (include "23")
  p-Socket: the Xserver assigns a socket number to index when client sends command "23" to the 7188E. So, the socket number can used to return messages to a specific client. To send message to specific client, call VcomSendSocket (int skt, char * data, int cnt). The first parameter should be the socket number. **See Demo23.**

- **VcomCmdModbus (TCPREADDATA *p):**
  Xserver executes this function when client program sends commands to port 502 of the Xserver. This function is used to implement Modbus TCP protocol to access devices. **See Demo46.**

- **VcomCmd7000 (TCPREADDATA *p):**
  Xserver executes this function when client program sends commands to port 999 of the Xserver. This function is used to implement 7000 series-compatiable commands to access expansion boards.
  See 8000\843x883x\TCP\Xserver\v7000\*.*

Note: Please rafer to 8000\843x883x\Document\ModbusTCP.pdf for more information about how to use ModbusTCP to communicate our 8000 series hardware.

4.3 Demo program list of Xserver

After developing the Xserver. Users must download the program into the 7188E/8000E and execute one client program to test if all functions run properly.

Demo	Function	Explanation	Client
Demo4 (Original X-Server)	Echoes command string		Client1
Demo5	Echoes special string to clients	Xserver will echo "7188_Series." or "8000_Series." to clients.	Client1
Demo7	Uses printCom1 to debug programs	You can use "Print" or "printCom1" to send a Debug string to PC monitor by 7188E/8000E's COM1. If you want to use "Print", you must use "DisableCom" and "RestoreCom" to disable "printCom1".	Client1
Demo8	Uses Print to debug programs	You can use "Print" or "printCom1" to send a Debug string to PC monitor by 7188E/8000E's COM1. If you want to use "Print", you must use "DisableCom" and "RestoreCom" to disable "printCom1".	Client1
Demo9	Timer trigger demo	UserCount will be executed every second. Count value will be icreased in UserCount. PC can read count value to know how many seconds after count value be cleared.	Client1
Demo10	Refreshes Watchdog demo	If user's function cost more than 1.6 seconds. User must insert RefreshWDT function to refresh WDT avoid the OS restarting itself in 1.6 second.	Client1

Demo12	Scan-time evaluation	UserLoopFun will increase count value every scan loop. PC can read how many scan loops there are after clear count value. So user can use this demo to test Xserver performance.	Client2
Demo14	Controls 7-SEG LED	Show5DigitLed, Show5DigitLedWithDot can show 5 digits to 7-SEG LEDs The two functions can show '0' ~ '9' 'a' ~ 'f' 'A' ~ 'F' ' ', '-', '.'	Client4
Demo18	Reads I-7000 series module's ID	This demo shows how to communication with the 7000 series which are connected to COM2 of 7188E or COM3 of 8000E.	Client4
Demo19	Reads 64 bits unique hardware serial number	Unique serial number is used to protect user's software. Using 7188xw.exe to enter 7188E, MiniOS7 will show a number. User can check the number at first, then decide to execute Xserver from that point on.	Client4
Demo20	Reads/Writes/Clears NVRAM	NVRAM's characteristic is short response time, limitless erasure and battery backup for 10 years.	Client4
Demo21	Controls hardware in UserCount	Actions concerning hardware control in UserCount is prohibited. If users want to control hardware in UserCount, they must use flag variable to pass the command to UserLoopFun. Function of hardware can be executed correctly in UserInit, UserLoopFun, UserCmd. This demo will increase numbers every second and show the value in the LEDs in UserCount function.	Client4
**Demo23	Echoes all data	Uses VcomSendSocket to echo all data	Client4

	(including "23") to specific clients.	(including "23") to specific clients.	
Demo24	Uses countdown timer	There are 8 countdown timer (channel 0 to channel 7). This demo uses channel 0. The countdown timer initial value is 1000 ms. When the countdown timer value become 0, the value of the LED will increase.	None
Demo26	Links to MMI by Modbus Serial portocol		
Demo36 (8E only)	Reads Units and modules's ID	This demo is the default shipping program for 8431/8831. Users can use this demo to find out which modules are plugged in 8000E.	Client4
Demo37 (8E only)	Reads Sytsemkey status	This demo uses command 23 to switch on/off auto return system status. PC sends command "23 on" at beginning. When user press system key, Xserver will auto reply which system key pressed.	Client4
Demo38 (8E only)	Shows LED of 8000 modules	Only LED of DO and DIO parallel modules can be set by function ShowLED8 & ShowLED16 And the module must have total 16 channels.	Client4
Demo39 (8E only)	Sets LED of 8000 units	Controls L1, L2 and L3 of 8000 units.	Client4
Demo40 (8E only)	D/I from 8000 modules (16 DI channels)	Uses DI_16 to D/I from parallel DI modules (16 channels) on 8000 slots	Client4
Demo41 (8E only)	D/I from 8000 modules (8 DI	Uses DI_8 to D/I from parallel DI or DIO modules (8 DI channels) on 8000 slots	Client4

	channels)		
Demo42 (8E only)	D/O to 8000 modules (16 DO channels)	Uses DO_16 to D/O to parallel DO modules (16 DO channels) on 8000 slots	Client4
Demo43 (8E only)	D/O to 8000 modules (8 DO channels)	Uses DO_8 to D/O to parallel DO modules (16 DO channels) on 8000 slots	Client4
Demo44 (8E only)	D/O to 8000 modules (8 DO channels)	Uses DIO_DO_8 to D/O to parallel DIO modules (8 DO channels) on 8000 slots	Client4
Demo45	Uses COM3 to control 87k modules		Client4
Demo46 (8E only)	Uses Modbus TCP protocol to access 8000 modules		
..... More demo programs			

We will provide **50 ~100 demo programs** for the Xserver in the future. Please refer to 8000\843x883x\TCP\Xserver\Xserver.htm and **Function.htm** for more information of demo programs.

4.4 Client program list for Xserver

- **8430/8830**

Client programs	Explanation	Special Demo
8430	Full features demo (include **Virtual 7000** function)	All simple demos
..... **More client programs**		

Client programs are located at 8000\843x883x\TCP\Xserver\8430

- **8431/8831**

Client programs	Explanation	Special Demo
8431	Full features demo (include **Virtual 7000** function)	All simple demos
..... **More client programs**		

Client programs are located at 8000\843x883x\TCP\Xserver\8431

- **Common clients**

Client programs	Explanation	Special Demo
Client1	Used to test all simple Xserver demos	All demos
Client2	Sends "19 0", then delay n ms and send command "19 1".	Demo12
Client4	Similar to Client1, just adds sending string with "CR" function.	All demos
Client5	Sends period step function signals to I/O port of the 7188E.	Demo17
***CheckValue	CheckValue main functions: 1. Catches below errors: a. Winsock error	Demo22, Demo25 or demos which return value (total 7

	b. Receive data from Xserver timeout 2. Checks received data out of range 3. Automatic reconnect ability.	characters)
***CheckString	CheckString main functions: 1. Catches below errors: a. Winsock error b. Receive data from Xserver timeout 2. Checks received data a. Full comparison b. Part comparison 3. Automatic reconnect ability.	All demos
***GetString	GetString main functions: 1. Catches below errors: a. Winsock error b. Receive data from Xserver timeout 2. Automatic reconnect ability. 3. Large text box can show 1600 bytes on one page.	All demos
..... More client programs		

Common client programs are located at 8000\843x883x\TCP\Xserver\Client

Note:

- You can install all 8000E special clients (8430, 8431, etc.) by executing 8000\843x883x\TCP\Xserver\Setup\Setup.exe.
- You can install all common clients (Client1, Client2, Client4, Client5, CheckValue, CheckString, GetString, etc.) by executing 8000\843x883x\TCP\Xserver\Client\Setup\Setup.exe.
- Please refer to 8000\843x883x\TCP\Xserver\Xserver.htm and **Function.htm** for more information of demo programs.

4.5 Programming of Xserver

To develop the Xserver, you must have the 9 files listed below:

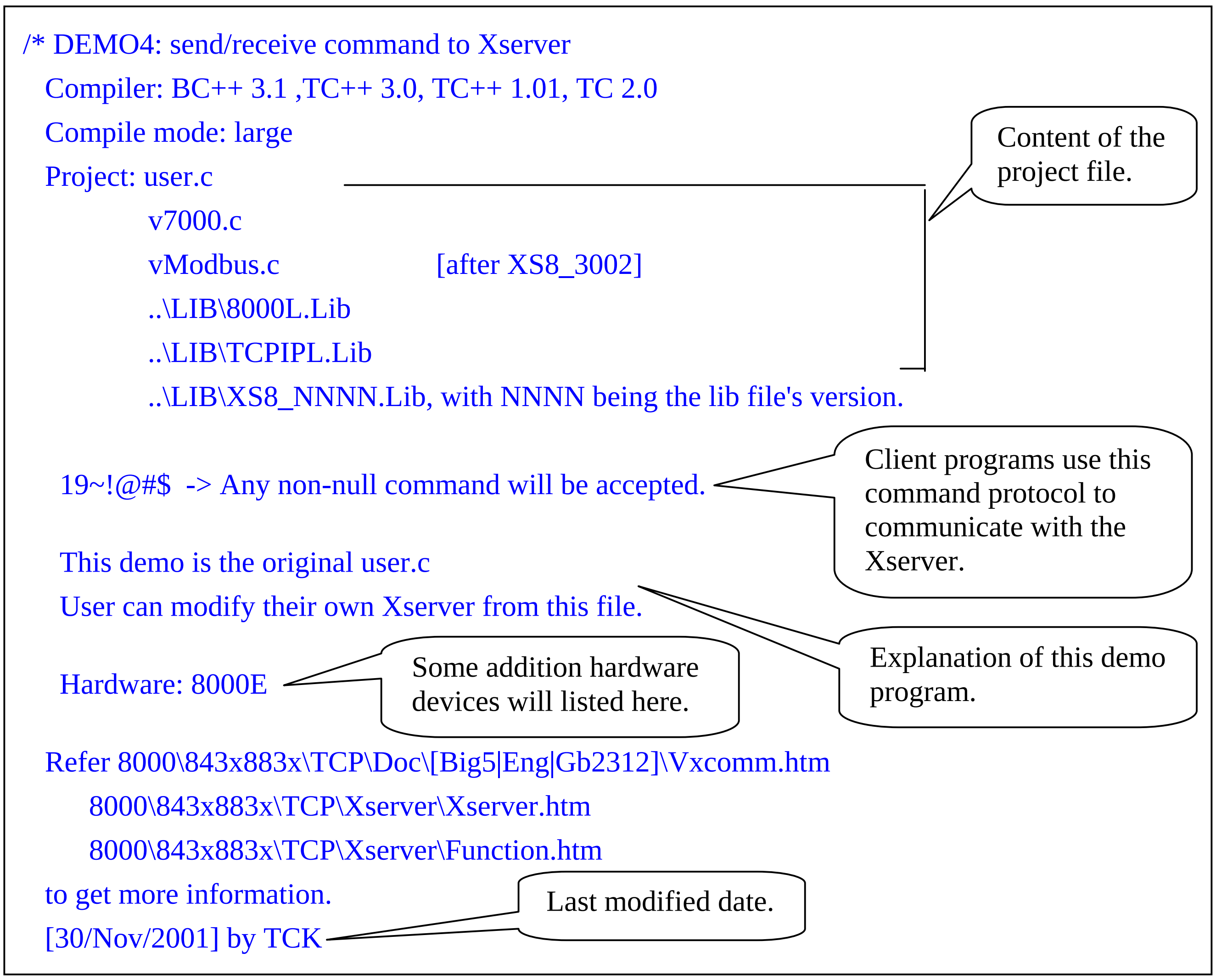
Item	**Files**	**Location**
Head file	8000.h, TCPIP.h, Vxcomm.h	8000\843x883x\TCP\Xserver\Demo\BC\Lib
Library	8000L.Lib, TCPIPL.Lib, XS8_NNNN.Lib	
User's file	User.c, vModbus.c, V7000.c	8000\843x883x\TCP\Xserver\Demo\BC\Demo4

Note: The NNNN of XS8_NNNN.Lib is the lib file's version.

4.5.1 Original Xserver

All **user.c** files are devided into two parts. One is the explanation head, the other one is the program body.

- **Explanation head:**



```
/* DEMO4: send/receive command to Xserver
   Compiler: BC++ 3.1 ,TC++ 3.0, TC++ 1.01, TC 2.0
   Compile mode: large
   Project: user.c
            v7000.c
            vModbus.c              [after XS8_3002]
            ..\LIB\8000L.Lib
            ..\LIB\TCPIPL.Lib
            ..\LIB\XS8_NNNN.Lib, with NNNN being the lib file's version.

     19~!@#$  -> Any non-null command will be accepted.

     This demo is the original user.c
     User can modify their own Xserver from this file.

     Hardware: 8000E

   Refer 8000\843x883x\TCP\Doc\[Big5|Eng|Gb2312]\Vxcomm.htm
       8000\843x883x\TCP\Xserver\Xserver.htm
       8000\843x883x\TCP\Xserver\Function.htm
   to get more information.
   [30/Nov/2001] by TCK
```

Content of the project file.
Client programs use this command protocol to communicate with the Xserver.
Explanation of this demo program.
Some addition hardware devices will listed here.
Last modified date.

● Program body:

```
#include<string.h>
#include "..\lib\8000.h"
#include "..\lib\vxcomm.h"
void UserCount(void)
{
/*
        // user's timer trigger function
        //
        // In this function, users cannot use any function that will use the hardware signal "clock",
        // Such as:
        //  1. ClockHigh(),ClockLow(), ClockHighLow(),
        //  2. Any EEPROM functions.
        //  3. Any 5DigitLed functions.
        //  4. Any NVRAM function.
        //  5. Any RTC function.(GetTime(),SetTime(),GetDate(),SetDate())
        //
        // refer to demo9 for example code
*/
}

void UserInit(void)
{
/*
        // user's initial function
        // timer initialized for UserCount()
        // I/O or variables initialized for UserLoopFun()
        // I/O or variables initialized for User's functions in this file
        // refer to demo9 & demo11 for example code
*/
}

void UserLoopFun(void)
{
/*
        // VxComm.exe will call this function every scan time
        // refer to demo11 for scan-time evaluation
*/
}
```

```
int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
/*
        // user's command interpreter
        // refer to all demo
*/
        if (Cmd[0])                     /* Not Null command */
        {
                strcpy(Response,Cmd); /* echo user's command back */
                return 1;                       /* return OK */
        }
        return 0;          /* return ERROR */
}

int VcomUserBinaryCmd(TCPREADDATA *p)
{
  /* VXCOMM.EXE 2.6.12(09/04/2001) or later will support this function.

        TCP PORT 10000, command 23 will call this function.
        user can get the following message:
        p->ReadUartChar : the buffer store the command data(include "23")
        p->Length : the command data length(include the two byte "23")
        p->Socket : the socket number that receives the command, that is, when the user function wants
                    to return a message to the client, just use the socket to send data.
                    use:  VcomSendSocket(p->Socket,pdata,datalength);
  */
        return 1;  /* any value will be accept */
}
```

Note:

- ➢ We will enhance the Xserver continually. Please refer to 8000\843x883x\Document\Readme.htm and
- ➢ 8000\843x883x\TCP\Vxcomm\Doc\[Big5|Eng|Gb2312]**Vxcomm.htm** for new version information.

4.5.2 Default shipping Xserver for 8431/8831

The default shipping Xsever for 8431/8831 is **Demo36.exe**. With the help of the Demo36.exe, you can read the 8000E Unit’s Name and modules’s ID plugged in the Unit.

Plese refer to sec. 4.6.3 for how to use client program to test the default shipping Xserver.

- **Part of the explanation head:**

```
//  190 ---> Read the module's ID plugged in slot 0
//  191 ---> Read the module's ID plugged in slot 1
//   :
//  197 ---> Read the module's ID plugged in slot 7
//  19N ---> Read the Unit's ID, with N > 7

//  This demo is the default shipping program for 8431/8831.
//  Users can use this demo to find out which modules are
//  plugged in 8000E.

//  Hardware: 8000E
```

- **Part of the program body:**

```
int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
        int iTotalSlotNum;
        int iTotalComNum;
        int iSlot;

        if(Cmd[0]) // Not null command
        {
                sscanf(Cmd,"%d",&iSlot);
```

```
          switch(iSlot)
          {
               case 0:
               case 1:
               case 2:
               case 3:
               case 4:
               case 5:
               case 6:
               case 7:
                    if(NameOfModule[iSlot] != 0xff)
                    {
                         if( (ModuleType[iSlot] & 0x80) == 0)
                              sprintf(Response,"Slot(%d) Find
870%02d\n\r",iSlot,NameOfModule[iSlot]);
                         else
                              sprintf(Response,"Slot(%d) Find
80%02d\n\r",iSlot,NameOfModule[iSlot]);
                    }
                    else
                         sprintf(Response,"Slot(%d) No Find\n\r",iSlot);
                    return 1;

               default:
                    iTotalSlotNum=GetNumberOfSlot();
                    iTotalComNum=GetComportNumber();
                    if(iTotalComNum==3)
                         sprintf(Response,"Main Unit => I-8%d30",iTotalSlotNum);
                    else
                         sprintf(Response,"Main Unit => I-8%d31",iTotalSlotNum);
                    return 1;
          } // of switch
     } // of if
     return 0;
}
```

4.5.3 Setting of compiler (BC++ 3.1)

To develop programs for 7188/7188X/7188E/8000 series, you can use the compilers below:

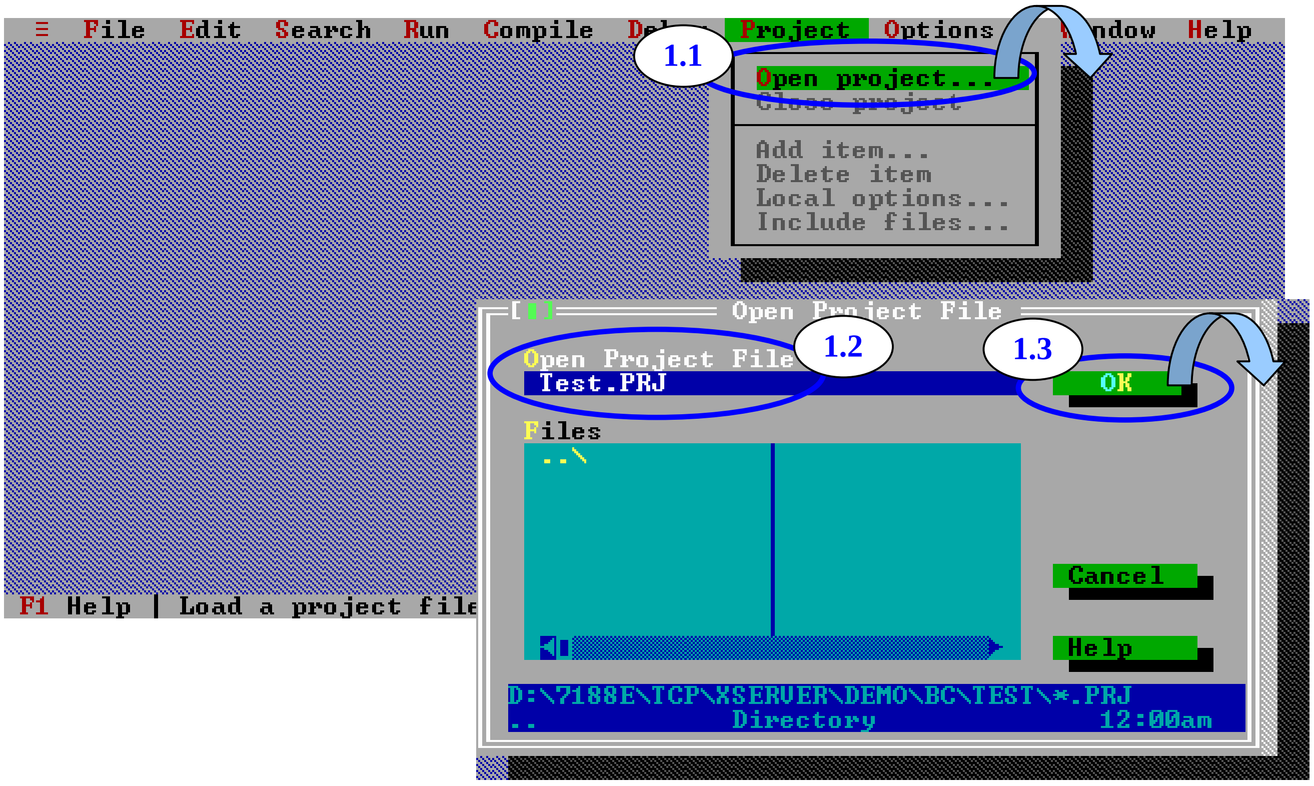
1. BC++ 3.1~5.02
2. MSC
3. MSVC (before version 1.52)
4. TC 2.01
5. TC++ 1.01

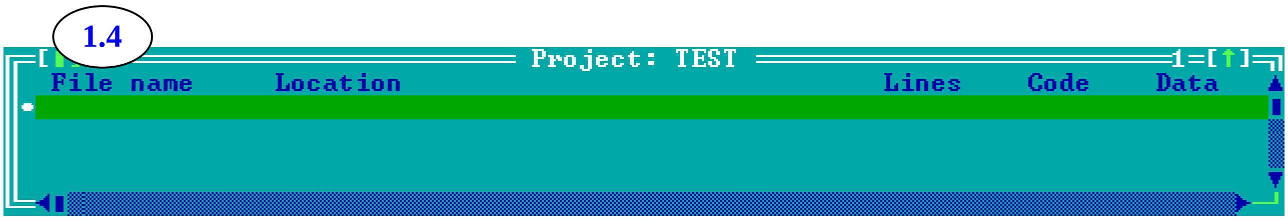
From Borland's web site, you can download the free TC 2.01 and TC++ 1.01 compilers.

Web site: http://community.borland.com/museum/

How to use BC++ 3.1's IDE to compile projects? Please follow the steps below:

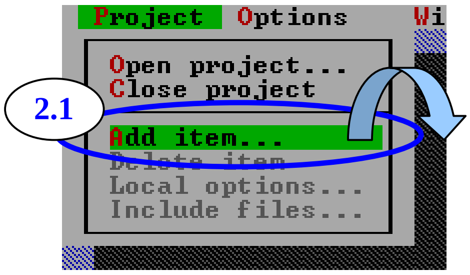
Step 1: Create a new project.



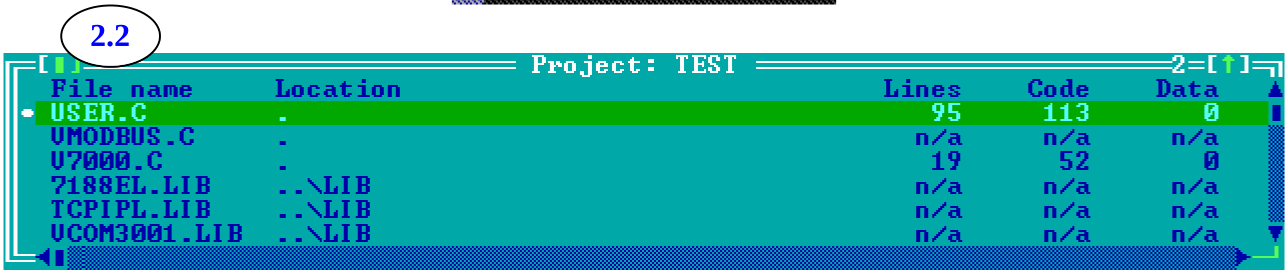


```
Project: TEST
File name        Location                          Lines     Code     Data
```

Step 2: Add all necessary files into the project.

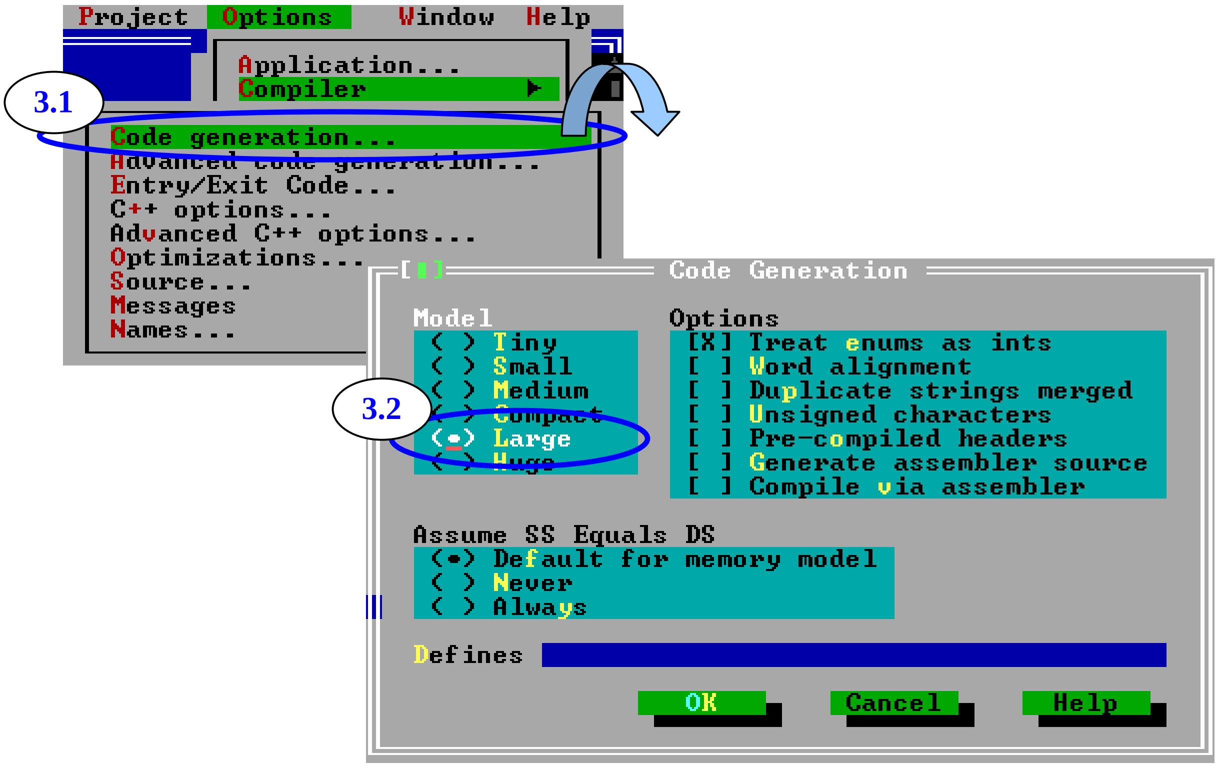


```
Open project...
Close project
Add item...
Delete item
Local options...
Include files...
```



```
Project: TEST
File name        Location                          Lines     Code     Data
USER.C           .                                    95      113        0
VMODBUS.C        .                                   n/a      n/a      n/a
V7000.C          .                                    19       52        0
7188EL.LIB       ..\LIB                              n/a      n/a      n/a
TCPIPL.LIB       ..\LIB                              n/a      n/a      n/a
VCOM3001.LIB     ..\LIB                              n/a      n/a      n/a
```

Step 3: Set Code generation options.



```
Application...
Compiler
  Code generation...
  Advanced code generation...
  Entry/Exit Code...
  C++ options...
  Advanced C++ options...
  Optimizations...
  Source...
  Messages...
  Names...
```

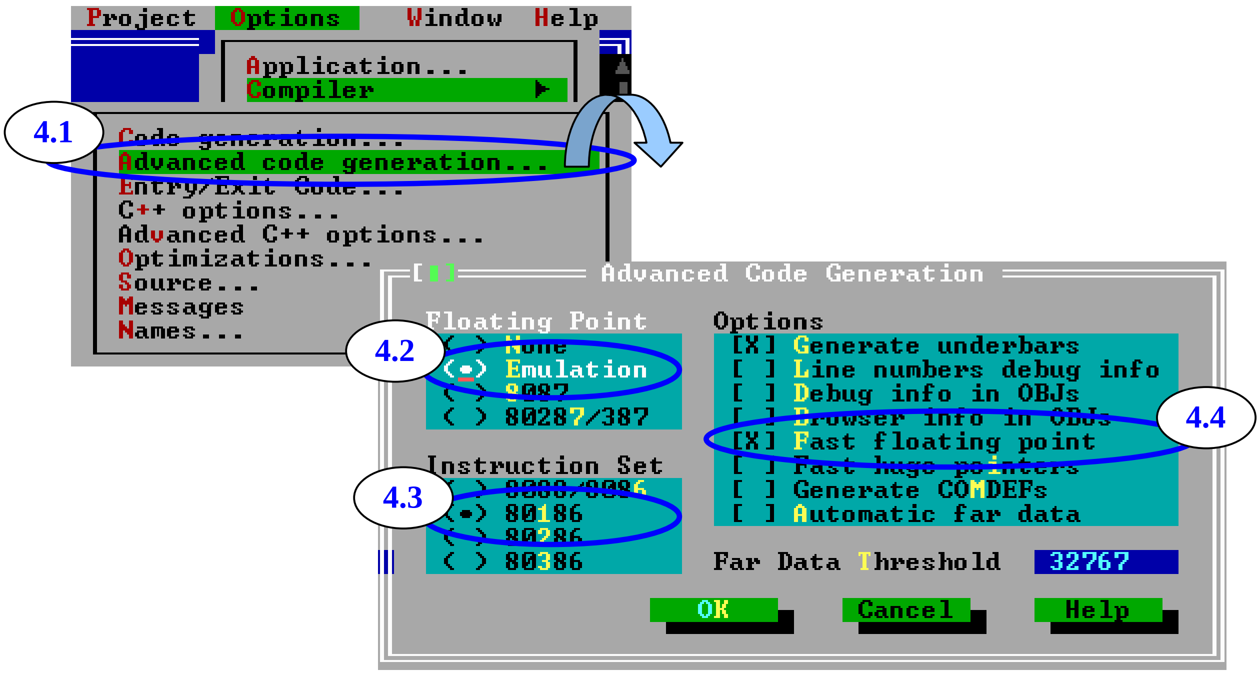
```
Code Generation

Model                Options
( ) Tiny             [X] Treat enums as ints
( ) Small            [ ] Word alignment
( ) Medium           [ ] Duplicate strings merged
( ) Compact          [ ] Unsigned characters
(•) Large            [ ] Pre-compiled headers
( ) Huge             [ ] Generate assembler source
                     [ ] Compile via assembler

Assume SS Equals DS
(•) Default for memory model
( ) Never
( ) Always

Defines
```

Step 4: Set Advanced code generation options.



Advanced Code Generation

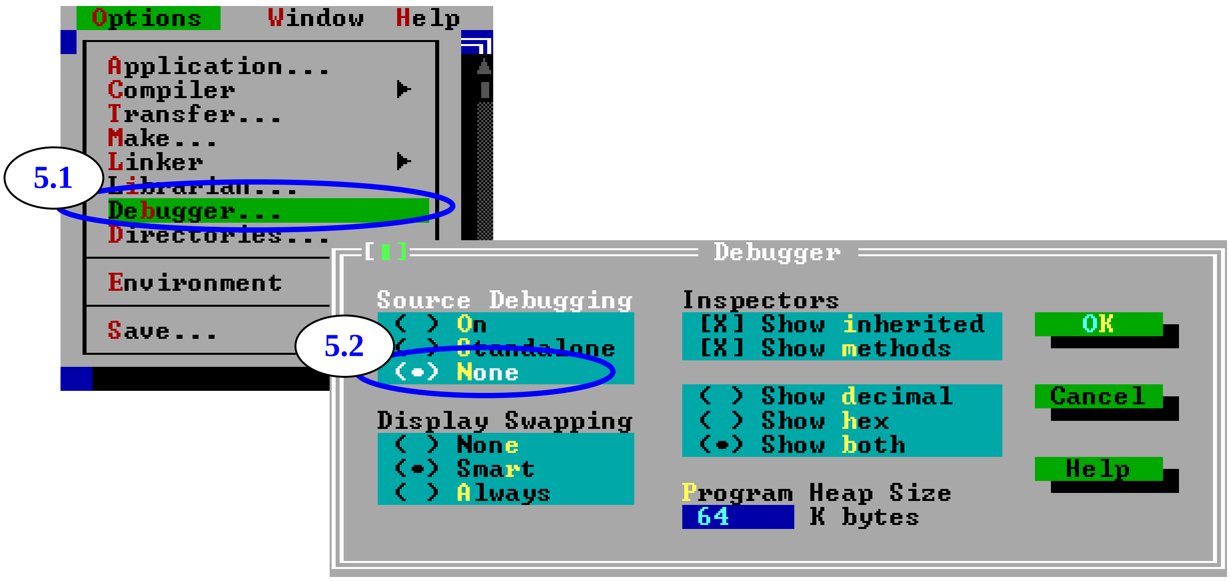
Floating Point
( ) None
(•) Emulation
( ) 8087
( ) 80287/387

Instruction Set
( ) 8088/8086
(•) 80186
( ) 80286
( ) 80386

Options
[X] Generate underbars
[ ] Line numbers debug info
[ ] Debug info in OBJs
[ ] Browser info in OBJs
[X] Fast floating point
[ ] Fast huge pointers
[ ] Generate COMDEFs
[ ] Automatic far data

Far Data Threshold 32767

Step 5: Set Debugger options.



Debugger

Source Debugging
( ) On
( ) Standalone
(•) None

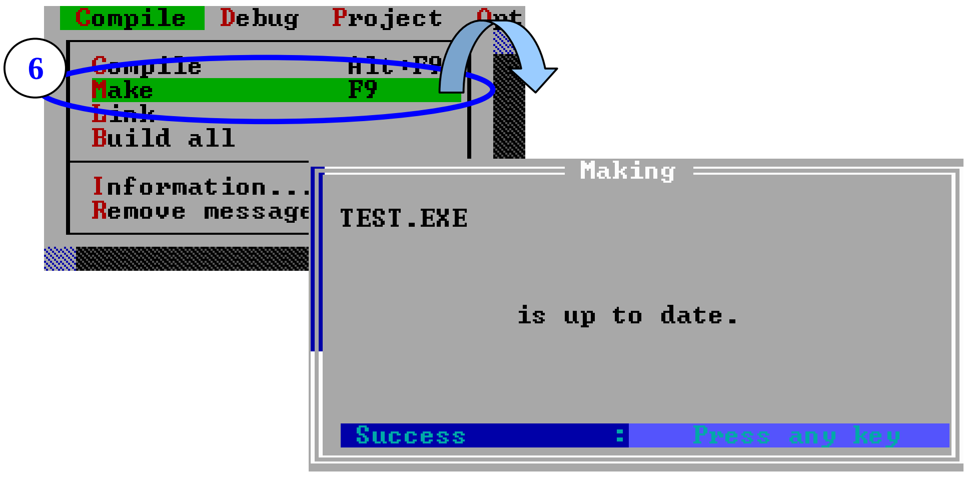
Display Swapping
( ) None
(•) Smart
( ) Always

Inspectors
[X] Show inherited
[X] Show methods

( ) Show decimal
( ) Show hex
(•) Show both

Program Heap Size
64 K bytes

Step 6: Make the project.



```
TEST.EXE

        is up to date.

Success     :     Press any key
```

Note:

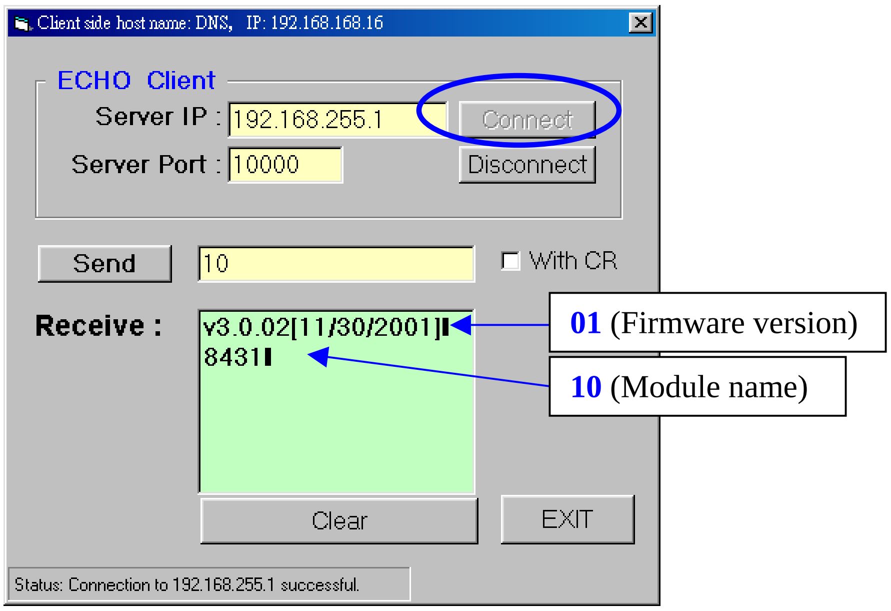
Please refer to 8000\843x883x\Document\TCPLib.pdf for more information about settings of other compilers.

4.6 Using client programs

To test functions of Xserver, you must run a client program. We support many Quick Start documents for specific modules, like **8430_Quick_Start.pdf**, **8431_Quick_Start.pdf**. Please refer to documents in 8000\843x883x\Document.

4.6.1 Using Client4.exe to link 8431

Step 1: Run 8000\843x883x\TCP\Xserver\Client\Vb5\Client4\Client4.exe in host-PC. Press "**Connect**" button to connect to 8431.
Send command "**01**", "**10**".



Client side host name: DNS, IP: 192.168.168.16

ECHO Client

Server IP : 192.168.255.1

Server Port : 10000

With CR: unchecked

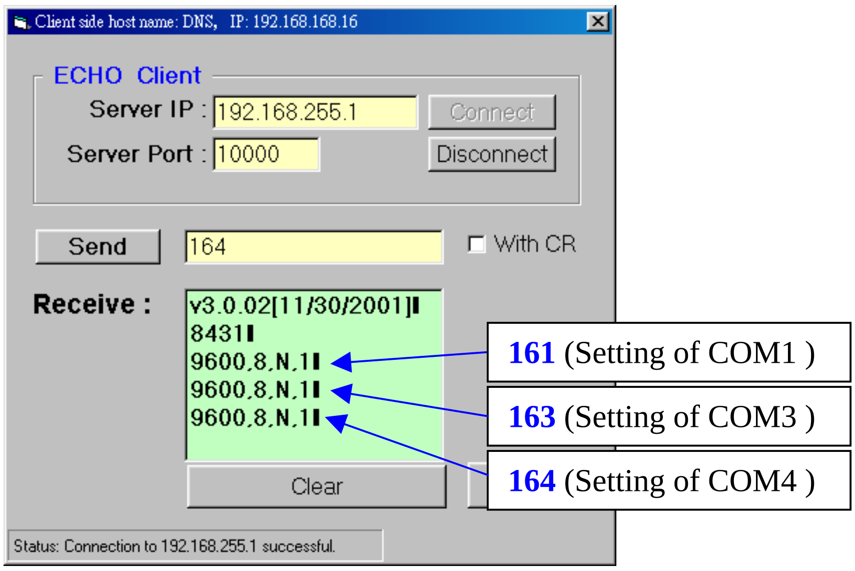
Send: 10

Receive :

```
v3.0.02[11/30/2001]
8431
```

Status: Connection to 192.168.255.1 successful.

Step 2: Send "**161**", "**162**" and "**163**" to readout COM port setting.



Client side host name: DNS, IP: 192.168.168.16

ECHO Client

Server IP : 192.168.255.1

Server Port : 10000

With CR: unchecked

Send: 164

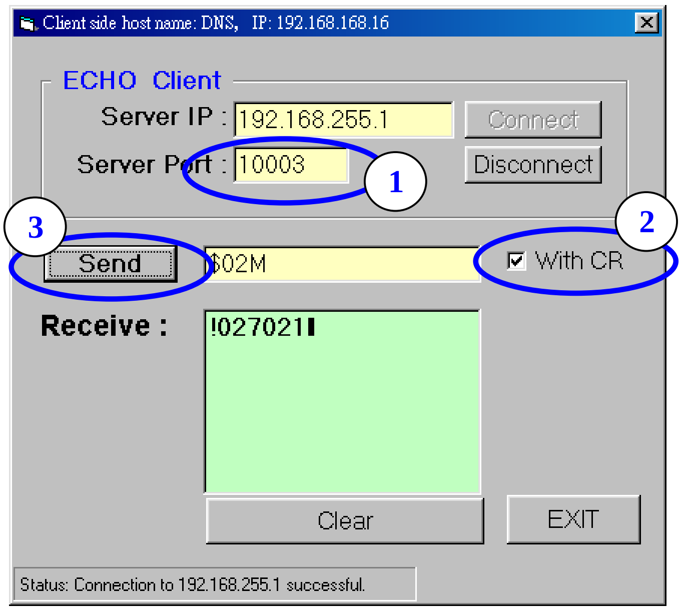
Receive :

```
v3.0.02[11/30/2001]
8431
9600,8,N,1
9600,8,N,1
9600,8,N,1
```

Status: Connection to 192.168.255.1 successful.

Step 3: Disconnect and then reconnect at port 10003.

Step 4: Select "**With CR**" and then send "**$02M**" to read 7000 module's ID which is connected to 8431's COM3.



Client side host name: DNS, IP: 192.168.168.16

ECHO Client

Server IP : 192.168.255.1

Server Port : 10003

Send: $02M

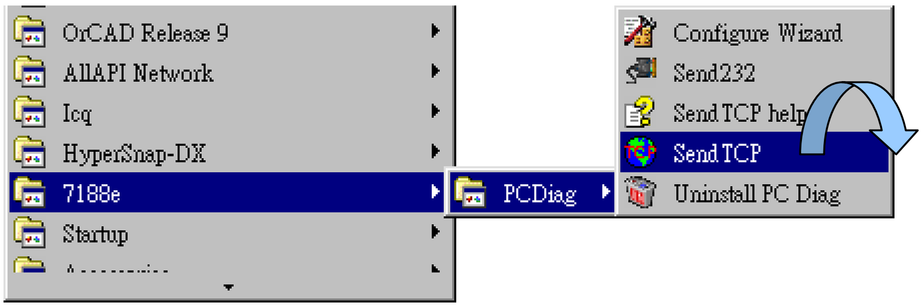
[✓ With CR]

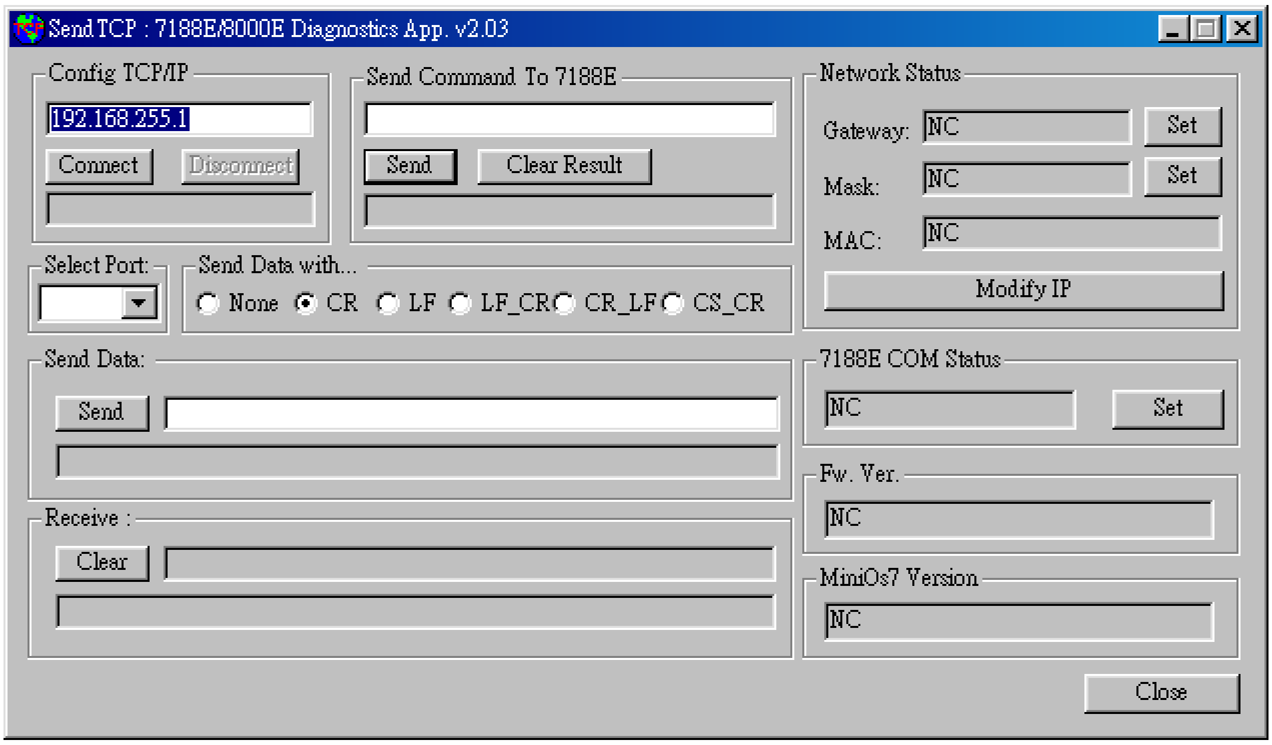
Receive : !027021

Status: Connection to 192.168.255.1 successful.

4.6.2 Using SendTCP to link 8431

Step 1: Run SendTCP in host-PC.





IP: 192.168.255.1

Gateway: NC

Mask: NC

MAC: NC

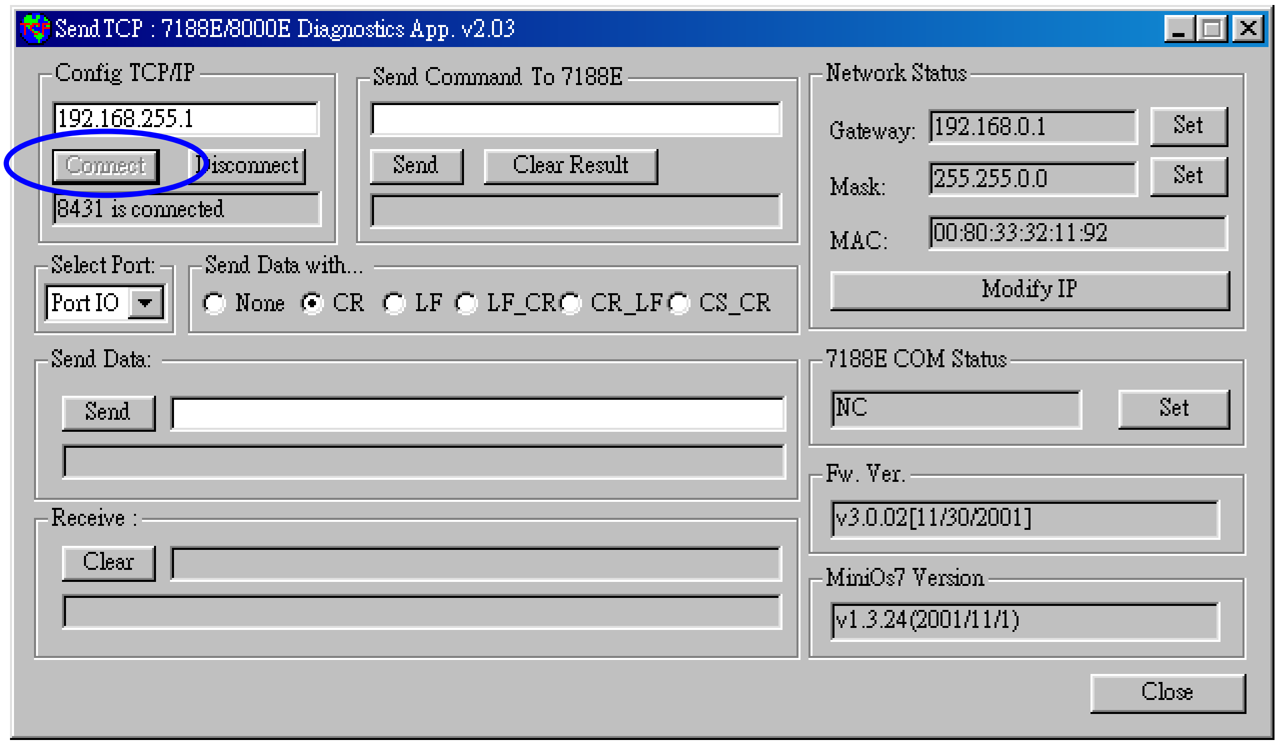
Send Data with: ( ) None [✓ CR] ( ) LF ( ) LF_CR ( ) CR_LF ( ) CS_CR

7188E COM Status: NC

Fw. Ver.: NC

MiniOs7 Version: NC

Step 2: Press "**Connect**" button to connect to the 8431.



IP: 192.168.255.1

8431 is connected

Select Port: Port IO

Gateway: 192.168.0.1

Mask: 255.255.0.0

MAC: 00:80:33:32:11:92

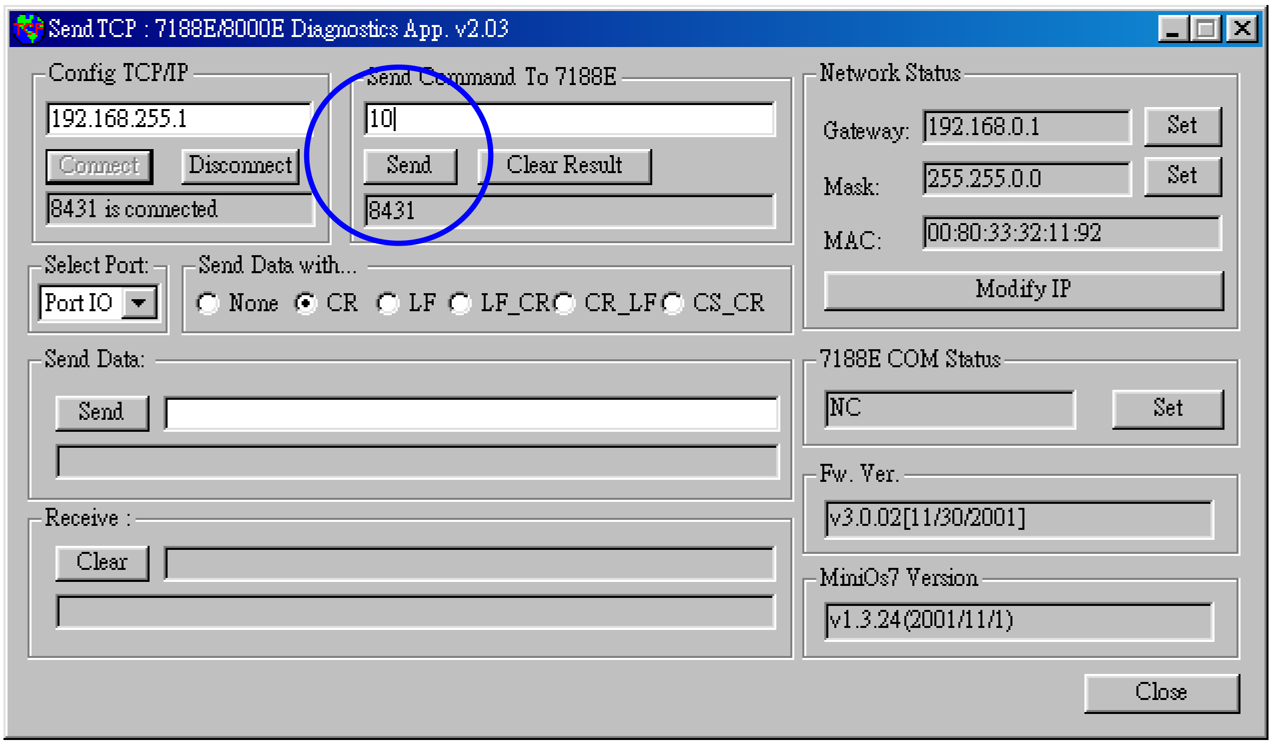
Send Data with: ( ) None [✓ CR] ( ) LF ( ) LF_CR ( ) CR_LF ( ) CS_CR

7188E COM Status: NC

Fw. Ver.: v3.0.02[11/30/2001]

MiniOs7 Version: v1.3.24(2001/11/1)

Step 3: Send command “**10**” to the 8431.



Send TCP : 7188E/8000E Diagnostics App. v2.03

Config TCP/IP: 192.168.255.1 — Connect / Disconnect — 8431 is connected

Send Command To 7188E: 10 — Send / Clear Result — 8431

Select Port: Port IO

Send Data with...: None [✓ CR] LF LF_CR CR_LF CS_CR

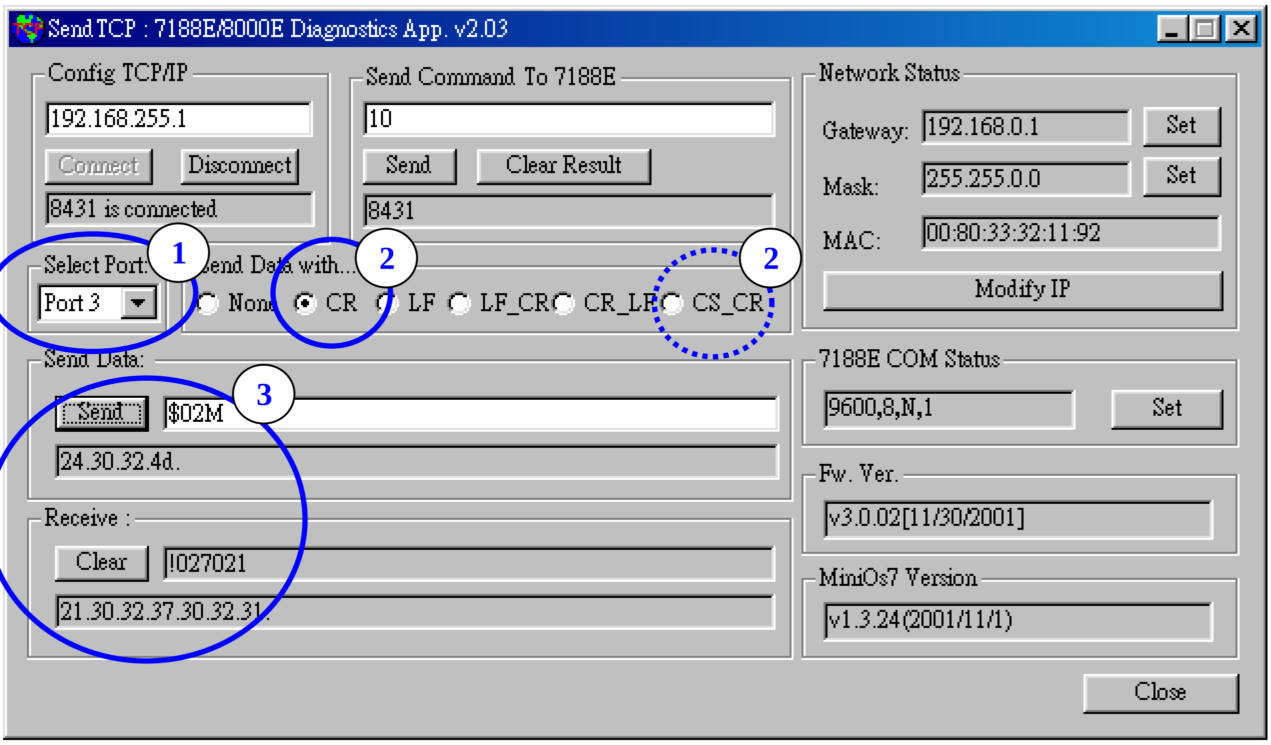
Network Status: Gateway: 192.168.0.1 — Mask: 255.255.0.0 — MAC: 00:80:33:32:11:92 — Modify IP

7188E COM Status: NC

Fw. Ver.: v3.0.02[11/30/2001]

MiniOs7 Version: v1.3.24(2001/11/1)

Step 4: Select “**Port 3**” and “**CR**”. Then send “**$02M**” to read the 7000 module’s ID which is connected to the 8431’s COM3. If you enable the 7000 module’s checksum function, select “**CS_CR**”. The “**CS_CR**” option will add two checksum bytes, then adds “**CR”**.



Send TCP : 7188E/8000E Diagnostics App. v2.03

Config TCP/IP: 192.168.255.1 — Connect / Disconnect — 8431 is connected

Send Command To 7188E: 10 — Send / Clear Result — 8431

Select Port: Port 3

Send Data with...: None [✓ CR] LF LF_CR CR_LF CS_CR

Send Data: $02M — 24.30.32.4d.

Receive: !027021 — 21.30.32.37.30.32.31.

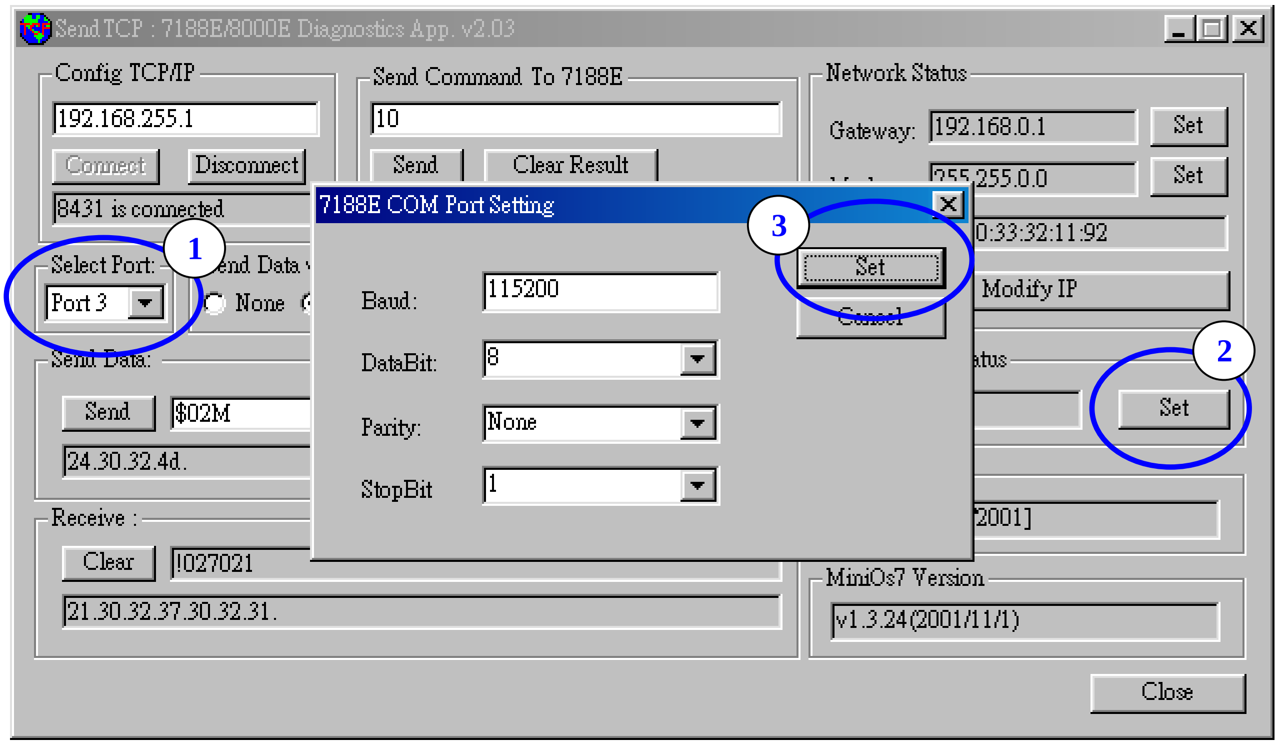
Network Status: Gateway: 192.168.0.1 — Mask: 255.255.0.0 — MAC: 00:80:33:32:11:92 — Modify IP

7188E COM Status: 9600,8,N,1

Fw. Ver.: v3.0.02[11/30/2001]

MiniOs7 Version: v1.3.24(2001/11/1)

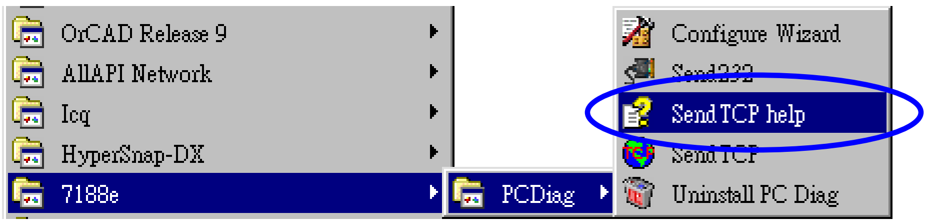
If you want to change the 7188E/8000E's COM ports settings, click "**Set**" to change them. The 7188E/8000E's COM port that you want to configure is specified by "**Select Port**" combo list. Port 3 means you want to configure the 7188E/8000E's COM3.



192.168.255.1
8431 is connected
10
Gateway: 192.168.0.1
255.255.0.0
0:33:32:11:92
Select Port: Port 3
None
$02M
24.30.32.4d.
2001]
!027021
21.30.32.37.30.32.31.
v1.3.24(2001/11/1)

Baud: 115200
DataBit: 8
Parity: None
StopBit: 1

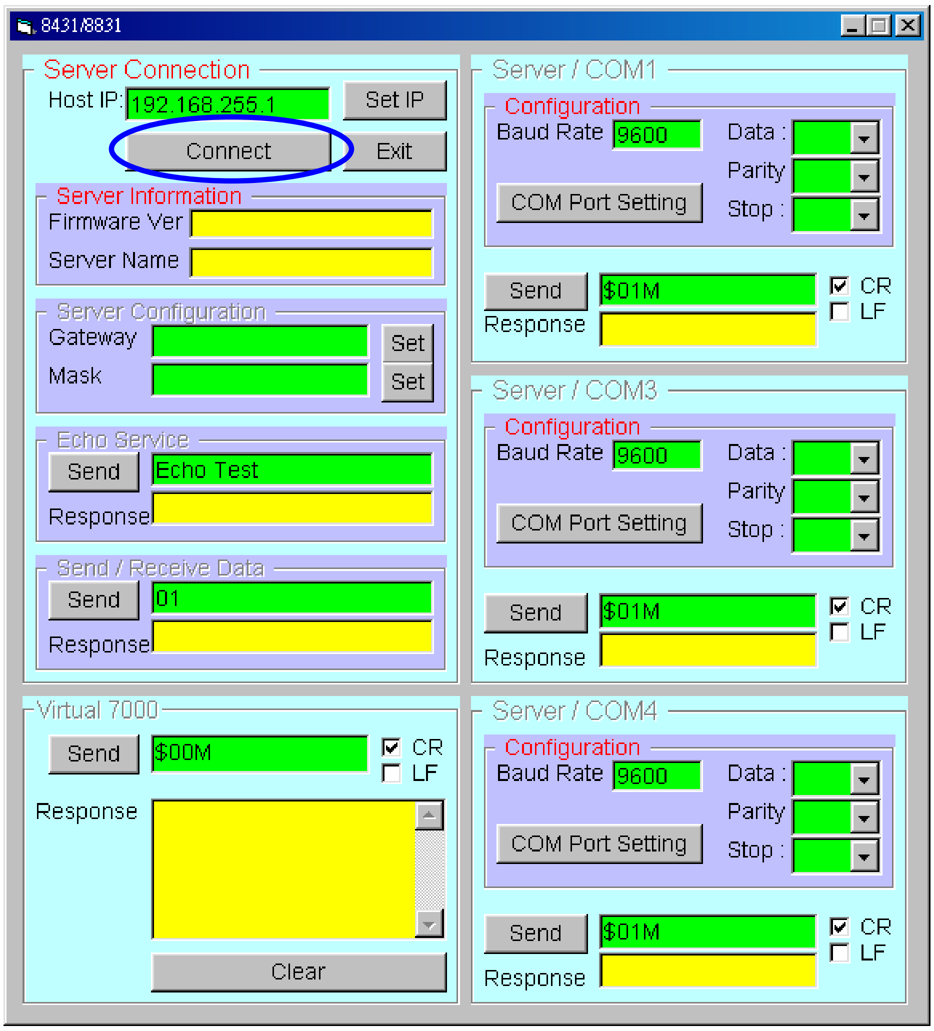
Please refer to SendTCP's help document for more information.



4.6.3 Using 8431.exe to link 8431

Step 1: Make sure the firmware inside 8000E is **Demo36.exe** (default shipping Xserver for 8431/8831).

Step 2: Run 8000\843x883x\TCP\Xserver\8431\Vb5\8431.exe in host-PC.


8431/8831

Server Connection
Host IP: 192.168.255.1
Set IP
Connect
Exit

Server Information
Firmware Ver
Server Name

Server Configuration
Gateway
Mask
Set
Set

Echo Service
Send
Echo Test
Response

Send / Receive Data
Send
01
Response

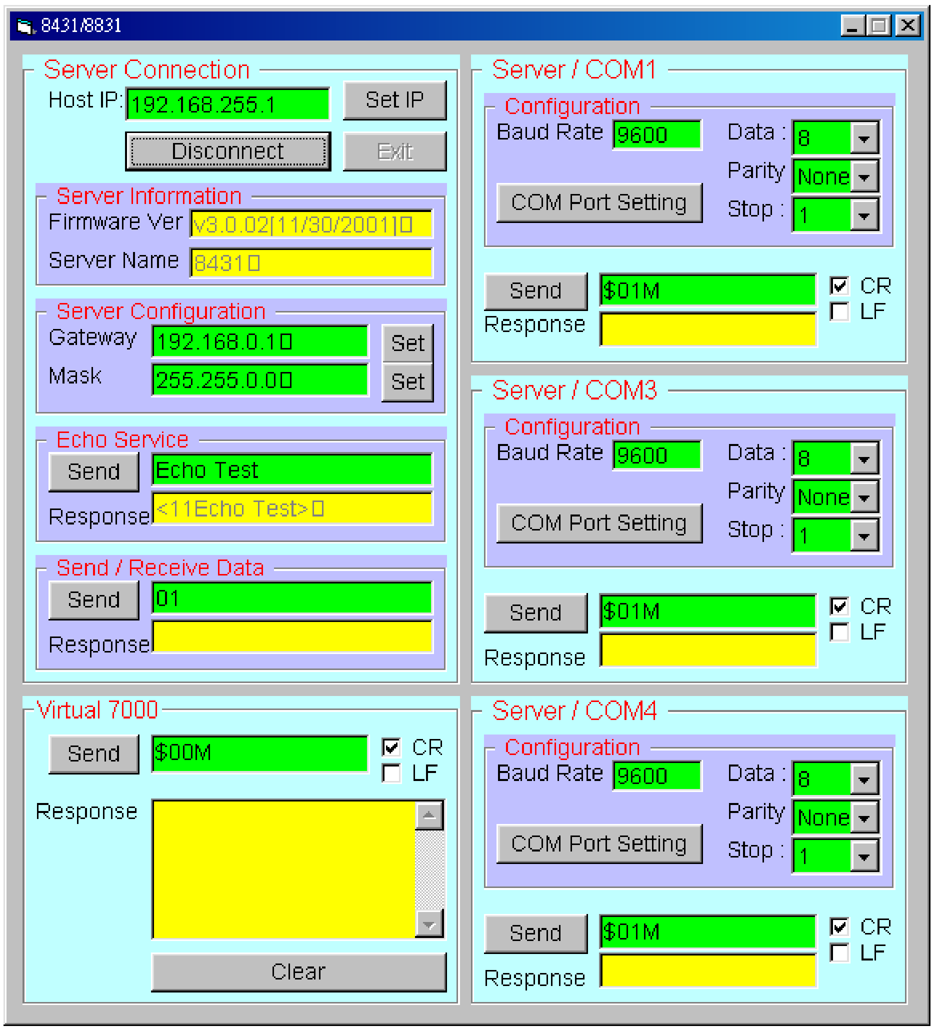
Virtual 7000
Send
$00M
☑ CR
☐ LF
Response
Clear

Server / COM1
Configuration
Baud Rate 9600
Data :
Parity
Stop :
COM Port Setting
Send
$01M
☑ CR
☐ LF
Response

Server / COM3
Configuration
Baud Rate 9600
Data :
Parity
Stop :
COM Port Setting
Send
$01M
☑ CR
☐ LF
Response

Server / COM4
Configuration
Baud Rate 9600
Data :
Parity
Stop :
COM Port Setting
Send
$01M
☑ CR
☐ LF
Response

Step 3: Press "**Connect**" button. The program will then readout relative information concerning the 8431.


8431/8831

Server Connection
Host IP: 192.168.255.1 [Set IP]
[Disconnect] [Exit]

Server Information
Firmware Ver: v3.0.02[11/30/2001]□
Server Name: 8431□

Server Configuration
Gateway: 192.168.0.1□ [Set]
Mask: 255.255.0.0□ [Set]

Echo Service
[Send] Echo Test
Response: <11Echo Test>□

Send / Receive Data
[Send] 01
Response:

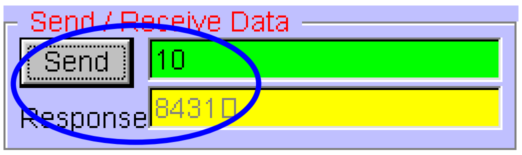
Virtual 7000
[Send] $00M ☑ CR ☐ LF
Response:
[Clear]

Server / COM1
Configuration
Baud Rate: 9600 Data: 8 Parity: None Stop: 1
[COM Port Setting]
[Send] $01M ☑ CR ☐ LF
Response:

Server / COM3
Configuration
Baud Rate: 9600 Data: 8 Parity: None Stop: 1
[COM Port Setting]
[Send] $01M ☑ CR ☐ LF
Response:

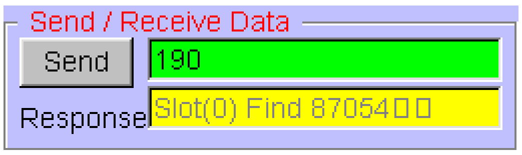
Server / COM4
Configuration
Baud Rate: 9600 Data: 8 Parity: None Stop: 1
[COM Port Setting]
[Send] $01M ☑ CR ☐ LF
Response:

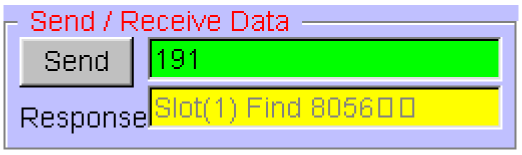
Step 4: Send “**10**” to readout the module name.


Send / Receive Data
[Send] 10
Response: 8431□

Step 5: Send “19N” to read the module’s ID which is plugged in 8431, with N being the slot number.

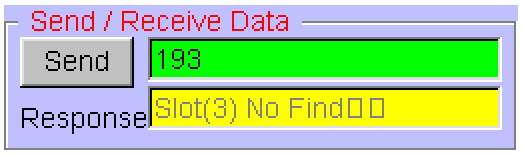
Note: The firmware inside 8000E must be **Demo36.exe** (default shipping Xserver for 8431/8831).


Send / Receive Data
[Send] 190
Response: Slot(0) Find 87054□□


Send / Receive Data

Send 191

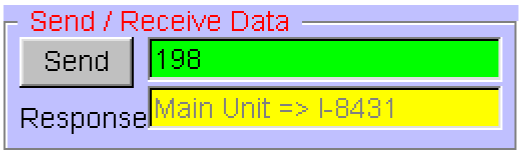
Response Slot(1) Find 8056□□


Send / Receive Data

Send 193

Response Slot(3) No Find□□

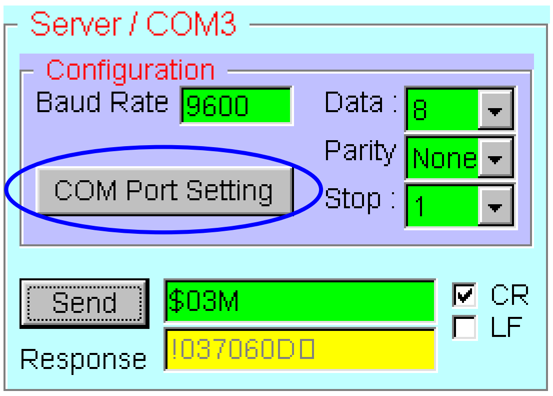
Step 6: Send “19N” to read the main unit’s name , with N larger than 7.


Send / Receive Data

Send 198

Response Main Unit => I-8431

Step 7: Send “$03M” to read the 7000 module’s ID which is connected to 8431’s COM3 (using CA0915 to connect 8431’s COM3 with 7520 directly).


Server / COM3

Configuration

Baud Rate 9600

Data : 8

Parity None

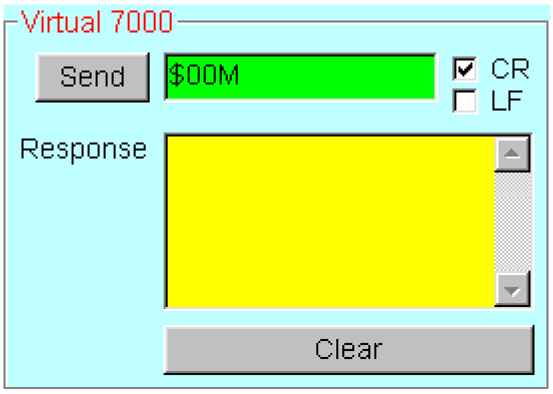
Stop : 1

Send $03M

[✓ CR] LF

Response !037060D□

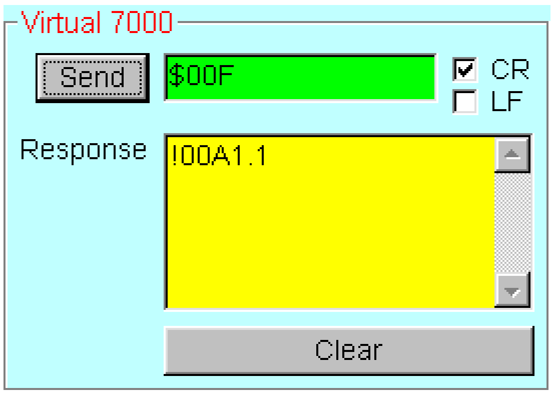
Note: You cannot get any correct response when you download any Xserver program (excep you implement the firmware in v7000.c) into 8000Es by sending commands (like “$00M”) from Virtual 7000 function.



$00M

☑ CR
☐ LF

If you want to use 8000's commands to access modules, you must first download **8KE10.exe** (default shipping for 8430/8830) into the 8000E. Then, reconnect to the 8000E and send commands form Virtual 7000.



$00F

☑ CR
☐ LF

Response: !00A1.1

Of course, using the 7000 Utility is the easiest way to configure modules plugged in 8000E. Please refer to sec. 3.6.1 example 3 for more information.

4.7 Demo Programs of Xserver

We will provide 50~100 demo programs for our users. Please refer to 8000\843x883x\TCP\Xserver\Xserver.htm and **Function.htm** for more information about demo programs of Xserver and client programs for Xserver.

Demo5: Echoes special string to clients

Part of the program body:

```
int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
        // user's command interpreter
        // refer to all demo

         strcpy(Response,"8000_Series."); // returns "8000_Series." string to clients.
         return 1;                  // return OK
}
```

Demo7: Uses printCom1 to debug programs

Part of the explanation head:

```
//   19~!@#$  -> Any non-null command will be accepted.

//    You can use "Print" or "printCom1" to send Debug string to PC monitor by 8000E's COM1.
//    "printCom1" is the defualt function of Xserver.
//    If you want use "Print", you must use "DisableCom" and “RestoreCom”
//    disable "printCOM1".

//   Hardware: 8000E
```

Part of the program body:

```
int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
  // user's command interpreter
  // refer to all demo

        if (Cmd[0])    // Not Null command
        {
                printCom1("%s\n\r",Cmd); // Send debug string to PC monitor by 8000E's COM1
                strcpy(Response,Cmd);
                return 1;                // return OK
        }
        return 0;                  // return ERROR
}
```

Demo9: Timer trigger demo

Part of the explanation head:

```
//   19c, 19C  -> clear count value
//   19r, 19R  -> read count value

//   UserCount will be executed every second.
//   Count value will be increased in UserCount.
//   PC can read count value to know number of seconds
//   after count value is cleared.

//   Hardware: 8000E
```

Part of the program body:

```
unsigned int cnt;
char c_cnt[20];

void UserCount(void)
{
        // user's timer trigger function
        // refer to demo9 for example code
        // printCom1("\n\rUserCount %d",cnt);
        cnt++;
}
```

```
void UserInit(void)
{
        // user's initial function
        // timer initialized for UserCount()
        // I/O or variables initialized for UserLoopFun()
        // I/O or variables initialized for User's functions in this file
        // refer to demo9 & demo11 for example code
        cnt=0;
        AddUserTimerFunction(UserCount,1000); // call UserCnt every 1000 ms
}

int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
        // user's command interpreter
        // refer to all demo

        int i;
        if (Cmd[0])             // Not Null command
        {
                switch(Cmd[0])
                {
                        case 'c':         // 19c, 19C --> clear count
                        case 'C':
                                cnt=0;
                                break;
                        case 'r':             // 19r, 19R --> read count
                        case 'R':
                                break;
                        default:
                                return 0;    // Command Error
                }

                sprintf(c_cnt,"%d",cnt);
                strcpy(Response,c_cnt); // return count value
                return 1;              // return OK
        }
        return 0;                       // return ERROR
}
```

Demo12: Scan-time evaluation

Part of the explanation head:

```
//   19c, 19C  -> clear count value
//   19r, 19R  -> read count value

//   UserLoopFun will increase count value every scan loop.
//   PC can read how many scan loops after clear count value.
//   So user can use this demo to test the Xserver’s performance.

//   Hardware: 8000E
```

Part of the program body:

```
unsigned long  cnt;

void UserInit(void)
{
       cnt=0;
}

void UserLoopFun(void)
{
       cnt++;
}

int UserCmd(unsigned char *Cmd,unsigned char *Response)
{

       if (Cmd[0])            // Not Null command
       {
            switch(Cmd[0])
            {
                  case 'c':           // "19c" "19C" --> clear
                  case 'C':
                        cnt=0;
                        break;
                  case 'r':             // "19r" "19R" --> read count
                  case 'R':
                        break;
                        break;
                  case 'r':             // "19r" "19R" --> read count
                  case 'R':
                        break;
                  default:
                        return 0;  // Command Error
            }
            sprintf(Response,"%d",cnt);  // return count value
            return 1;                 // return OK
      }
      return 0;                      // return ERROR
}
```

Demo14: Controls 7-SEG LEDs

Part of the explanation head:

```
//   1912345, 19abcde  -> Show 5 digits to 7-SEG LED

//   Show5DigitLed, Show5DigitLedWithDot can show 5 digits to 7-SEG LEDs
//   The two functions can show '0' ~ '9'
//                                        'a' ~ 'f'
//                                        'A' ~ 'F'
//                                        ' ', '-', '.'

//   Hardware: 8000E
```

Part of the program body:

```
extern int bLedControl;  // key variable,
                        // 0: disable LED control.
                        //    Xserver will not show default information on LED.
                        // 1: enable LED control.
                        //    Xserver will show default information on LED.

void UserInit(void)
{
        Init5DigitLed();
        bLedControl=0;
}
char cLED[6];

int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
// user's command interpreter
 int i;

 // set default LED char is blank.
 for(i=0;i<5;++i)
        cLED[i]=16;

 strcpy(cLED,Cmd);

 for(i=0;i<5;++i){
        if (cLED[i]>='0' && cLED[i]<='9')
                cLED[i]=cLED[i]-'0';
        if (cLED[i]>='a' && cLED[i]<='f')
                cLED[i]=cLED[i]-'a'+10;
        if (cLED[i]>='A' && cLED[i]<='F')
                cLED[i]=cLED[i]-'A'+10;
        if (cLED[i]==' ')
                cLED[i]=16;
        if (cLED[i]=='-')
                cLED[i]=17;
```

```
       if (cLED[i]=='.')
              cLED[i]=18;
       Show5DigitLed(i+1,cLED[i]);
 }
 strcpy(Response,Cmd);
 return 1;
}
```

Demo18: Reads I-7000 series module's ID.

Part of the explanation head:

```
//   19  -> Any command will be accepted.

//   This demo shows how to communication with the 7000 series which are connected
//   to COM3 of the 8000E.

//   Hardware: 8000E + any 7000 series module whose address is 01,
//                                                        baudrate is 9600,
//                                                        checksum is disable.
```

Part of the program body:

```
char cStr[9];            // Receive data from 7000: "!017021" + 0x0c
char cModuleID[5];       // Store module ID: "7021" + 0x00

void UserCount(void)
{
       int i;

       SendCmdTo7000(3,"$01M",0);
       ReceiveResponseFrom7000(3,cStr,1000,0);

       // cut "!01" and cr
       strcpy(cModuleID,cStr+3);
       cModuleID[4]=0;


       // Show moduleID to PC monitor
       printCom(1,"Address: 01,  ModuleID:%s\n\r",cModuleID);
}
```

```
void UserInit(void)
{
        AddUserTimerFunction(UserCount,1000);
        InstallCom(2,9600,8,0);
        SetBaudrate(1,115200L);
}
```

Demo19: Reads system serial number.

Part of the explanation head:

```
//   19  -> Any command will be accepted.

//   Unique hardware serial number is used to protect user's software.
//   Using 7188xw.exe to enter 7188E and then execute command 'ver'.
//   MiniOS7 will show the serial number.
//   User can check the number at first, then decide to execute
//   Xserver forward.

//   Hardware: [8431] or [8831]
```

Part of the program body:

```
int bSerialNumOk;
char cID[8]={0x9,0x31,0xa4,0x39,0x3,0,0,0x5};

void UserInit(void)
{
        char cSerialNumber[8];

        GetSerialNumber(cSerialNumber);
        if(!strcmp(cSerialNumber,cID))
                bSerialNumOk=1;                 // Matching !
        else
                bSerialNumOk=0;                 // Unmatched !
}
```

```
int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
        if (bSerialNumOk)
                strcpy(Response,"ID ok.");
        else
                strcpy(Response,"ID error.");
        return 1;               // return OK
}
```

Demo37 (8E only): Reads SystemKey status.

Part of the explanation head:

```
//   23 on, 23 On, 23 ON  → Turns on  auto reply system key status.
//   23on, 23On, 23ON     → Turns on auto reply system key status.
//   23 other             → Turns off

//   This demo use command 23 to switch on/off auto return system status.
//   PC send command "23 on" at beginning. When user press system key,
//   Xserver will auto reply which system key pressed.

//   Hardware: 8000E
```

Part of the program body:

```
char sMode[3];                  //Switchs on/off auto reply system key status.
TCPREADDATA t_Status;       //Variable to record socket.

void UserLoopFun(void)
{
        // VxComm.exe will call this function in every scan time
        // refer to demo11 for scan-time evaluation

        if(!strcmp(sMode,"ON"))
        {
                if(IsSystemKey())
                {
                        switch(GetSystemKey())
                        {
```

```
                case SKEY_UP:
                    t_Status.ReadUartChar="UP";
                    t_Status.Length=3;
                    break;
                case SKEY_DOWN:
                    t_Status.ReadUartChar="DOWN";
                    t_Status.Length=5;
                    break;
                case SKEY_SET:
                    t_Status.ReadUartChar="SET";
                    t_Status.Length=4;
                    break;
                case SKEY_MODE:
                    t_Status.ReadUartChar="MODE";
                    t_Status.Length=5;
                    break;
            }//of switch
            // Send data to PC
            VcomSendSocket(t_Status.Socket,t_Status.ReadUartChar,t_Status.Length);
        }
    }
}

int VcomUserBinaryCmd(TCPREADDATA *p)
{
    t_Status.Socket=p->Socket;
    p->ReadUartChar[p->Length]=0;  // Add zero end

    sscanf(p->ReadUartChar+2,"%s",sMode);

    //Turn to upper case
    strupr(sMode);

    if(!strcmp(sMode,"ON"))
        VcomSendSocket(t_Status.Socket,"Turn on auto reply.",19);  // String length is 19
    else
        VcomSendSocket(t_Status.Socket,"Turn off auto reply.",20); // String length is 20

    return 1;  /* any value will be accept */
}
```

Demo40 (8E only): D/I from 8000 modules (16 DI channels).

Part of the explanation head:

```
//   19i 1, 19I 1  -> D/I from slot 1.
//   19i1 , 19I1   -> D/I from slot 1.

//   Uses DI_16 to D/I from parallel DI modules(16 channels) on 8000 slots.
//   Shows value both decimal & binary mode
//                        0.......0
//                        bit0....bit15

//   Hardware: 8000E + parallel DI module (16 DI channels)
```

Part of the program body:

```
int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
      // user's command interpreter
      // refer to all demo
      char sResult[17];     // Store 16 bits binary input status.      High → 1
                            //                                         Low → 0
      int iSlot,iTotalChannel=16;
      unsigned int iValue;        // Store 16 bits decimal input value.
      unsigned int iBit;          // Store input value. Used to read every bit status.
      int i;

      if (Cmd[0]=='i' || Cmd[0]=='I')             // Not Null command
      {
            sscanf(Cmd+1,"%d",&iSlot);
            iValue=DI_16(iSlot);

            iBit=iValue;
            for(i=0;i<iTotalChannel;i++)
            {
                  if((iBit&1)!=0)
                        sResult[i]='1';
                  else
                        sResult[i]='0';
                  iBit>>=1;
            }
```

```
          sResult[iTotalChannel]=0;  // string with zero end.

          sprintf(Response,"%u (DI0-->DI%d) %s",iValue,iTotalChannel-1,sResult);
          return 1;             // return OK
}
     return 0;        // return ERROR
}
```

Demo43 (8E only): D/O to 8000 modules (8 channels).
Part of the explanation head:

```
//   19o 1 0F  -> D/O to slot 1. 0F is Hex value means DO0~DO3 are on.
//   19O 1 0F
//   19o1 0F
//   19O1 0F

//   Uses DO_8 to D/O to parallel DO modules (16 DO channels) on 8000 slots.

//   Hardware: 8000E + parallel DO modules (8 DO channels)
```

Part of the program body:

```
int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
     // user's command interpreter
     // refer to all demo
     unsigned int iValue; // Output value.
     int iSlot;

     if (Cmd[0]=='o' || Cmd[0]=='O')          // Not Null command
     {
          sscanf(Cmd+1,"%d %x",&iSlot,&iValue);
          DO_8(iSlot,iValue);

          sprintf(Response,"Output ok.");
          return 1;             // return OK
     }
     return 0;        // return ERROR
}
```

5. Glossary

1. **Ethernet:**
   The term Ethernet generally refers to a standard published in 1982 by Digital Euqipment Corp., Intel Corp. and Xerox Corp. Ethernet is the most popular physical layer local area network technology today. Ethernet is a best-effort delivery system that uses CSMA/CD technology. It recognizes hosts using 48-bit MAC address.

2. **Internet:**
   Physically, a collection of packet switching networks interconnected by gateways along with TCP/IP protocol that allows them to perform logically as a single, large and virtual network. Internet recognizes hosts using 32-bit IP address.

3. **TCP/IP:**
   Transmission Control Protocol (TCP) and Internet Protocol (IP) are the standard network protocols. They are almost always implemented and used together and called TCP/IP. TCP/IP can be used to communicate across any set of interconnected network.

4. **TCP (Transmssion Control Protocol):**
   TCP provides a reliable flow of data between two hosts. It is connected with things such as dividing the data passed to it from applications into appropriately sized chunks for the network layer below, acknowledging received packets, setting timeouts to make certain that the other end acknowledges packets that are sent, and so on.

5. **UDP (User Datagram Protocol):**
   UDP provides a much simpler service to the application layer. It just sends packets of data from one host to the other. But there is no guarantee that the packets reach the destination host.

6. **Gateway:**
   Computers that interconnect two networks and pass packets form one to the other are called Internet Gateways or Internet Routers. Gateways route packets are based on destination network, not on destination host.

7. **IP (Internet Protocol) address:**

Every interface on an Internet must have a unique IP address (also called an Internet address). These addresses are 32-bit numbers. They are normally written as four decimal numbes, one for each byte of the address such as "**192.168.41.1**". This is called dotted-decimal notation.

8. **MAC (Media Access Control) address:**

To allow a computer to determine which packets are meant for it, each computer attached to an Ethernet is assigned a 48-bit interger known as its MAC address (also called an Ethernet address, hardware address or physical address). They are normally written as eight hexadecimal numbers such as "**00:71:88:af:12:3e:0f:01**". Ethernet hardware manufacturers purchase blocks of MAC addresses and assign them in sequence as they manufacture Ethernet interface hardware. Thus, no two hardware interfaces have the same MAC address.

9. **Subnet Mask:**

Subnet mask is often simply called mask. Given its own IP address and its subnet mask, a host can determine if a TCP/IP packet is destined for a host that is (1) on its own subnet, (2) on a different network. If (1), the packet will be delivered directly; else, will be delivered by gateways or routers.

10. **ARP (Address Resolution Protocol):**

Consider two machines A and B that share a physical network. Each has an assigned IP address IP_A and IP_B and a MAC address MAC_A and MAC_B. The goal is to devise low-level software that hides MAC addresses and allows higher-level programs to work only with IP addresses. Ultimately, however, communication must be carried out by physical networks using whatever MAC address scheme the hardware supplies.

Suppose machine A wants to send a packet to machine B across a physical network to which they both are attached, but A has only B's Internet address IP_B. The question arises: how does A map that address to B's MAC address, MAC_B?

ARP provides a dynamic mapping from 32-bit IP address to the corresponding 48-bit MAC address. We use the term dynamic since it happens automatically and is normally not a concern of either the application user or the system administrator.

11. RARP (Reverse Address Resolution Protocol):

RARP provides a dynamic mapping from 48-bit MAC address to the corresponding 32-bit IP address.

$$\begin{tikzcd}
\text{32-bit IP address} \arrow[d, shift right=3, "\text{ARP}"'] \\
\text{48-bit MAC address} \arrow[u, shift right=3, "\text{RARP}"']
\end{tikzcd}$$

12. ICMP (Internet Control Messages Protocol):

No system works correctly all the time. The ICMP provides communication between the Internet Protocol software on one machine and the Internet Protocol software on another. It allows gateways to send error or control messages to other gateways or hosts to know what wrong with the network communication.

13. Ping:

Ping sends an ICMP echo request message to a host, expecting an ICMP echo reply to be returned. Normally if you cannot Ping a host, you won’t be able to Telnet or FTP to the host. Conversely, if you cannot Telnet or FTP to a host, Ping is often the starting point to determine what the problem is.

14. Packet:

The unit of data sent across a physical network. It is consisted of a series of bits containing data and control information, including source and destination node (host) address, formatted for transmission from one node to another.

15. Socket:

Each TCP segment contains the source and destination port number to identify the sending and receiving application. These two values, along with the source and destination IP address in the IP header, uniquely identify each connection.

The combination of an IP address and a port number is called a socket.

16. Clients and Servers:

The client-server paradigm uses the direction of initiation to categarize whether a program is a client or server. In general, an application program that initiates peer to peer communication is called a client. End users usually invoke client programs when they use network services.

Most client program consists of conventional application program develop tools. Each time a client program executes, it contacts a server, sends a request and awaits a response. When the response arrives, the client program continues processing. Client programs are often easier to develop than servers, and usually require no special system privileges to operate.

By comparison, a server is any program that awits for incoming requests from a client program. The server receives a client's request, performs the necessary computation and returns the result to the client.

17. Firmware:

Alterable programs in semipermanent storage, e.g., ROM, EEPROM, or Flash memory.